

Белорусский национальный технический университет
Факультет строительный
Кафедра «Строительные материалы и технология строительства»

Электронный учебно-методический комплекс

РЕШЕНИЕ ТЕХНОЛОГИЧЕСКИХ ЗАДАЧ НА ЭВМ

для студентов специальности 1-70 01 01 «Производство
строительных изделий и конструкций»

Составители: ст. преподаватель Ж.Л. Зеленковская
 ст. преподаватель О.Б. Сенько

Рассмотрено и утверждено
на заседании совета строительного факультета
23 декабря 2022 г.
протокол N 3

Минск БНТУ 2022

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
РАЗДЕЛ 1. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ. ЛЕКЦИОННЫЙ КУРС ПО ДИСЦИПЛИНЕ "РЕШЕНИЕ ТЕХНОЛОГИЧЕСКИХ ЗАДАЧ НА ЭВМ".....	7
1 Алгоритмические языки программирования. Возможности алгоритмических языков	8
1.1 Эффективность применения информационных технологий в области строительства	8
1.2 Объектно-ориентированное программирование.....	10
1.3 Данные, Выражения, Функции	24
1.4 Выбор в программе, организация циклов	30
1.5 Программирование графики.....	39
1.6 Массивы данных, создание и использование вызываемых процедур и функций	41
2 ГРАФИЧЕСКИЕ ПРИМИТИВЫ, СВОЙСТВА ОБЪЕКТОВ	47
2.1 Развитие систем автоматизированного проектирования	47
2.1.1 Общие сведения о системе AutoCAD	52
2.1.2 Команды системы AutoCAD	59
2.1.3 Ввод координат и настройка параметров чертежа	65
2.1.4 Графические примитивы как основа изображений	74
2.1.5 Сложные примитивы. Работа с OLE-объектами.....	82
2.1.6 Основные свойства объектов	90
2.2 Редактирование объектов, размерные стили	98
2.2.1 Команды редактирования.....	98
2.2.2 Размеры и размерные стили	114
2.2.3 Работа с текстом	123
2.2.4 Штриховка и градиент	128
2.2.5 Получение твердой копии чертежа	130
2.3 Построение технологических схем и спецификаций	138
2.3.1 Построение сложных объектов.....	138
2.3.2 Создание таблиц	147
2.3.3 Блоки, внешние ссылки и динамические блоки.....	153
2.3.4 Видовые экраны в пространстве модели и в пространстве листа	171

3. РАЗДЕЛ 2. ПРАКТИЧЕСКИЙ РАЗДЕЛ. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ ПО ДИСЦИПЛИНЕ "РЕШЕНИЕ ТЕХНОЛОГИЧЕСКИХ ЗАДАЧ НА ЭВМ"	181
3.1 Лабораторная работа №1 "ЗНАКОМСТВО С ИНТЕРФЕЙСОМ СИСТЕМЫ ПРОГРАММИРОВАНИЯ VISUAL BASIC"	182
3.2 Лабораторная работа №2 "ТИПЫ ДАННЫХ. ОПЕРАТОРЫ ПРИСВАИВАНИЯ. ПРОЦЕДУРЫ"	189
3.3 Лабораторная работа №3 "ФУНКЦИИ ВВОДА-ВЫВОДА И ПРЕОБРАЗОВАНИЯ ТИПОВ ДАННЫХ"	195
3.4 Лабораторная работа №4 "ПРОГРАММИРОВАНИЕ РАЗВЕТВЛЯЮЩИХСЯ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ"	198
3.5 Лабораторная работа №5 "ПРОГРАММИРОВАНИЕ ЦИКЛИЧЕСКИХ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ"	204
3.6 Лабораторная работа №6 "РАБОТА С МАССИВАМИ"	209
3.7 Лабораторная работа №7 "ПРОЦЕДУРЫ И ФУНКЦИИ"	214
3.8 Лабораторная работа №8 "ИСПОЛЬЗОВАНИЕ ГРАФИКИ В VISUAL BASIC"	221
3.9 Лабораторная работа №9 "НАСТРОЙКА ГРАФИЧЕСКОГО ОКНА. СПОСОБЫ ВВОДА КОМАНД. РЕЖИМЫ РИСОВАНИЯ"	227
3.10 Лабораторная работа №10 "КОМАНДЫ ПОСТРОЕНИЯ ГРАФИЧЕСКИХ ПРИМИТИВОВ"	231
3.11 Лабораторная работа №11 "КОМАНДЫ РЕДАКТИРОВАНИЯ" ..	235
3.12 Лабораторная работа №12 "РАБОТА С ТЕКСТОМ. СЛОИ. СВОЙСТВА ОБЪЕКТА"	240
3.13 Лабораторная работа №13 "РАЗМЕРЫ И РАЗМЕРНЫЕ СТИЛИ. РЕДАКТИРОВАНИЕ РАЗМЕРОВ"	242
3.14 Лабораторная работа №14 "ШТРИХОВКА. МУЛЬТИЛИНИИ" ...	245
3.15 Лабораторная работа №15 "ПОСТРОЕНИЕ СЛОЖНЫХ ОБЪЕКТОВ. ТАБЛИЦА И ВИДОВЫЕ ЭКРАНЫ"	250
3.16 Лабораторная работа №16 "ПРОСТЫЕ И ДИНАМИЧЕСКИЕ БЛОКИ"	255
4 Раздел 3. КОНТРОЛЬ ЗНАНИЙ	262
5 Раздел 4. ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ	266
5.1 Литература.....	266
5.2 Учебная программа по учебной дисциплине "Решение технологических задач на ЭВМ"	267

ВВЕДЕНИЕ

Комплексная графическая подготовка инженера в настоящее время тесно связана с освоением компьютерных технологий. Владение современными компьютерными графическими редакторами позволяет выйти на качественно новый уровень проектирования.

Целью изучения дисциплины является формирование у студентов компьютерной культуры и практических навыков записи математической модели поставленной задачи, составления принципиальной схемы алгоритма, реализации алгоритма на алгоритмическом языке программирования, комплекса знаний по современным методам технологического проектирования при решении строительно-технологических задач.

Знания, умения и навыки, приобретенные в результате освоения дисциплины, помогут студентам самостоятельно проектировать технологические схемы и применять средства автоматизации проектирования при решении конкретных задач на базе современных достижений в компьютерной технике на всех этапах последующего обучения, в том числе при курсовом и дипломном проектировании.

В результате изучения дисциплины "Решение технологических задач на ЭВМ" студент должен:

знать:

- краткую характеристику математического пакета и его возможности;
- основы алгоритмизации инженерных задач;
- принципы программирования на алгоритмическом языке;
- технологии применения стандартных программ для компьютерного моделирования технологических задач;
- общие принципы о процессе, технологии и организации технологического проектирования с применением современных технических и программных средств;

уметь:

- ставить прикладные задачи, строить их математические модели, разрабатывать алгоритмы решения;
- реализовывать построенный алгоритм в виде собственной программы на алгоритмическом языке или с использованием стандартных программ;
- применять средства автоматизации проектирования для создания проектной документации;

владеть:

- численными методами при проведении сложных расчетов и науч-

ных исследований для решения технологических задач;

- методами использования компьютерных технологий для построения чертежей;
- приемами построения технологических схем в приложении AutoCAD .

Освоение данной учебной дисциплины обеспечивает формирование умения использовать программные средства для решения технологических задач при производстве сборных бетонных и железобетонных изделий и конструкций. Владеть основными программными средствами автоматизации разработки технологической и конструкторской документации.

В последнее время возрос интерес к программированию. Это связано с развитием и внедрением в повседневную жизнь информационно-коммуникационных технологий.

Бурное развитие вычислительной техники, потребность в эффективных средствах разработки программного обеспечения привели к появлению систем программирования, ориентированных на так называемую "быструю разработку", пионером среди которых был пакет Microsoft Visual Basic. В основе систем быстрой разработки (RAD-систем, Rapid Application Development — среда быстрой разработки приложений) лежит технология визуального проектирования и событийного программирования. Суть этой технологии заключается в том, что среда разработки берет на себя большую часть рутинной работы, оставляя программисту работу по созданию диалоговых окон и функций обработки событий. Visual Basic — это система программирования, предназначенная для написания программ, работающих под управлением операционной системы Windows. Используя Visual Basic, можно разрабатывать очень сложные приложения практически для любой области современных компьютерных технологий: бизнес-приложения, игры, мультимедиа, базы данных.

Данная дисциплина поможет изучить основные принципы программирования на языке и получить необходимые навыки для создания собственных программ на этом языке и произвести автоматизацию разработки конструкторской документации средствами VB.

Комплексная графическая подготовка инженера в настоящее время тесно связана с освоением компьютерных технологий. Владение современными компьютерными графическими редакторами позволяет выйти на качественно новый уровень проектирования.

Учебное пособие предназначено также для обучения основам работы с графическим пакетом AutoCAD. AutoCAD – интерактивная графическая система, позволяющая в режиме диалога разрабатывать чертежи на плоскости,

моделировать объекты в трехмерном пространстве, а также создавать программные модули на встроенном алгоритмическом языке AutoLISP.

Практически все основные функциональные возможности AutoCAD нашли отражение в лабораторных работах. Знания, умения и навыки, приобретенные в результате освоения дисциплины, помогут студентам самостоятельно проектировать технологические схемы и применять средства автоматизации проектирования при решении конкретных задач на базе современных достижений в компьютерной технике на всех этапах последующего обучения, в том числе при курсовом и дипломном проектировании.

В лабораторных работах дается подробное описание команд, их опций и указана последовательность действий при решении определенных задач, что позволяет студенту практически самостоятельно освоить команды построения и редактирования графических изображений, работу со слоями, основы простановки размеров, работу с текстом и OLE-объектами.

Современный уровень программных и технических средств электронной вычислительной техники дает возможность перехода от традиционных, ручных методов конструирования к новым информационным технологиям с использованием компьютеров, к созданию систем автоматизированного проектирования и разработки конструкторской документации, удовлетворяющих стандартам ЕСКД.

**РАЗДЕЛ 1. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ. ЛЕКЦИОННЫЙ КУРС ПО
ДИСЦИПЛИНЕ "РЕШЕНИЕ ТЕХНОЛОГИЧЕСКИХ ЗАДАЧ НА ЭВМ"**

1 АЛГОРИТМИЧЕСКИЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ. ВОЗМОЖНОСТИ АЛГОРИТМИЧЕСКИХ ЯЗЫКОВ

1.1 Эффективность применения информационных технологий в области строительства

История создания языка программирования Basic и Visual Basic

Язык программирования Basic был создан в 1964 году двумя профессорами из Dartmouth College - Джоном Кенеми и Томасом Куртцом для обучения студентов навыкам программирования.

Язык получился настолько простым и понятным, что через некоторое время его начали применять и в других учебных заведениях.

В 1975 году, с приходом первых микрокомпьютеров, эстафету Basic приняли Билл Гейтс и Пол Аллен, основатели Microsoft.

Именно они создали новую версию Basic для первых компьютеров "Альтаир" (MITS Altairs), способную работать в 4КБ оперативной памяти. Со временем именно эта версия и превратилась в один из самых популярных языков программирования в мире и когда появились первые персональные компьютеры IBM PC, именно он стал стандартом в программировании, но уже в виде GW-Basic. Потом был Turbo Basic, QuickBasic, Basic PDS, но всегда при разработке новой версии языка сохранялась совместимость с прежними версиями.

В начале **90-х** появляется операционная система Microsoft Windows с новым графическим интерфейсом пользователя (GUI).

В это время в недрах Microsoft велось несколько параллельных проектов по созданию нового языка программирования для Windows.

И в **1991** году под лозунгом "теперь и начинающие программисты могут легко создавать приложения для Windows" появилась первая версия нового инструментального средства - Microsoft Visual Basic.

И хотя многие С-программисты тихо посмеивались над попытками Microsoft сделать простой и понятный инструмент разработки Windows-программ, **Visual Basic** начал свое победное шествие по миру, и ничто не могло остановить этот процесс.

Последние барьеры упали в 1994 году с выпуском Visual Basic for Applications. Именно в это время, после включения VBA в состав Microsoft Office, Basic начинает превращаться в один из основных стандартов программирования для Windows. Для фирмы Microsoft язык Basic имеет особое значение, в свое время разработка варианта Basic для компьютера Altair 8800 положила начало трудовому программистскому пути ее основателей, Билла Гейтса и Пола Аллена.

BASIC - Beginner's All-purpose Symbolic Instructional Code, что в переводе - Многоцелевой символьный командный код для начинающих.

Первоначально задумывавшийся как игрушка, Visual Basic от Microsoft невероятно быстро завоевал программистский мир.

VB версии 6 - это уже мощный язык, который можно использовать для создания распределенных приложений с применением компонентов COM и Microsoft Transaction Server.

При помощи VBScript и VBA (VB для приложений) можно писать сценарии для web-браузеров и автоматизировать приложения Microsoft Office.

Более того, VB6 можно использовать для создания элементов управления Active-X, работающих вместе с Internet Explorer.

Microsoft предложила трехуровневый подход для архитектур "клиент-сервер", в котором "тонкие" пользовательские интерфейсы взаимодействуют с удаленными компонентами VB для получения данных из базы данных или с другой машины.

Интересно то, что компилятор VB для трансляции использует многопроходный режим, а в конечном счете полагается на компилятор Microsoft C++ для получения выходного машинного кода после компиляции в промежуточный язык.

1964-1985 гг. Первое поколение Basic

1985-1992 гг. Второе поколение Basic

1991 год – VB 1.0 Назначение: решение научно-технических и экономических задач, игры и т.п.

1992 год – VB 2.0 Назначение: Создание полноценных синтаксических языковых конструкций, полномасштабное использование процедур, библиотек подпрограмм, поддержка технологии смешанного программирования и т.д.

1993 год – VB 3.0 Интеграция в состав VB широкого набора "профессиональных" средств (СУБД, OLE – технологий)

1993 г. Переход к третьему поколению Basic - Microsoft Visual Basic (for Windows)

1994 год – VBA 1.0 в составе Excel 4.0

1995 год – VB 4.0

1996 год – VBA 5.0 в составе Microsoft Office 97

1998 год – VB 6.0

1999 год – VBA 6.0 в составе Microsoft Office 2000

В 2001 году VBA уже входит во все офисные приложения.

Средство быстрой разработки с применением событийной схемы управления процессами, визуальной разработки пользовательского интерфейса, объектной программной модели и возможности расширения с помощью внешних компонентов.

Используется только в среде Windows. На рынке полностью доминирует система Microsoft Visual Basic, представляющая собой стандарт "де-факто" языка Basic. На рынке есть несколько поставщиков средств разработки на базе Basic, но их продукты имеют локальное применение в узких вертикальных сегментах.

Сформировался большой рынок поставщиков дополнительных средств для VB.

Интеграция в состав VB широкого набора "профессиональных" средств (СУБД, Интернет, групповая разработка).

1.2 Объектно-ориентированное программирование

Visual Basic – один из первых языков программирования, поддерживающих *событийно-управляемое* программирование. Основной смысл событийно-управляемого программирования следующий: вместо скрупулезного описания каждого шага Вы лишь указываете, как реагировать на различные события (действия пользователя): выбор команды, щелчок в окне, перемещение мыши и т.д. Вы создаете не одну большую программу, а приложение, состоящее из набора взаимодействующих микропрограмм (процедур), управляемых пользователем.

Visual Basic – это среда разработки приложений под Windows, в которую включено все, что необходимо для создания, модификации, тестирования, корректирования и компиляции ваших программ.

Слово *Visual* «визуальный» (наглядный) означает способ разработки пользовательского интерфейса программы.

Интерфейс – совокупность средств, обеспечивающих физическое или логическое взаимодействие устройств и программ вычислительной системы. Здесь мы вкладываем в термин разработка интерфейса более узкий смысл – изобретение и создание такого способа ввода и вывода информации, который был бы максимально прост, удобен и приятен пользователю.

Экран проектирования в системе Visual Basic

После загрузки Visual Basic на экране отображается несколько окон, положением и размерами которых можно управлять (рис.1.1).

Экран содержит окна:

1. Строка меню

Меню содержит команды, используемые при работе Visual Basic . кроме стандартных меню File, Edit, View, Window, Help, здесь расположены меню, обеспечивающие доступ к функциям программирования, например, Project, Format, Debug

2. Панель инструментов (Toolbars Standard)

Предоставляет быстрый доступ к наиболее часто используемым командам среды программирования

3. Форма (Form)– окно будущего приложения

4. Панель элементов управления (Toolbox)

Панель обеспечивает проектировщика набором инструментов, необходимых во время разработки для размещения элементов управления на форме

5. Окно свойств (Properties)

Перечисляет установленные свойства для выбранной формы или элемента управления.

6. Окно проводника проекта (Project Explorer)

Представляет список форм и модулей текущего проекта. Проект – это набор файлов, используемых для построения приложения.

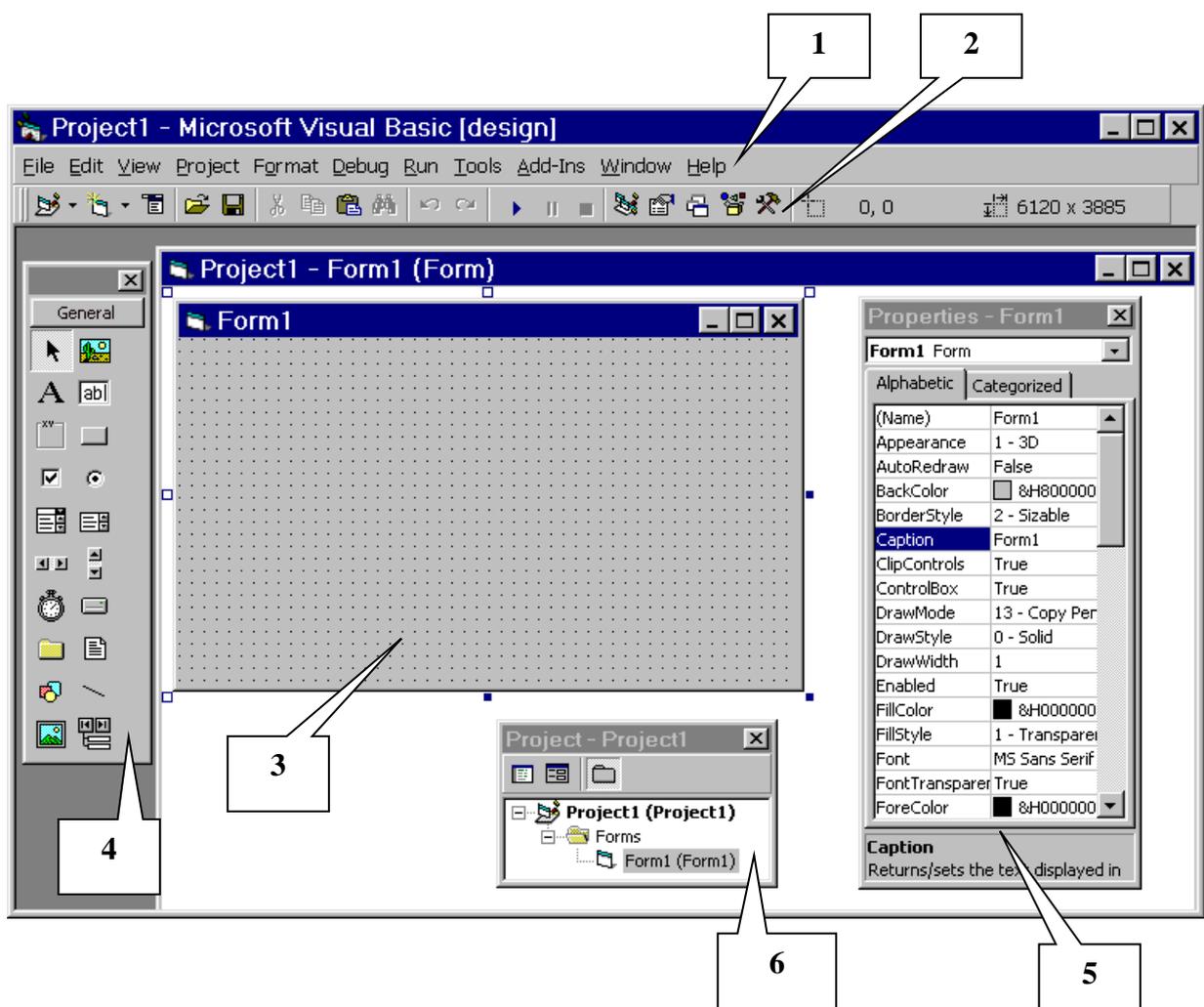


Рисунок 1.1 – Интерфейс Visual Basic

Объекты управления и их свойства

Объектом называется некая сущность, которая, во-первых, четко проявляет свое поведение, а во-вторых, является представителем некоторого класса подобных себе объектов.

Классом объектов называется общее описание объектов, для которых характерно наличие множества общих свойств и общих действий. Примером класса может служить класс **Командная кнопка** – общее описание кнопок в окнах приложений. Эти кнопки могут быть всех цветов и размеров, но должны иметь множество общих свойств и других характеристик, например, событий, которые для этих объектов одинаковы.

Форма – это окно будущего приложения. Форма обладает свойствами, определяющими ее внешний вид, методами, определяющими ее поведение, и событиями, которые определяют ее взаимодействие с пользователем.

Элементы управления – это объекты, содержащиеся внутри объекта - Форма. Каждый тип элемента управления имеет свой собственный набор свойств, методов и событий, что делает его пригодным для определенной цели.

Свойства в Visual Basic - количественно измеряемый атрибут объекта (элемента управления или формы). Значения свойств можно задать на стадии проектирования в окне свойств. Активизировать окно свойств, можно по-разному:

- в строке меню View / Properties Window
- клавиша F4
- кнопка  на панели инструментов

Установка значений свойств объектов

Значения свойств объектов можно менять двумя способами:

- при проектировании:

В каждый момент проектирования только один объект является выделенным (активным). Он окружен рамкой из восьми маркеров. В окне свойств отображается список свойств именно активного объекта. Новое значение свойства вводится в окне свойств.

- при выполнении приложения:

в программный код включается команда, имеющая следующий общий вид:

ИмяОбъекта.Свойство=Значение

Получение значений свойств

Получают значение свойств тогда, когда хотят определить состояние объекта до выполнения каких-либо действий из кода. Общий вид команды следующий:

Переменная=ИмяОбъекта.Свойство

Основные свойства объектов управления

Свойство	Назначение
Name	Имя объекта
Caption	Заголовок
Visible	Видимость
BorderStyle	Стиль границ
FontBold	Полужирный шрифт
FontItalic	Курсив
FontName	Тип шрифта
FontSize	Размер шрифта
FontUnderline	Шрифт подчеркнутый
Enabled	Доступ
Left	Координата по горизонтали
Top	Координата по вертикали
Height	Высота объекта
Width	Ширина объекта
BorderColor	Цвет фона
ForeColor	Цвет шрифта
BorderColor	Цвет границ
FillStyle	Стиль заполнения
MousePointer	Вид курсора при наведении на объект

Свойства, используемые для управления формой

Свойство	Назначение
MinButton	Наличие кнопки минимизации окна
MaxButton	Наличие кнопки максимизации окна
KeyPreview	Определяет, вызываются ли процедуры обработки события клавиатуры формы перед событиями клавиатуры элементов управления
Left Top	Определяют местоположение формы по отношению к левому верхнему углу экрана монитора
Icon	Устанавливает отображаемый при сворачивании формы значок
WindowState	Состояние окна после загрузки приложения

Auto Redraw	Определяет возможность автоматического перерисования
ClipControls	Определяет необходимость перерисовки всего объекта или появляющейся части
ControlBox	Определяет наличие кнопки системного меню на форме
DrawWidth	Определяет ширину рисуемой линии (точки)
ScaleHeight, ScaleWidth	Определяет число единиц измерения по вертикали и горизонтали
ScaleLeft, ScaleTop	Определяет координаты верхнего левого угла

Основные объекты управления и их специфичные свойства

Пиктограмма	Назначение	Специфические свойства
	Командная кнопка (CommandButton)	<p>Default при значении True командная кнопка определена как кнопка по умолчанию, т.е. при нажатии Enter она будет нажата.</p> <p>Cancel определяет как кнопку отмены по умолчанию т.е. при нажатии Esc она будет нажата.</p> <p>Style стиль, принимает два значения: стандартный и графический</p> <p>Если стиль Graphical, то можно менять свойства:</p> <p>Picture картинка</p> <p>DownPicture- картинка внизу</p> <p>DisabledPicture- картинка если у кнопки нет доступа</p>

	<p>Текстовое окно (Text-Box) – экранная область, в которое можно вводить текст</p>	<p>MaxLength максимальная длина, если значение нуль, то можно вводить любое кол-во символов Multiline значение False запрещает ввод более одной строки, значение True – разрешает ввод нескольких строк после нажатия Enter ScrollBars наличие (1, 2, 3) или отсутствие (0) линеек прокруток в текстовом поле Text текст, отображаемый в поле. Locked блокировка редактирования Следующие свойства доступны в режиме выполнения SelStart число, указывающее место вставки в строке текста SelLength количество выделяемых символов SelText определяет выделенный текст</p>
	<p>Метка (Label) – применяется для отображения текста, который пользователь не может редактировать</p>	<p>Alignment выравнивание AutoSize автоподстройка размера. При значении True размер метки подгоняется под размер текста, заданный свойством Caption. Если значение False метка сохраняет размер, установленный при проектировании WordWrap перенос слов BorderStyle стиль границ</p>
	<p>Переключатель (OptionButton) для организации выбора из нескольких возможностей. Выбор одного сбрасывает все другие переключатели.</p>	<p>Value показывает выбран переключатель или нет Style стиль, изменение вида переключателя Если стиль Graphical, то можно менять свойства: Picture картинка DownPicture- картинка внизу DisabledPicture- картинка если у переключателя нет доступа</p>
	<p>Флажок (Check Box) - для организации выбора типа да/нет. Работают независимо друг от друга, пользователь может установить любое их число одновременно.</p>	

	<p>Рамка (Frame) для объединения объектов в группы</p>	
 	<p>Линейки прокрутки (Scroll bar) горизонтальная и вертикальная действуют совершенно одинаково. Эти объекты позволяют узнавать о позиции движка (scroll-box), кроме того контролировать диапазон действия линейки прокрутки и дискретность перемещения движка</p>	<p>LargeChange определяет величину, которая добавляется или вычитается из значения Value при щелчке внутри линейки прокрутки Max число, определяющее крайнюю правую или нижнюю позицию Min число, определяющее крайнюю левую или верхнюю позицию SmallChange -определяет величину, которая добавляется или вычитается из значения Value при щелчке на одной из стрелок на концах линейки прокрутки Value число, которое отражает текущую позицию движка на линейке</p>
	<p>Таймер (Timer) – это объект, способный инициировать события через регулярные промежутки времени</p>	<p>Interval число (от 0 до 65535), определяющее интервал времени в мс между двумя событиями. Интервал, равный нулю, отключает таймер</p>
	<p>Линия (Line) –для вычерчивания линий на поверхности формы. Не поддерживает никаких событий.</p>	<p>X1, Y1 координаты левого края линии X2, Y2 координаты правого края линии BorderWidth толщина линии BorderStyle стиль линии</p>
	<p>Список (ListBox)-предоставляет список возможных вариантов выбора, позволяет ограничить ввод элементами списка</p>	<p>Style стиль списка Sorted сортировка элементов списка List позволяет заполнить список на стадии проектирования ListIndex определяет положение выбранного элемента списка</p>
	<p>Комбинированный список (ComboBox) Совмещает возможности списка и текстового окна, содержит редактируемое поле</p>	<p>ListCount определяет количество элементов списка Text элемент списка или строка, введенная пользователем</p>

	<p>Окно рисунка (Picture-Box)- для размещения графической информации в определенных участках формы. Требуют больше памяти и времени на обработку, больше подходят для динамических объектов. Может выполнять функции контейнера для других элементов управления.</p>	<p>Picture позволяет выводить растровую картинку (.bmp), либо значок (.icon) AutoSize автоподстройка размера</p>
	<p>Изображение (Image) – для размещения графической информации в определенных участках формы. Удобно использовать в статической среде (не предполагается изменение)</p>	<p>Picture позволяет выводить растровую картинку (.bmp), метафайл, файлы JPEG или GIF либо значок (.icon) Stretch Растягивать. Если значение True картинка подгоняется под размер элемента управления.</p>
	<p>Контур или фигура (Shape) для вычерчивания контуров в виде прямоугольника, окружности, овала, квадрата, прямоугольника, квадрата с закругленными углами</p>	<p>Shape тип контура FillStyle стиль заполнения BorderStyle стиль границ контура BorderWidth толщина контура</p>
	<p>Список файлов (FileListBox) позволяет узнать, какие есть файлы на дисках системы и выбрать один из них</p>	<p>Pattern определение шаблона для списка файлов Следующие свойства определяют тип отображаемых файлов Archive архивный System системный Hidden скрытый ReadOnly только для чтения</p>
	<p>Список каталогов (DirListBox) позволяет узнать, какие есть каталоги на дисках системы и выбрать один из них</p>	<p>Path позволяет установить или получить текущий каталог</p>

Наименование объектов Visual Basic

При изменении имени (Name) объектов Visual Basic рекомендуется использовать следующую простую схему:

- начинать название с трехбуквенного префикса;
- использовать только буквы, цифры и знак подчеркивания (_);
- использовать не более 40 символов.

Создатели Visual Basic рекомендуют начинать название с трехбуквенного префикса в соответствии с типом объекта. Например, у вас может быть командная кнопка с названием `cmdCancel` и форма `frmMain`. Рекомендуемые префиксы перечислены в табл.

Объект	Рекомендуемый префикс
Форма	Frm
Флажок	Chk
Комбинированное окно	Cbo
Командная кнопка	Cmd
Окно данных	Dat
Список каталогов	Dir
Список дисков	Dsk
Рамка	Fra
Сетка	Grd
Горизонтальная линейка прокрутки	Hsb
Изображение	Img
Метка	Lbl
Линия	Lin
Список	Lst
Меню	Mnu
Переключатель	Opt
Окно рисунка	Pic
Фигура	Shp
Текстовое окно	Txt
Таймер	Tmr
Вертикальная линейка прокрутки	Vsb

События

Событием называется характеристика класса объектов, описывающая внешнее воздействие, на которое реагирует объект этого класса во время работы приложения.

Программы на Visual Basic управляются событиями, другими словами – действия пользователя вызывают выполнение различных процедур. Работает это примерно так: программа ждет, пока пользователь не сделает что-либо,

т.е. пока не произойдет событие; затем программа реагирует на это событие, запуская соответствующую процедуру или процедуры, затем программа снова терпеливо ждет следующего события.

События, возникающие при работе с мышью

Событие	Описание	Параметры событий
Click	Щелчок	
DblClick	Двойной щелчок	
MouseDown	Кнопка мыши нажата	Shift – определяет статус клавиш Shift(1), Ctrl(2), Alt(4) ни одна кнопка не нажата - 0 Button – определяет статус нажатой кнопки (левая - 1, правая - 2, средняя-4,) X,Y – позиция указателя курсора
MouseUp	Нажатая кнопка мыши отпущена	
MouseMove	Мышь перемещается из своей текущей позиции	
DragDrop	Завершение перетаскивания	Source – ссылка на объект, который был перемещен, X,Y – позиция курсора
DragOver	«Буксируемый» объект попадает в область другого объекта	Stale – принимает значения 0 –область занята, 1–область свободна

События, возникающие при работе с клавиатурой

Событие	Описание	Параметры событий
KeyPress	Нажата клавиша, соответствующая символу ASCII	KeyAscii- значение ASCII-кода нажатой клавиши
KeyDown	Нажата любая клавиша на клавиатуре	KeyCode – указывает нажатую клавишу
KeyUp	Отпущена любая клавиша	Shift – определяет статус клавиш Shift(1), Ctrl(2), Alt(4)

События, связанные с фокусом

Объект, имеющий фокус, может получать вводимую пользователем информацию с помощью мыши и клавиатуры.

Событие	Описание
GotFocus	Получении фокуса
LostFocus	Потеря фокуса

События, специфичные для объектов

Событие	Описание	Для какого объекта характерны
Load UnLoad	Загрузка Закрывать форму	Форма
Resize	Изменение размера	Форма, Картинка
Change	Изменение	Текстовое окно, Полосы прокрутки, Список каталогов Комбинированный список,
Scroll	Прокрутка	Полосы прокрутки, Список, Комбинированный список, Список файлов, Список каталогов
Timer	Истечение интервала времени	Таймер

Окно кода процедуры обработки события

Любой объект можно связать с набором процедур, исполняемых в строго определенных моменты. Процедура (procedure) – это группа операторов языка. Исполняется процедура – исполняются ее операторы. Так или иначе, весь составленный вами исполняемый код обязательно помещается в какую-нибудь процедуру. Процедура, присвоенная объекту, связана с определенным событием и поэтому называется процедурой обработки события. Важно отметить, что с одним объектом могут быть связаны несколько событий.

Имя процедуры обработки события для элемента управления составляется из имени элемента управления (Name), знака подчеркивания (_) и имени события.

Для открытия окна кода процедуры (рис.1.2) существует три способа:

1. двойной щелчок по объекту
2. клавиша F7
3. в меню выбрать View / Code

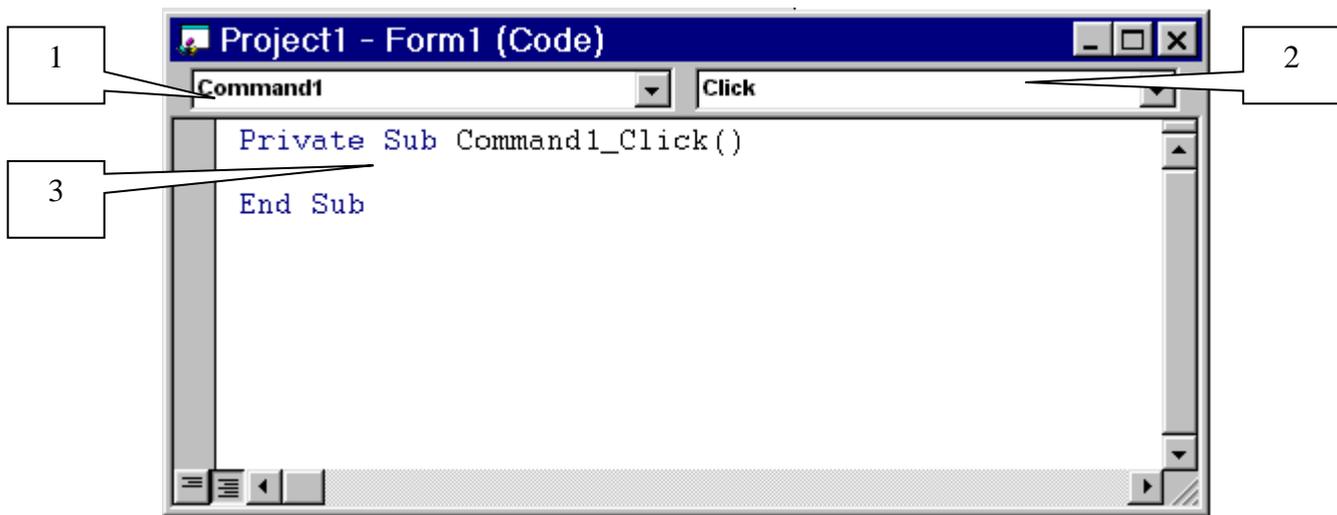


Рисунок 1.2 - Окно кода процедуры обработки события
 1 - список объектов формы; 2 - список событий объекта; 3-процедура

Процедуры по умолчанию не делают ничего; они состоят лишь из объявления процедуры (Sub) и оператора, помечающего конец процедуры (End Sub).

Программный код вводится между строками Private Sub и End Sub.

Понятие метода в Visual Basic

Название метода	Элементы управления, для которых используются методы	Описание действия
AddItem	ListBox, ComboBox	Добавление элемента в список
Circle	Form, PictureBox	Рисование дуги, эллипса или окружности
Clear	ListBox, ComboBox	Удаление всех элементов списка
Cls	Form, PictureBox	Очистка от графических элементов или текста
Drag	Все, кроме Line, Menu, Shape, Timer	Перемещение объекта
Hide	Form	Спрятать форму с экрана без ее выгрузки
Line	Form, PictureBox	Рисование линии или прямоугольника
Move	Все, кроме Timer и Menu	Перемещение элемента
Point	Form, PictureBox	RGB (red-green-blue) цвет точки
PopupMenu	Form	Вывод всплывающего меню в заданной точке формы
Print	Form, PictureBox.	Печать строки на объекте

Visual Basic предоставляет большое количество встроенных процедур и функций (функции – это особый класс процедур, которые возвращают значения аргументов). Эти процедуры и функции описаны в справочной системе или документации на систему программирования.

Pset	Form, PictureBox.	Точка на объекте
Refresh	Все	Немедленная перерисовка на экране
RemoveItem	ListBox, ComboBox	Удаление элемента списка или ячеек в сетке
Scale	Form, PictureBox.	Координаты объекта
SetFocus	CheckBox, ComboBox, CommandButton, FileListBox, Form, HscrollBar, ListBox, OptionButton. PictureBox, TextBox. VScrollBar	Установка курсора
Show	Form	Вывод формы на экран
ZOrder	Все	Расположение на переднем или заднем плане

Процедуры и функции уже не одно десятилетие являются неотъемлемой частью любого языка программирования. Лишь сравнительно недавно, когда появились языки, поддерживающие объектно-ориентированное программирование, были введены особые типы процедур и функций, названные методами. Методы «работают» как процедуры и функции, но принадлежат конкретным объектам, так же как и свойства. Методы объекта определяют те задачи, которые может выполнить данный объект.

Чтобы вызвать метод, надо указать имя объекта и через точку имя метода.

Некоторые методы предусмотрены практически для всех типов объектов, другие же более специфичны. В таблице приводится список основных методов, элементы управления, для которых они определены, и краткое описание действия.

Проект приложения

Программный проект – это совокупность частей, составляющих будущее Windows-приложение. Любой проект должен состоять из одной или нескольких экранных форм и одного или нескольких программных модулей.

Структура проекта

При создании приложения в Visual Basic программист работает с проектами. Проект (project) - это набор файлов, используемых для построения приложения. В этот набор входят:

Файл формы(для каждой формы) – содержит описание формы и ее элементов управления, включая установленные значения свойств, а также

используемые в форме процедуры обработки событий. Файл формы имеет расширение .FRM

Стандартные модули хранятся в файлах с расширением .BAS (они могут использоваться при решении не одной, а нескольких задач и в данном пособии не рассматриваются, как и некоторые другие)

Файл проекта – содержит сведения о размещении всех других файлов проекта, имеет расширение .VBP.

Все файлы проекта могут быть объединены в один исполняемый файл (с расширением .EXE).

Запуск проекта

Для запуска проекта используйте один из приведенных способов:

- с помощью меню выбрать Run / Start
- нажать клавишу F5
- нажать кнопку запуска на панели инструментов

Сохранение проекта

Для сохранения следует выполнить File / Save Project или нажать кнопку  на панели инструментов. Visual Basic сначала предложит сохранить форму, а затем проект.

В появившемся диалоговом окне :

- установите диск и каталог назначения;
- измените имя файла формы;
- нажмите кнопку Сохранить
- установите диск и каталог назначения;
- измените имя файла проекта;
- нажмите кнопку Сохранить

Открытие проекта

Для открытия существующего проекта следует выполнить File / Open Project или нажать на кнопке панели инструментов. В появившемся диалоговом окне:

- установите диск и каталог назначения;
- выберите имя файла проекта;
- нажмите Открыть

Компиляция проекта

Когда вы закончите разработку проекта, вы можете скомпилировать его, чтобы создать стандартный файл .EXE прикладной программы Windows. Эту программу можно будет запускать самостоятельно, без Visual Basic. Чтобы скомпилировать проект:

- выберите File, Make EXE File
- поменяйте, если надо, диск и папку назначения
- в строке File Name введите требуемое имя, щелкните на Ok.

Важное замечание: чтобы запустить вашу прикладную программу на компьютере, где не установлен Visual Basic , сначала надо скопировать файл VB5.dll в каталог Windows /System. Если ваш проект использует файлы специальных средств управления .OCX или .VBX, их тоже надо скопировать в каталог Windows /System.

1.3 Данные, Выражения, Функции

Данные в Visual Basic

Любая программа выполняет обработку данных. Данные могут быть постоянными величинами (константы) и переменными величинами. Переменные и постоянные величины должны быть отнесены к какому-либо типу данных. В Visual Basic существует набор стандартных типов данных, кроме того можно объявить свой собственный тип данных.

Типы данных

Типом данных называется способ хранения и представления данных в компьютерной системе. Стандартные типы данных, поддерживаемые языком Visual Basic , приведены в таблице.

Тип данных	Диапазон значений	Объем занимаемой памяти
Integer (целое)	от -32768 до 32767	2 байта
Long (длинное целое)	от -2147483648 до 2147483647	4 байта
Byte (короткое целое число)	от 0 до 255	1 байт
Boolean (булево)	False, True	2 байта
Single (десятичное обычной точности)	от 1,41 E-45 до 3,4 E+38 (по модулю)	4 байта
Double (десятичное двойной точности)	от 4,95 E-324 до 1,79 E+308 (по модулю)	8 байт
String (строка)	Строка любых символов	1 байт на каждый символ
Object (объект)	Ссылка на объект	4 байта
Variant (произвольный)		16 байт (числ. тип) 22 байта +длина строки (строковый тип)
Date (дата)	Значения даты и времени диапазон от 01 янв. 100 г до 31 дек. 9999г.	8 байт
Currency	Значением является	8 байт

Десятичное протяженное	число до 15 знаков до десят.точки до 4 знаков после точки.	
------------------------	--	--

Константы

Константы – это такие величины, которые в процессе выполнения программы не изменяют своих значений. Visual Basic имеет целый ряд внутренних (системных) констант, которые имеют префикс **vb**. В качестве примера - *Таблица констант дней недели*

Имя константы	Значение
VbSunday	1
VbMonday	2
VbTuesday	3
VbWednesday	4
VbThursday	5
VbFriday	6
VbSaturday	7

Константы, определяющие цвет, смотрите в приложении *Учебного пособия часть 1*

Константа **VbTab** делает промежутки при выводе информации.

Подробно константы системы Visual Basic перечислены в библиотеках объектов VB и VBA в окне **Object Browser** (просмотр объектов)

Переменные

Переменные величины при выполнении программы могут свои значения изменять. Переменная имеет имя. Правила формирования имени переменной:

- имя начинается с буквы; остальные символы – буквы и цифры, можно использовать знак подчеркивания «_»;
- имя не должно содержать точки;
- длина имени не более 255 символов;
- имя должно быть уникальным в пределах данной процедуры;
- выбирать имя лучше так, чтобы оно несло информацию о назначении переменной;
- имя не должно быть *ключевым словом* Visual Basic.

Visual Basic не требует обязательного объявления переменной перед ее использованием. Если переменная не объявлена, Visual Basic использует тип данных, заданный по умолчанию - Variant. Этот тип может содержать любую информацию. Частое использование этого типа для хранения информации имеет два существенных недостатка – влечет лишнюю трату ресурсов памяти и может привести к непредсказуемому поведению заданных по умолчанию значений. Поэтому все же лучше объявлять переменные перед их использованием.

Объявление типа переменной означает команду установить границы значений переменной, которые определяются ее типом. Команда записывает-

ся с помощью программного кода. Существует несколько способов объявления типа переменной.

1 способ. Объявление типа с помощью **суффикса**

К имени переменной приписывается один из следующих суффиксов:

Суффикс	%	&	!	#	\$	@
Тип	Integer	Long	Single	Double	String	Currency

2 способ. Объявление типа с помощью **оператора описания типа**

Dim *ИмяПеременной* **As** *ТипПеременной*

Dim, As- это ключевые слова Visual Basic.

При объявлении переменных следует учитывать следующее: переменная, объявленная в процедуре, инициализируется каждый раз, когда происходит вызов этой процедуры. Чтобы переменная сохраняла свое старое значение при повторном обращении к процедуре, при ее описании надо использовать ключевое слово **Static** (статическая):

Static *ИмяПеременной* **As** *ТипПеременной*

Область видимости переменных

Переменные могут быть объявлены в модулях, формах или процедурах.

– Переменная, объявленная в процедуре, доступна для использования только в пределах этой процедуры (*локальная*), даже если она была объявлена как **Static**;

– Переменная, объявленная в **General Declarations** (раздел общих объявлений) формы доступна в пределах всей формы.

– Область видимости переменных, объявленных на уровне модуля, зависит от ключевых слов, использованных при объявлении.

– Если переменная объявлена посредством **Dim**, область видимости – модуль.

– Если переменная объявлена посредством ключевого слова **Public**, область ее видимости – вся программа (переменная - *глобальная*).

Присвоение значения переменной

Для присвоения переменной некоторого значения используется *оператор присвоения*.

Общий вид оператора присвоения:

[**Let**] *ИмяПеременной* = *ЗначениеПеременной*

Ключевое слово **Let** в квадратных скобках может отсутствовать.

ЗначениеПеременной – это либо константа или выражение, значение которого должно быть вычислено. В том и другом случае это значение должно находиться в диапазоне возможных значений для данного типа переменной.

Выражения и функции

Арифметическое выражение

Арифметическое выражение – это последовательность чисел, констант, переменных, числовых функций, которые соединены между собой знаками арифметических действий (см. табл.).

Операция	Описание операции
$A \wedge B$	Возведение A в степень B
- A	Перемена знака A
$A * B$	Умножение A на B
A / B	Деление A на B
$A \setminus B$	Целая часть от деления A на B
$A \bmod B$	Остаток от деления A на B
$A + B$	Сложение A с B
$A - B$	Вычитание B из A

Логическое выражение

Логическое выражение – это последовательность символьных, числовых констант и переменных, математических и строковых функций, которые соединены между собой знаками сравнения и знаками логических операций (см. табл.). Логическое выражение принимает только два значения: истина (True) и ложь (False).

Знак сравнения	Действие
=	Равно
<	Меньше
>	Больше
<=	Меньше или равно
>=	Больше или равно
<>	Не равно

Знак логической операции	Действие
And	Логическое «И»
Eqv	Эквивалентность
Imp	Импликация
Or	Логическое «ИЛИ»
Xor	Исключающее «ИЛИ»
Not	Логическое «НЕ»

Строковое выражение

Строковое выражение – это последовательность символьных констант, переменных, символьных функций и знака присоединения «+» (конкатенации)

Функция – это процедура, которая выполняет и возвращает значение. Общий вид обращения к функции:

ИмяФункции (СписокАргументовФункции)

ИмяФункции - это либо имя уже имеющейся в языке (встроенной) функции, либо имя функции, *определяемой* программистом.

АргументФункции – это либо константа, либо переменная, либо выражение.

Встроенные математические функции

Функция	Значение функции
<i>Abs (x)</i>	Абсолютная величина числа x
<i>Cint (x)</i>	Целое число, ближайшее к числу x
<i>Fix (x)</i>	Целое число, равное числу x без дробной части
<i>Int (x)</i>	Наибольшее целое число, не превышающее x
<i>Sqr (x)</i>	Квадратный корень из x
<i>Sin (x)</i>	Синус
<i>Cos (x)</i>	Косинус
<i>Tan (x)</i>	Тангенс
<i>Atn (x)</i>	Арктангенс
<i>Rnd</i>	Псевдослучайное число от 0 до 1

Финансовые функции

Функция	Значение функции
<i>Pmt</i> (проц.ставка, число выплат, кредит)	Периодическая выплата
<i>PV</i> (проц.ставка, число выплат, выплата)	Кредит
<i>Rate</i> (число выплат, выплата, кредит)	Процентная ставка, отнесенная на период выплаты

Функции обработки строк

Функция	Значение функции
<i>InStr</i> (строка, подстрока)	Позиция подстроки в строке
<i>Lcase</i> (строка)	Строка со строчными буквами
<i>Left</i> (строка, длина)	Левая подстрока данной длины

<i>Len</i> (строка)	Длина строки
<i>Mid</i> (строка, позиция, длина)	Подстрока данной длины, начиная с данной позиции
<i>Right</i> (строка, длина)	Правая подстрока данной длины
<i>Trim</i> (строка)	Строка без первых и последних пробелов
<i>Ucase</i> (строка)	Строка с прописными буквами

Функции даты и времени

Функция	Значение функции
<i>Time</i>	Текущее системное время
<i>Timer</i>	Число прошедших с полуночи секунд
<i>Now</i>	Системная дата и время
<i>Date</i>	Системная дата
<i>Month</i> (дата)	Номер месяца
<i>Year</i> (дата)	Год
<i>Day</i>	День
<i>WeekDay</i> (дата)	Номер дня недели (воскресенье – 1 и т.д)

Системные функции

Функция	Значение функции
<p><i>InputBox</i> (Приглашение[, Заголовок] [, НачЗначение] [,X] [,Y])</p> <p>Предназначена для ввода данных пользователем через системное окно X,Y – координаты левого верхнего угла окна</p>	Введенный пользователем текст
<p><i>MsgBox</i> (Сообщение [, Опция, Заголовок])</p> <p>Предназначена для вывод сообщения через системное окно</p> <p><i>Опция</i> определяет набор кнопок и вид пиктограммы в окне сообщения</p> <p>0 Ок 1 Ок, Отмена 2 Стоп, Повтор, Пропустить 3 Да, Нет, Отмена 4 Да, Нет 5 Повтор, Отмена</p>	<p>Целое число, определяющее какая из кнопок <i>Окна сообщения</i> была нажата</p> <p>1 (VbOk) Ок 2 (VbCancel) Отмена 3 (VbAbort) Стоп 4 (VbRetry) Повтор 5 (VbIgnore) Пропустить 6 (VbYes) Да 7 (VbNo) Нет</p>

16	критическое сообщение	
32	вопрос	
48	предупреждение	
64	информация	

Функции преобразования типов

Функция	Значение функции
<i>Asc</i> (строка)	Код ASCII первого символа строки
<i>Chr</i> (код ASCII)	Символ, соответствующий коду
<i>CInt</i> (строка цифр или число)	Целое число
<i>Val</i> (строка)	Число
<i>Str</i> (число)	Строка
<i>CDate</i> (выражение)	Выражение типа Date
<i>CBool</i> (выражение)	Выражение типа Boolean
<i>CByte</i> (выражение)	Выражение типа Byte
<i>CCur</i> (выражение)	Выражение типа Currency
<i>CDbl</i> (выражение)	Выражение типа Double
<i>CInt</i> (выражение)	Выражение типа Integer
<i>CLng</i> (выражение)	Выражение типа Long

Функции для работы с графикой

Функция	Значение функции
<i>LoadPicture</i> (строка-путь к графическому файлу)	Содержимое графического файла
<i>QBColor</i> (число от 0 до 15)	Цвет
<i>RGB</i> (N1, N2, N3) N1 – интенсивность красного цвета (от 0 до 255) N2 - интенсивность зеленого цвета (от 0 до 255) N3 - интенсивность синего цвета (от 0 до 255)	Цвет

1.4 Выбор в программе, организация циклов

Выбор в программе выполняют условные операторы. Условные операторы помогают осуществить «ветвление» программы, т.е. передать управление по условию, на ту или иную «ветку» - это может быть фрагмент текста, процедура, подпрограмма или даже другой модуль.

Виды условных операторов Visual Basic:

If ... Then ... Else ...

Select ... End Select

Ветвление If ... Then ... Else

Этот оператор обычно используется, когда проверяется одно или два условия в программе. Конструкцию If ... Then ... Else ... (Если ... Тогда ... Иначе ...) можно записать в одном из двух форматов - блочном (в несколько строк) и линейном (в одной строке).

Линейный формат:

If логич_выражение **Then** блок_команд_1 [**Else** блок_команд_2]

Если логич_выражение принимает значение «истина», выполняется блок_команд_1, иначе выполняется блок_команд_2. **Else** можно опустить, в этом случае, если логич_выражение ложно, блок_команд_1 просто пропускается (рис. 1.3).

Пример 1:

Если значение переменной А больше нуля, то вычислить A^2 , иначе - вычислить A^3

If A>0 Then S= A^2 Else S=A^3

Пример 2:

Если значения переменных А и В равны между собой, то прервать выполнение процедуры

If A=B Then Exit Sub

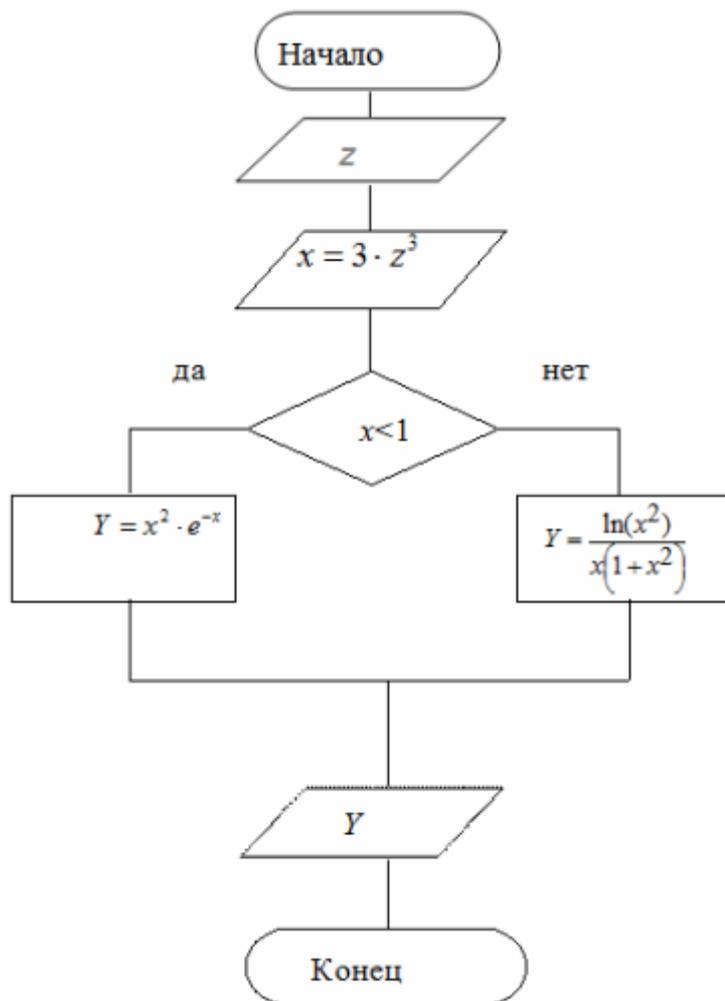


Рисунок 1.3 - Блок-схема организации условного оператора

Блочный формат:

```
If логич_выражение_1 Then  
    блок_команд_1  
[ElseIf логич_выражение_2 Then  
    блок_команд_2  
.....  
[Else  
    блок_команд_n  
End If
```

Если логич_выражение_1 принимает значение «истина», выполняется блок_команд_1, иначе, если логич_выражение_2 истинно, выполняется блок_команд_2 и т.д. Если ни одно условие не удовлетворяется, то выполняется блок_команд_n, следующий за **Else**.

Пример 1:

Если значения переменных А и В положительны, то вывести их сумму и произведение, в противном случае - вывести сообщение об ошибке.

```
If A>0 And B>0 Then  
    Print A+B  
    Print A*B  
Else  
    Print «ошибка»  
End If
```

Пример 2:

Если значение переменной В меньше 10, то увеличить его в 2 раза, если значение от 10 до 20, то увеличить в 3 раза, в остальных случаях - уменьшить в 10 раз.

```
If B<10 Then  
    B=B*2  
ElseIf B>=10 And B<=20 Then  
    B=B*3  
Else  
    B=B/10  
End If
```

Оператор множественного выбора Select Case

Используется при необходимости осуществить проверку более сложных условий.

Формат команды

```
Select Case арифм_выражение или симв_выражение
```

Case условие 1

блок команд 1

Case условие 2

блок команд 2

.....

Case Else

блок команд n

End Select

В поле операндов **Select Case** записывается произвольное арифметическое выражение или символьное выражение, которое в процессе выполнения программы принимает то или иное числовое, логическое или символьное значение.

В поле операндов каждого оператора **Case** надо указать условие в одном из трех форматов:

1. **Case** константа_1, константа_2,...
2. **Case Is** знак_отношения константа
3. **Case** константа_1 **To** константа_2

Алгоритм множественного выбора заключается в следующем. Сначала вычисляется значение выражения, записанного в **Select Case**. Далее проверяется, удовлетворяет ли это значение одному из указанных в **Case** условий. Если значение удовлетворяет какому-то условию, выполняется блок команд, следующий за данным **Case**. Если ни одно условие не удовлетворяется, выполняется блок команд, следующий за **Case Else**. При выполнении того или иного блока команд управление передается команде, следующей за **End Select**.

Пример:

Input A

Select Case A

Case 1,5

Print «A равно 1 или 5»

Case Is >5

Print «A больше 5»

Case -8 TO 2.5

Print «A не меньше -8, но не больше 2.5»

Case Else

Print «Ни одно условие не выполняется»

End Select

Цикл - это многократно повторяемая часть программы.

В языке Visual Basic предусмотрено два основных способа организации циклов:

– повторение блока команд заданное количество раз (*цикл со счетчиком*);

– циклическое повторение блока команд, пока выполняется (или не выполняется) некоторое условие.

Цикл со счетчиком For ...Next (Для ... Следующий)

Цикл For ... Next - это цикл с заранее заданным количеством повторений.

```
For переменная_цикла=нач_значение To конеч_значение [Step шаг]  
блок_команд  
Next переменная_цикла
```

В поле операндов оператора For указываются:

– числовая переменная, которая называется *счетчиком* или *переменной цикла*.

– начальное и конечное значение счетчика.

Если необходимо, укажите шаг изменения счетчика по окончании каждого цикла (по умолчанию этот шаг равен 1).

Конструкция завершается оператором Next.

Принцип работы оператора:

Переменной цикла присваивается начальное значение, после этого первый раз выполняется блок команд (тело цикла). Оператор NEXT увеличивает текущее значение переменной цикла на величину шага, и, если новое значение переменной цикла не превышает заданное конечное значение, в очередной раз выполняется блок команд.

Пример 1: Вывести 10 раз текст «привет»

```
For I=1 To 10  
Print «привет»  
Next I
```

Пример 2: Вывести все четные натуральные числа от 10 до 20.

```
For K=10 To 20 Step 2  
Print K  
Next K
```

Можно выйти из цикла не дожидаясь выполнения всех повторений при помощи оператора **Exit For**. Управление будет передано на оператор, стоящий после **Next**.

С помощью For ... Next можно организовывать вложенные циклы - каждый со своим For, Next и счетчиком.

Универсальный цикл Do ...Loop (Делать ... Цикл)

Наиболее гибкий и универсальный способ организации цикла по условию обеспечивает конструкция Do ...Loop. Конструкция имеет четыре формата:

Циклы с предусловием.

1. **Do While** логич_выражение
блок_команд
Loop
(рис.1.4)

Блок_команд выполняется до тех пор, пока значение логич_выражения **ИСТИННО**.

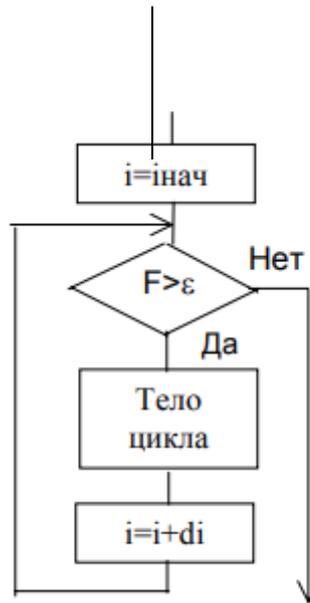


Рисунок 1.4 – Блок-схема алгоритма с предусловием

2. **Do Until** логич_выражение
блок_команд
Loop

Циклы с постусловием

При первом входе условие не проверяется, поэтому блок_команд будет выполнен хотя бы один раз (рис. 1.5).

3. **Do**
блок_команд
Loop While логич_выражение

Блок_команд выполняется до тех пор, пока значение логич_выражения **ИСТИННО**.

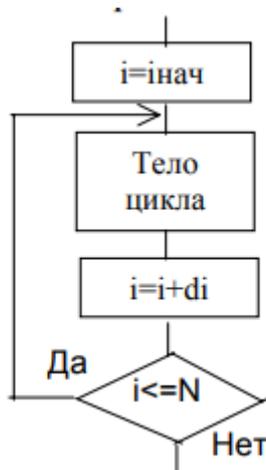


Рисунок 1.5 – Блок-схема алгоритма с постусловием

4. **.Do**

блок_команд

Loop Until логич_выражение

Блок_команд выполняется до тех пор, пока значение логич_выражения **ЛОЖНО**.

Можно выйти из цикла не дожидаясь выполнения всех повторений при помощи оператора **Exit Do**. Управление будет передано на оператор, стоящий после **Loop**.

Решение нелинейного уравнения методом половинного деления

Целью данной работы является приобретение навыков решения задач итерационными методами, в частности решение нелинейных уравнений

методом половинного деления, используя конструкцию DO WHILE...LOOP.

Метод половинного деления (или дихотомии)

При решении нелинейного уравнения методом половинного деления задаются интервал $[a, b]$, на котором существует только одно решение, и желаемая точность ε (рис. 1.6).

Постановка задачи.

Пусть функция $y=f(x)$ определена и непрерывна на отрезке $[a; b]$.

$$f(a) \cdot f(b) < 0$$

Требуется найти корень на отрезке $[a; b]$ с точностью 0,00001 .

Последовательность решения задачи.

Для реализации алгоритма дихотомии отрезок $[a; b]$ делится пополам точкой $c = (a + b)/2$ (рис. 1).

Если $f(c)$ не равно 0, то возможны два случая:

а) $f(x)$ меняет знак на отрезке $[a; c]$;

б) $f(x)$ меняет знак на отрезке $[c; b]$.

Выбираем тот отрезок, на котором функция меняет знак. Если $f(x)$

меняет знак на отрезке $[a; c]$, то $b=c$; если $f(x)$ меняет знак на отрезке $[c;b]$, то $a=c$.

Деление отрезка пополам продолжается пока $|b-a| > \epsilon$.

Корень уравнения: $x = (a + b)/2$

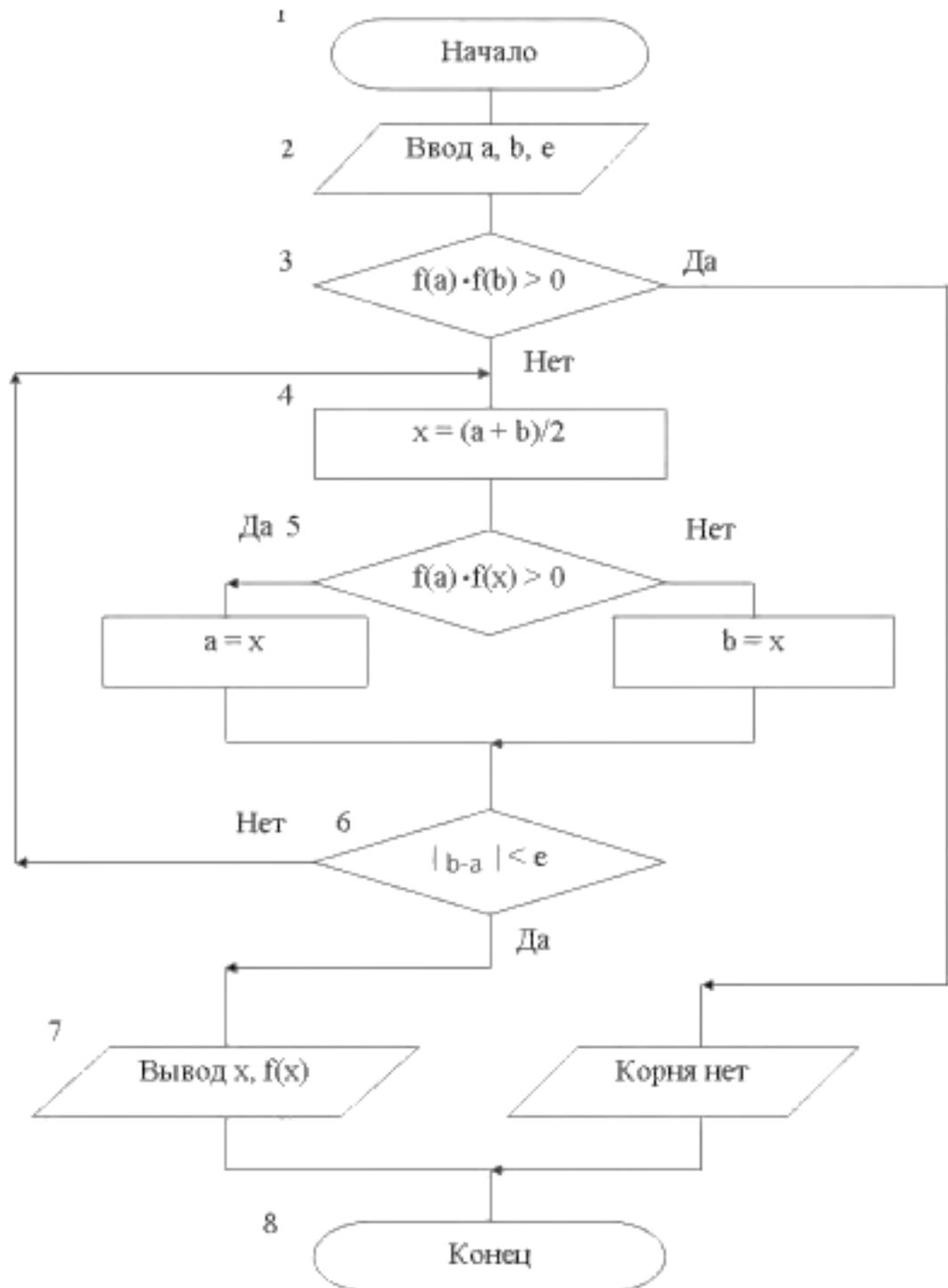


Рисунок 1.6 – Блок-схема алгоритма решения нелинейного уравнения методом половинного деления

Алгоритм решения нелинейного уравнения методом половинного деления

```
Private Sub Command1_Click()  
Dim a As Single, b As Single, c As Single, eps As Single  
Dim result As Single  
a = Val(Text1.Text)  
b = Val(Text2.Text)  
eps = Val(Text3.Text)  
If y(a) * y(b) < 0 Then  
c = (a + b) / 2  
Do While Abs(b - a) > eps  
    If y(a) * y(c) < 0 Then  
        b = c  
    ElseIf y(b) * y(c) < 0 Then  
        a = c  
    Else  
        a=c  
        b=c  
    End If  
    'Debug.Print a, b, c  
    c = (a + b) / 2  
Loop  
result = y(c)  
Text4.Text = Format(c, "0.000")  
Text5.Text = result  
Else  
Text4.Print "Корня нет"  
End If  
End Sub  
  
Private Function y(x As Single) As Single  
y = 10 * Sin(x + Exp(1)) / Exp(Abs(0.2 * x)) + 1  
End Function
```

Результат решения нелинейного уравнения методом половинного деления отображен на форме и представлен на рисунке 1.7.

Значение начала интервала: a	<input type="text" value="2"/>
Значение конца интервала: b	<input type="text" value="4"/>
Заданная точность: eps	<input type="text" value="0.001"/>
Значение корня уравнения	<input type="text"/>
Значение функции в этой точке	<input type="text"/>
<input type="button" value="Вычислить"/>	

Рисунок 1. 7 - Результат решения нелинейного уравнения методом половинного деления

1.5 Программирование графики

На форме или в графическом поле (PictureBox) можно рисовать различные графические примитивы с использованием графических методов:

Scale – позволяет задать систему координат и масштаб для формы или графического окна:

Object.Scale (X1,Y1) - (X2,Y2)

Pset – установка точки с заданными координатами и цветом:

Object.Pset (X,Y) [,Color]

PSet (x, y), Color-рисует точку

X и Y -это координаты точки,

Color-цвет

Пример:

1. Me.DrawWidth = 100 'Размер точки

PSet (1000, 1000), vbRed (Рисуется красная точка)

2. PSet (1000, 2000)(Рисуется точка с координатами x=1000, y=2000.

Цвет точки определяется свойством ForeColor)

– Cls - Очищает поверхность объекта от нарисованного и напечатанного.

Line – рисование линии, прямоугольника или закрашенного прямоугольника заданного цвета:

Object.Line (X1,Y1) - (X2,Y2) [,Color][,B][F]

– Line(x1,y1)-(x2,y2),Color-рисует линию;

x1 и y1 –это начальные координаты линии,

x2 и y2 – это конечные координаты,

Color- это цвет (может задаваться функцией RGB(R,G,B), R-красный, Зеленый, B-синий. Цвета могут принимать значение от 0 до 255.)

Пример:

Line(100,150)-(250,300),RGB(255,0,0)

– Line(x1,y1)-(x2,y2),Color,B-рисует квадрат

x1 и y1 –это начальные координаты линии,

x2 и y2 – это конечные координаты,

Color- это цвет (может задаваться функцией RGB(R,G,B), R-красный, G-зеленый, B-синий. Цвета могут принимать значение от 0 до 255.)

B-параметр для рисования квадрата

Пример:

1. Line (2000, 1000)-(5000, 3000) , vbRed , B (Прямоугольник красного цвета)

2. Line (2000, 1000)-(5000, 3000) , vbRed , BF (Прямоугольник красного цвета, залитый этим же цветом)

3. Line (2000, 1000)-(5000, 3000) , , B (Прямоугольник. Цвет его определяется свойством ForeColor, так как в операторе там, где должен быть указан цвет, стоит пустота).

Circle – рисование окружности, овала или дуги с заданными координатами центра, радиусом, цветом, начальным и конечным углом дуги и коэффициентом сжатия:

Object.Circle (X,Y),Radius [,Color, Start, End, Aspect]

Circle(X,Y),R,Color- рисует окружность

Circle (x_центра , у_центра) , радиус , цвет , начальный_угол , конечный_угол , сжатие

X и Y -это центр окружности,

R -это радиус окружности

color- цвет окружности

Пример:

1.Circle(200,200),100,RGB(0,255,0)

2.Circle (4000, 2000) , 1000 (Окружность с центром в точке с координатами

(4000, 2000) и радиусом 1000)

3.Circle (4000, 2000) , 1000 , vbRed (Красная окружность)

4.Circle (4000, 2000) , 1000 , , 1 , 3 (Дуга окружности, начинающаяся от угла в 1 радиан и кончающаяся углом в 3 радиана. Угол отмеряется от направления на восток против часовой стрелки)

5.Circle (4000, 2000) , 1000 , , -1 , -3 (Сектор круга, начинающийся от угла в 1 радиан и кончающийся углом в 3 радиана. Угол отмеряется от направления на восток против часовой стрелки)

6.Circle (4000, 2000) , 1000 , , , 2(Эллипс с центром в точке с координатами (4000, 2000). Получен из окружности радиусом 1000 горизонтальным сжатием в 2 раза)

7.Circle (4000, 2000) , 1000 , , , 1/3(Эллипс с центром в точке с координатами (4000, 2000). Получен из окружности радиусом 1000 вертикальным сжатием в 3 раза)

- 8.Circle (4000, 2000) , 1000 , , 1 , 3 , 2(Дуга эллипса)
9.Circle (4000, 2000) , 1000 , , -1 , -3 , 2(Сектор эллипса).

Свойства объектов, влияющие на графические методы

AutoRedraw = True/False - в общих чертах, - это свойство разрешает/запрещает рисование. Если оно установлено в True, создаётся как-бы слой над загруженной в объект картинкой, на котором собственно и рисуют. Причём операторы не влияют на загруженную картинку.

ScaleMode - Масштаб. По умолчанию - твипы(Twips). Выбор единиц измерения большой, но по-моему, разумно использовать только ещё Поинты(Points) и Пикселы(Pixels).

DrawMode - Тип рисования, а точнее логическая операция, применяемая к существующему изображению.

DrawStyle - Стиль рисования. Вам позволяет выбрать вариант изображения линий: 0-Полностью(Solid), 1-Тире(Dash), 2-Точка(Dash), 3-Тире-точка(Dash-Dot), 4-Тире-Точка-Точка(Dash-Dot-Dot) и 5-Прозрачно(Transparent).По умолчанию Solid.

DrawWidth - Размер точки.

ForeColor - Цвет линий и текста по умолчанию.Чёрный по умолчанию

FillColor - Цвет заливки, для замкнутых фигур.Чёрный по умолчанию

FillStyle - Стиль заливки для замкнутых фигур: 0-Полностью(solid), 1-Прозрачный(Transparent), 2-Горизонтальные линии(Horizontal Line), 3-Вертикальные линии(Vertical Line), 4-По диагонали направо сверху(Upward Diagonal), 5-По Диагонали направо снизу(Downward Diagonal), 6-Крест(Cross), 7-Дигональный крест(Diagonal Cross). По умолчанию значение - 1.

Координаты при рисовании отсчитываются из левого верхнего угла

Пример:

PSet (1000, 2000) -(точка - тонкая, черная)

Circle (4000, 2000), 1000 - (окружность - тонкая, черная)

DrawWidth = 20 (меняем толщину линий)

ForeColor = RGB(230, 250, 100) (меняем цвет линий)

PSet (1000, 5000) (точка - толстая, цветная)

Circle (4000, 5000), 1000 (окружность - толстая, цветная).

1.6 Массивы данных, создание и использование вызываемых процедур и функций

Массив - упорядоченная совокупность однотипных данных, обозначенных одним именем. *Имя массива* образуется по тем же правилам, что и имена переменных.

Отдельные величины, образующие массив, называются *элементами* массива.

Каждый элемент в массиве имеет номер (индекс), по которому можно обратиться к значению элемента. *Индекс* - числовое значение целого типа.

Количество элементов в массиве называется *размером* массива. Массивы бывают:

- *одномерные*, элемент определяется одним индексом (можно сравнить с таблицей из одной строки).
- *двумерные* - два индекса определяют элемент (таблица из нескольких строк и столбцов).

Перед использованием в программе массив должен быть объявлен оператором DIM, который резервирует память.

Dim ИмяМассива (границы) [As тип]

Статические массивы

При объявлении статического массива можно указывать границы (верхнюю и нижнюю).

Пример:

DIM A(10 TO 15) одномерный массив, индексы элементов с 10 до 15

DIM B(12) одномерный массив из 13 элементов (если верхняя граница не указана, то она = 0)

DIM C(3,10) двумерный массив из 4-х строк и 11-ти столбцов

DIM N(5 TO 10, 3 TO 13) двумерный массив из 6-х строк и 11-ти столбцов

Ввод массива

Существует два основных способа заполнения массива.

1. В первом случае значения элементов последовательно вводятся с клавиатуры. На первом шаге указывается количество элементов в массиве и соответствующим образом переопределяется размер массива. На втором шаге организуется цикл, на каждом шаге которого вводится значение одного элемента. Размер массива храниться в переменной n. Для организации цикла For потребуется счетчик i.

```
Dim n, i As Integer
```

```
Dim a() As Integer
```

```
n= Val(TextBox("Введите размерность массива"))
```

```
For i = 0 To n
```

```
a(i) = Val(TextBox("Введите " + i + "-й элемент массива"))
```

```
Next i
```

2. Второй способ ввода массива – это заполнение случайными числами. В Visual Basic есть специальная функция, которая по определенным правилам генерирует рациональные случайные числа в диапазоне (0; 1). Она называется Rnd(). Используя эту функцию, можно заполнить массив случайными числами из любого диапазона. Начало и конец диапазона значений задаются с клавиатуры.

```
start = Val(TextBox("Введите начало диапазона"))
```

```
fin = Val(TextBox("Введите конец диапазона"))
```

```
Randomize()
```

```
For i = 0 To n
```

```
a(i) = Math.Round(start + (fin - start) * Rnd())
Next i
```

Вывод массива

Массивы можно выводить в окно списка или в текстовое поле.

Вывод массива в окно списка практически не отличается от вывода совокупности.

```
Сначала очищаем окно списка. lstA.Items.Clear()
```

Выводим в окно списка поясняющий текст и заголовки колонок.

```
lstA.Items.Add("Исходный массив")
```

lstA.Items.Add("Номер" + vbTab + "Значение") Константа vbTab позволяет организовать вывод в две колонки.

Организуем цикл для вывода значений элементов массива

```
For i = 0 To n
```

```
lstA.Items.Add(Str(i) + vbTab + Str(a(i)))
```

```
Next i
```

На каждом шаге цикла в окно списка выводим номер элемента – i и его значение – a(i).

Для вывода массива в текстовое поле нам потребуется специальная переменная, в которой будет формироваться выводимая строка. Так как массив выводится в текстовое поле, то эта переменная будет иметь символьный тип данных, а точнее – String. Dim s As String.

Сначала в этой строке нет никакой информации. Она пустая. Пустая строка обозначается парой кавычек, между которыми нет ни одного символа. Организуем цикл для формирования строки выводимой информации.

```
s = ""
```

```
For i = 0 To n
```

```
s = Str(a(i)) + " "
```

```
Next i
```

```
txtA.Text = s
```

Матрицы – двумерные массивы вводятся и выводятся с помощью вложенных циклов: один перебирает строки, другой – столбцы матрицы. Какой из циклов внешний – неважно, главное – вложенные циклы позволяют перечислить все сочетания номеров строк и столбцов, т.е. учесть все элементы матрицы. Рассмотрим, например, матрицу M размера 5×7.

Ввод и вывод двумерного массива (матрицы)

Ввод матрицы с клавиатуры:

```
For i = 1 To 5 'перебираем строки
```

```
For j = 1 To 7 'перебираем столбцы
```

```
    M(i,j) = InputBox("Введите элемент матрицы")
```

```
Next
```

Динамические массивы

Массив, размер которого можно менять в процессе работы программы, называется *динамическим*. Динамические массивы используются, когда заранее неизвестно какой длины он должен быть.

Для создания динамического массива следует:

- объявить массив без указания размерности массива **Dim ИмяМассива ()**
- с помощью оператора **ReDim** назначить действительное число элементов массива

Оператор **ReDim** изменяет количество элементов в массиве, при этом можно в качестве границ массива можно указывать не числа, а переменные. Один и тот же массив можно переопределять в одной программе несколько раз. Если вы хотите при переопределении границ сохранить значения, уже присвоенные элементам массива, нужно воспользоваться ключевым словом **Preserve** (сохранить), в противном случае после переопределения границ массива все его значения будут утеряны.

ReDim Preserve ИмяМассива (границы)

Тип массива менять нельзя.

Если необходимо освободить память, занимаемую динамическим массивом, то можно использовать оператор **Erase**.

Erase ИмяМассива

Память, занимаемая *статическим массивом*, не освобождается. Вместо этого **Erase** присваивает элементам массива «пустые» значения (для числовых массивов – 0, для строковых – пустая строка)

Массивы объектов

В Visual Basic разрешается определять массивы объектов (элементов управления), что весьма удобно, если в программе имеются группы объектов, действующих примерно одинаково. Такие массивы позволяют «привязывать» разные элементы управления к одной процедуре обработки события.

Существует два способа создания массива объектов:

1. установить на форму первый элемент управления, полностью описать его свойства, скопировать его необходимое количество раз и расставить на форме уже готовые копии
2. установить объект, изменить необходимые свойства, установить следующий объект, присвоив ему также же имя (Name) как и у первого объекта.

Элементы массива объектов имеют одинаковые имена, различаются по индексу (Index), остальные свойства могут совпадать или отличаться. Для обращения к элементу массива объектов необходимо указать имя объекта и в круглых скобках индекс.

Создание и использование вызываемых процедур и функций

Любая программа на Visual Basic является процедурой или функцией.

В предыдущих разделах рассматривались процедуры обработки событий, которые в процессе выполнения программы запускались в результате свершения некоторого определенного события (щелчок мыши, нажатие клавиши и т.п.). Однако в Visual Basic имеется возможность выделить неоднократно повторяющиеся фрагменты программы в определенном образом организованные функциональные блоки операторов, которые можно использовать без их повторного написания и отладки. При этом сложные программы получают более компактными. Организовать функциональные блоки в Visual Basic можно в виде процедур (**Procedure**) и функции (**Function**).

Процедура или функция, как и цикл, предназначены для неоднократного исполнения набора каких-то операторов. Отличие от цикла состоит в том, что сама по себе процедура (функция) операторы не повторяет, а объединяет их в единую группу, которую можно выполнить в любой момент, т.е. она действует при вызове. Такие функции и процедуры называют *вызываемые* или *процедурами общего назначения*.

Прекращение выполнения вызванной процедуры или функции (Возврат) означает возвращение в вызывающую процедуру или функцию, выполнение которой продолжается с оператора, следующего за вызывающим оператором.

Программный код процедур и функций, не являющихся процедурами обработки события, помещается в раздел **General** списка *Object* в окне программного кода.

Процедуры SUB

Общий формат процедуры SUB:

SUB ИмяПроцедуры (параметры)

операторы

END SUB

Процедура SUB вызывается как отдельный оператор и не может быть использована как часть выражения.

Вызов процедуры SUB:

1 способ

CALL ИмяПроцедуры(аргументы)

2 способ

ИмяПроцедуры аргументы

Список аргументов определяет связь по данным между процедурами. По количеству, порядку следования и типу параметры и аргументы должны совпадать. Процедура SUB может возвращать несколько значений в вызывающую подпрограмму, производя вычисления над переменными из списка аргументов. Это единственная возможность для SUB переслать значения.

Exit Sub вызывает прекращение выполнения процедуры и выход из нее в вызывающую программу.

Функции

Функции целесообразно создавать, когда необходимо многократно вычислять одно и то же выражение при различных значениях параметров.

Общий формат процедуры **FUNCTION**

FUNCTION ИмяФункции [(список параметров)] [As тип]

[операторы объявления]

[операторы]

ИмяФункции = выражение

END FUNCTION

Параметры - значения, которые передаются функции при ее вызове.

Выражение - возвращаемое значение функции

Список параметров имеет следующий синтаксис:

имя[()][As тип], имя[()][As тип]...

Имя - имя переменной или массива без указания границ значения индексов

Список аргументов определяет связь по данным между вызывающей (функция, из которой происходит вызов) и вызываемой (функция, которая вызывается) процедурами или функциями.

Exit Function вызывает прекращение выполнения функции и выход из нее в вызывающую программу.

Функция вызывается точно также, как и любая встроенная функция, т.е. она является операндом выражения с указанием списка аргументов. Список аргументов и список параметров должны соответствовать по количеству, порядку следования и типу.

Рекурсивные процедуры и функции

Объект называется *рекурсивным*, если он целиком или частично определяется через самого себя. Объекты, которые определяются через себя целиком, практического интереса не представляют, т.к. являются бесконечными.

Рекурсивная функция (процедура) – это такая функция (процедура), среди выполняемых операторов которой есть оператор вызова самой этой функции (процедуры). Среди этих выполняемых операторов обязательно должно быть условие завершения (продолжения) рекурсии. Последовательность рекурсивных вызовов называется *рекурсивным спуском*. Номер рекурсивного вызова в этой последовательности называется *глубиной рекурсии*. Операторы, расположенные после оператора рекурсивного вызова, выполняются в порядке, обратном тому, в котором происходят рекурсивные вызовы. Говорят, что действие выполняется на *рекурсивном подъеме*.

2 ГРАФИЧЕСКИЕ ПРИМИТИВЫ, СВОЙСТВА ОБЪЕКТОВ

2.1 Развитие систем автоматизированного проектирования

Информационные технологии уже давно уверенно вошли в нашу повседневную жизнь, и архитектурно-строительное проектирование в этом случае не является исключением.

Большое внимание в последние годы уделяется автоматизации расчетно-конструкторских работ при проектировании, где основные параметры выбираются и оптимизируются в интерактивном режиме диалога проектировщика и ЭВМ. Однако на всех этих стадиях автоматизации проектирования инженеру помимо изучения инструкций по эксплуатации и написанию программ приходится познавать ряд по сути дела ненужных ему подробностей системных программ и языков программирования.

Под автоматизацией проектирования понимают систематическое применение ЭВМ в процессе проектирования при распределении функций между проектировщиком и ЭВМ и выборе методов машинного решения задач.

САПР (в архитектурно-строительном направлении) - комплекс научных методов исследования деятельности в области архитектурно-строительного проектирования с целью разработки системы методик, математических моделей и алгоритмов, обеспечивающих наилучшее распределение функции между проектировщиком и машиной при поиске оптимальных проектных решений.

В результате изучения объектов в процессе проектирования разрабатываются модели и алгоритмы, с максимальной возможной точностью фиксирующие закономерности формообразования объектов строительства и логику процессов их проектирования.

Система автоматизированного проектирования (САПР) - это система, предназначенная для выполнения проектных работ с применением компьютерной техники, а также позволяющая создавать конструкторскую и технологическую документацию на отдельные изделия, здания и сооружения.

Поэтому автоматизация проектирования отличается от обычного использования ЭВМ в процессах проектирования тем, что в ней рассматриваются вопросы построения системы, а не совокупность отдельных задач. Эта дисциплина является методологической, поскольку она обобщает черты, являющиеся общими для разных конкретных приложений.

Основные требования к САПР связаны в основном с их эксплуатационными характеристиками, универсальностью САПР, а также возможностью адаптации к быстроменяющимся условиям проектирования и производства.

К основным требованиям относятся:

- Простой доступ пользователя к САПР. Чем выше «интеллект» системы, тем более прост и лаконичен язык общения пользователя с САПР. Основой простоты взаимодействия пользователя с системой САПР является программное обеспечение.

- Прямой доступ пользователя к САПР. Под прямым доступом понимается возможность оперативного ввода данных и отображение результатов проектирования. В подобном режиме прямого доступа пользователь может формировать и отлаживать программу, вводить новые данные, получать в виде распечаток и графических результатов проектные и расчетные операции.

Интерфейс пользователя является важнейшим параметром системы. Так, при использовании многоуровневых ниспадающих меню пользователь быстро утомляется. Использование пиктограмм вместо текстовых меню значительно упрощает освоение системы и ускоряет работу с системой.

Все САПР ориентированы на пользователя, т.е. внутренние процессы, происходящие в САПР, остаются для пользователя скрытыми.

Первые программы автоматизированного проектирования были созданы для нужд электронной и радиотехнической промышленности. Они появились в конце 50-х - начале 60-х годов прошлого века. В тот период основное внимание уделялось системам автоматизированного черчения. Разрабатываются методы оптимизации систем по заданным критериям.

В 60-е годы наблюдается широкое распространение машинных расчетов в области экономики и бухгалтерии, были разработаны алгоритмы решения систем линейных уравнений большого порядка.

В 70-х годах были получены результаты, показавшие, что область проектирования поддается компьютеризации.

В 80-х годах внедрились микро и суперкомпьютеры, которые стали доступны университетам и малым фирмам. Когда стол для черчения заменяется дисплеем, то скорость работы опытного чертежника повышается в 3 раза.

Девяностые годы - период зрелости, когда были созданы основные линейки программных продуктов, существующие до настоящего времени.

Системы автоматизированного проектирования позволили в значительной степени освободить конструктора-проектировщика от однообразной, трудоемкой и утомительной работы и повысить его возможности на этапах принятия решений.

Этапы развития САПР

1-й этап (1960-е годы) – время интегральных схем (ИС) низкой интеграции, содержащих десятки элементов, объединённых по простым электрическим схемам. Программного обеспечения для САПР тогда ещё не было. Расчёты таких схем производились с применением имевшихся на предприятиях ЭВМ первого поколения («ЭМ-220» (г. Ульяновск), ЭВМ «Минск-22 (рис. 2.1), - 32» и др.). Это было время зарождения «машинного проектирования», позже развившегося в САПР.



Рисунок 2.1 - Электронная цифровая вычислительная машина «Минск-22»

2-й этап (начало 1970х годов) – появление серийно выпускаемых Минрадиопромом вычислительных машин БЭСМ-6 и ряда ЕС ЭВМ (рис. 2.2) с развитой операционной системой и периферийными устройствами, удалёнными терминалами (дисплеями) и графопостроителями. В результате устанавливаются рабочие места разработчиков для моделирования, расчётов, подготовки данных, проектирования и контроля топологии интегральных схем.



Рисунок 2.2 - ЭВМ ЕС-1035

3-й этап – (середина 1970-х годов)

Термин САПР – системы автоматизированного проектирования, связан с появлением АРМ (автоматизированных рабочих мест) на базе мини-ЭВМ ряда СМ-2, СМ-3, СМ-4, выпускаемых на предприятиях Министерства приборостроения. Они пользовались популярностью, так как имели развитую периферию и отработанное программное обеспечение с хорошим текстовым описанием.

4-й этап – (начало 1980-х годов) заключался в создании мощных многомашинных иерархических комплексов, объединённых в единую систему и включающих большие ЭВМ типа ЕС1060, автоматизированные места программистов, конструкторов.

Основной задачей человеко-компьютерного взаимодействия является улучшение взаимодействия между человеком и компьютером, делая компью-

теры более удобными и восприимчивыми к потребностям пользователей. В частности, человеко-компьютерное взаимодействие занимается:

- методологией и развитием проектирования интерфейсов;
- методами реализации интерфейсов (например, программные инструментари, библиотеки и рациональные алгоритмы);
- методами для оценки и сравнения таких интерфейсов;
- разработкой новых интерфейсов и методов взаимодействия;

Долгосрочной задачей человеко-компьютерного взаимодействия является разработка системы, которая снизит барьер между человеческой когнитивной моделью того, чего они хотят достичь и пониманием компьютера поставленных перед ним задач.

Специалисты человеко-компьютерного взаимодействия - это, как правило, разработчики, занимающиеся практическим применением методик разработки к реальным всемирным проблемам. Их работа, зачастую, вращается вокруг разработки графических- и веб- интерфейсов.

Компоненты и виды обеспечений САПР

Система автоматизированного проектирования (САПР) — это организационно-техническая система, состоящая из комплекса средств автоматизации проектирования (КСАП), взаимосвязанного с соответствующими подразделениями проектной организации и коллективом специалистов (пользователей системы) и выполняющая автоматизированное проектирование.

Комплекс средств автоматизации проектирования — это совокупность различных видов (компонент) обеспечения автоматизированного проектирования, необходимых для выполнения проектных процедур в автоматизированном режиме.

Существуют семь видов обеспечения автоматизированного проектирования:

1. Техническое обеспечение — совокупность связанных и взаимодействующих технических (аппаратных) средств, используемых в САПР для обработки, хранения, передачи информации, организации общения человека с ЭВМ, изготовления проектной документации.

2. Информационное обеспечение — совокупность сведений, необходимых для выполнения проектирования. Состоит из описания стандартных проектных процедур, типовых проектных решений, комплектующих изделий и их моделей, правил и норм проектирования. Основная часть информационного обеспечения САПР — это базы данных.

3. Лингвистическое обеспечение — совокупность языков, используемых в САПР для представления информации о проектируемых объектах, процессах и средствах проектирования которой обмениваются пользователи САПР с ЭВМ и между собой в процессе автоматизированного проектирования. Включает термины, определения, правила формализации естественного языка, методы сжатия и развертывания.

4. Математическое обеспечение - совокупность математических моделей, методов, алгоритмов для решения задач автоматизированного проектирования. Реализуется в программном обеспечении САПР.

5. Программное обеспечение — совокупность программ вместе с необходимой программной документацией предназначенных для использования в САПР. Подразделяется на общесистемное и прикладное.

6. Организационное обеспечение — совокупность документов, определяющих состав проектной организации, положения, инструкции, приказы платные расписания квалификационные требования и другие документы, регламентирующие организационную структуру подразделений проектной организации и их взаимодействие с комплексом средств автоматизированного проектирования.

7. Методическое обеспечение — документы, в которых отражены состав, правила отбора и эксплуатации средств автоматизированного проектирования. Иногда понятие методическое обеспечение расширяют, включая в него лингвистическое и математическое обеспечения.

Технические средства систем автоматизированного проектирования — это совокупность систем, машин, приборов, механизмов, устройств и прочих видов оборудования, предназначенных для автоматизации различных технологических процессов информатики, причем таких, выходным продуктом которых является именно информация (сведения, знания) или данные, используемые для удовлетворения информационных потребностей в разных областях предметной деятельности общества.

Все технические средства систем автоматизированного проектирования в зависимости от выполняемых функций можно разделить на шесть основных групп:

- 1) устройства ввода информации;
- 2) устройства вывода информации;
- 3) устройства обработки информации;
- 4) устройства передачи и приема информации;
- 5) устройства хранения информации;
- 6) многофункциональные устройства.

В период становления и развития САПР широко применялись графические системы, которые обеспечивали только вывод графических изображений на экран монитора, графопостроителя или печатающего устройства. В настоящее время такие системы развиваются по пути создания интерактивных графических систем, обеспечивающих диалог пользователя с ЭВМ с включением режима геометрического моделирования.

Развитые графические системы САПР включают две компоненты, одна из которых называется базовая, а вторая - прикладная.

Базовую графическую систему (БГС) относят к инструментальным средствам компьютерной графики. Посредством БГС реализуется связь прикладных программ и графических устройств.

Состав прикладных графических систем САПР ориентирован на реше-

ние конкретных графических и геометрических задач. При разработке таких систем используются алгоритмические, а также специализированные графические языки.

2.1.1 Общие сведения о системе AutoCAD

Комплексная графическая подготовка инженера в настоящее время тесно связана с освоением компьютерных технологий. Владение современными компьютерными графическими редакторами позволяет выйти на качественно новый уровень проектирования.

Современный уровень программных и технических средств электронной вычислительной техники дает возможность перехода от традиционных, ручных методов конструирования к новым информационным технологиям с использованием компьютеров, к созданию систем автоматизированного проектирования и разработки конструкторской документации.

Разработанная фирмой AutoDesk и появившаяся на рынке в конце 1982г. система AutoCAD получила необычайно широкое распространение. AutoCAD представляет собой среду проектирования, которая постоянно развивается. Разработчики системы стараются сохранить преемственность, как в командах, так и в общей структуре. От версии к версии сохраняются, уже ставшие привычными для пользователя, возможность запуска команд из диалоговой строки, использование командного и выпадающих меню.

AutoCAD – интерактивная графическая система, позволяющая в режиме диалога разрабатывать чертежи на плоскости, моделировать объекты в трехмерном пространстве, а также создавать программные модули на встроенном алгоритмическом языке AutoLISP.

AutoCAD представляет собой систему, позволяющую автоматизировать чертежно-графические работы. В графическом пакете AutoCAD есть все, что необходимо конструктору для создания чертежа. Инструментам ручного черчения в автоматизированной среде соответствуют графические примитивы (точка, отрезок, окружность и др.), команды их редактирования (стирание, перенос, копирование и т. п.), команды установки свойств примитива (задание толщины, типа и цвета графических объектов). Для выбора листа нужного формата и масштаба чертежа в системе есть соответствующие команды настройки чертежа. Для нанесения размера конструктору необходимо лишь задать место его расположения на чертеже. Размерная и выносная линии, а также стрелки и надписи выполняются автоматически, а в последних версиях AutoCAD есть режим полной автоматизации простановки размеров.

В автоматизированной среде конструктору нет необходимости напрягать зрение при выполнении отдельных мелких частей чертежа, так как ему предоставляются средства управления изображением на экране. Соответствующие команды AutoCAD позволяют увеличивать изображение чертежа на экране или уменьшать его при необходимости, а также перемещать границы видимой на экране части чертежа без изменения масштаба изображения.

Система предоставляет конструктору возможность объединять графические объекты в единый блок, который хранится под определенным именем

и при необходимости вставляется в любой чертеж, что избавляет конструктора от вычерчивания одних и тех же часто повторяющихся элементов чертежа. Проектировщик также может создавать изображения отдельных элементов чертежа или отдельных деталей сборки на различных слоях. Это позволяет контролировать совместимость деталей при компоновке. Включая или выключая слои, можно вводить или выводить детали из общей компоновки, создавая тем самым удобство в подборе различных вариантов конструкции изделия. Слои полезно использовать даже в простых чертежах, размещая на каждом отдельном слое заготовку чертежа, обводку, размеры, надписи, осевые линии для последующей возможности быстрого выбора группы объектов и их редактирования.

Разработчики системы, ориентируясь на самый широкий круг пользователей, заложили в пакет богатые возможности настройки AutoCAD на любую предметную область. Опытные пользователи могут настраивать панели инструментов и создавать новые, разрабатывать слайд-фильмы с помощью пакетных файлов, вводить новые типы линий и образцы штриховки, образовывать новые меню. Встроенный в систему AutoCAD язык программирования AutoLISP позволяет описывать часто встречающиеся объекты в параметрической форме. Вызывая такой объект, конструктор может изменять его размеры, а значит, и геометрическую форму, обеспечивая тем самым многовариантность графического изображения. Помимо создания двухмерных чертежей, система AutoCAD позволяет моделировать трехмерные объекты и придавать трехмерным чертежам фотографическую реальность.

Для создания изображений на компьютере используются два принципиальных метода их представления: растровый и векторный.

Под **растровым** (bitmap, raster) понимают способ представления изображения в виде совокупности отдельных неделимых точек (пикселей) различных цветов или оттенков. При увеличении изображения можно увидеть составляющие его квадратики. Пример - Photoshop, Paint. Достоинством такого способа является возможность получения фотореалистичного изображения высокого качества в различном цветовом диапазоне. Недостатком – высокая точность и широкий цветовой диапазон требуют увеличения объема файла для хранения изображения и оперативной памяти для его обработки.

Векторные изображения – наборы точек, соединенных линиями. Графические элементы векторного изображения называются объектами. Построение и редактирование объектов в векторной графике основано на математических формулах. Например, при построении окружности в памяти компьютера хранятся координаты центра и радиус окружности. При изменении размера окружности новый радиус подставляется в формулу, и окружность строится заново. Таким образом, появляется возможность хранить не все точки изображения, а координаты узлов примитивов. Пример – 3DMAX, AutoCAD.

В векторной графике объекты создаются из комбинаций линий, а в растровых они состоят из отдельных элементов, образующих узоры.

Запуск системы AUTOCAD

Установленную программу Autocad можно запустить одним из способов:

1. Двойным щелчком левой кнопкой мыши по ярлыку программы, расположенному на рабочем столе.
2. Выбором программы из всплывающего меню: нажать кнопки Пуск – Программы – Autodesk – AutoCAD – Autocad.exe.
3. Щелкнуть два раза мышкой по значку файла одного из чертежей Autocad.

При запуске автоматически загружается Autocad с файлом шаблона acadiso.dwt (лист формата А3). При создании нового рисунка Файл-Создать (File-Open) появляется диалоговое окно Выбор шаблона (Select template), в котором можно выбрать другой шаблон из предложенного списка с различными форматами и рамками (рис. 2.3).

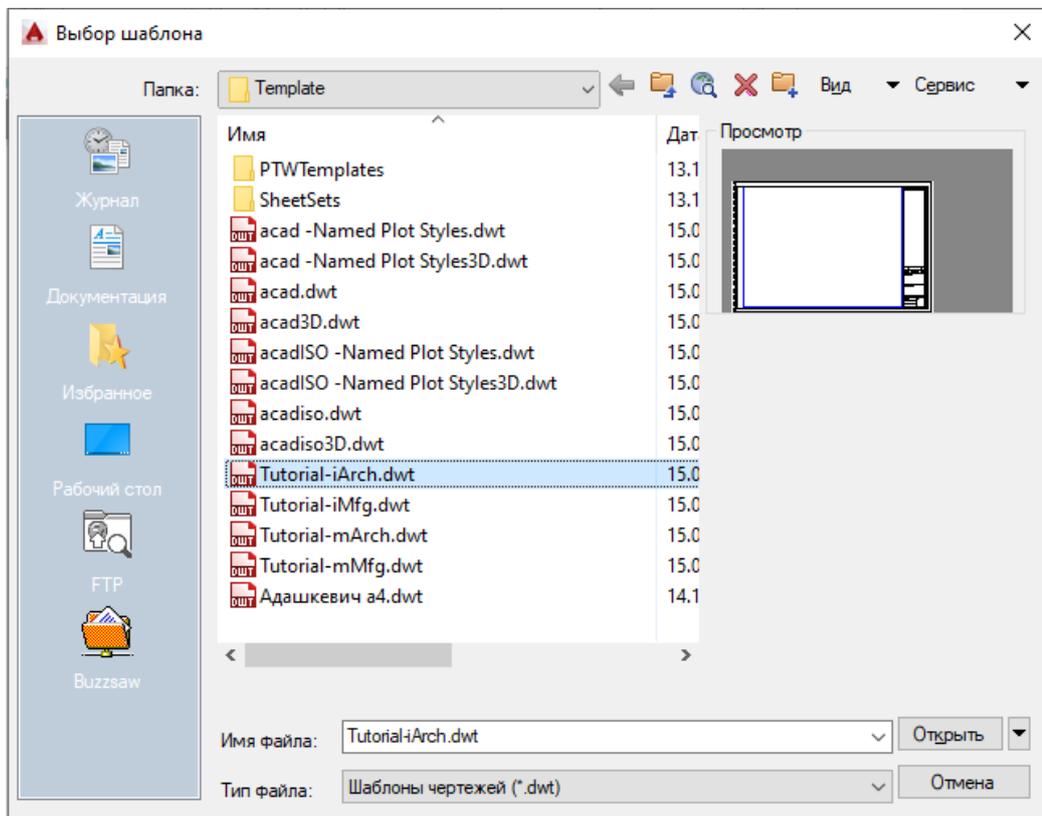


Рисунок 2.3 - Диалоговое окно Выбор шаблона

После загрузки появляется графическое окно Autocad (рис.2.4), которое содержит основные элементы интерфейса, многие из которых встречаются и в других приложениях Windows, и область, где выполняется построение чертежа.

Графическое окно состоит из следующих областей:

1. Заголовок окна.
2. Системное (падающее) меню.
3. Лента с вкладками.

4. Графическое поле.
5. Командная строка.
6. Вкладки модели и компоновки листа.
7. Строка состояния.
8. Переключение пространств.
9. Компас.

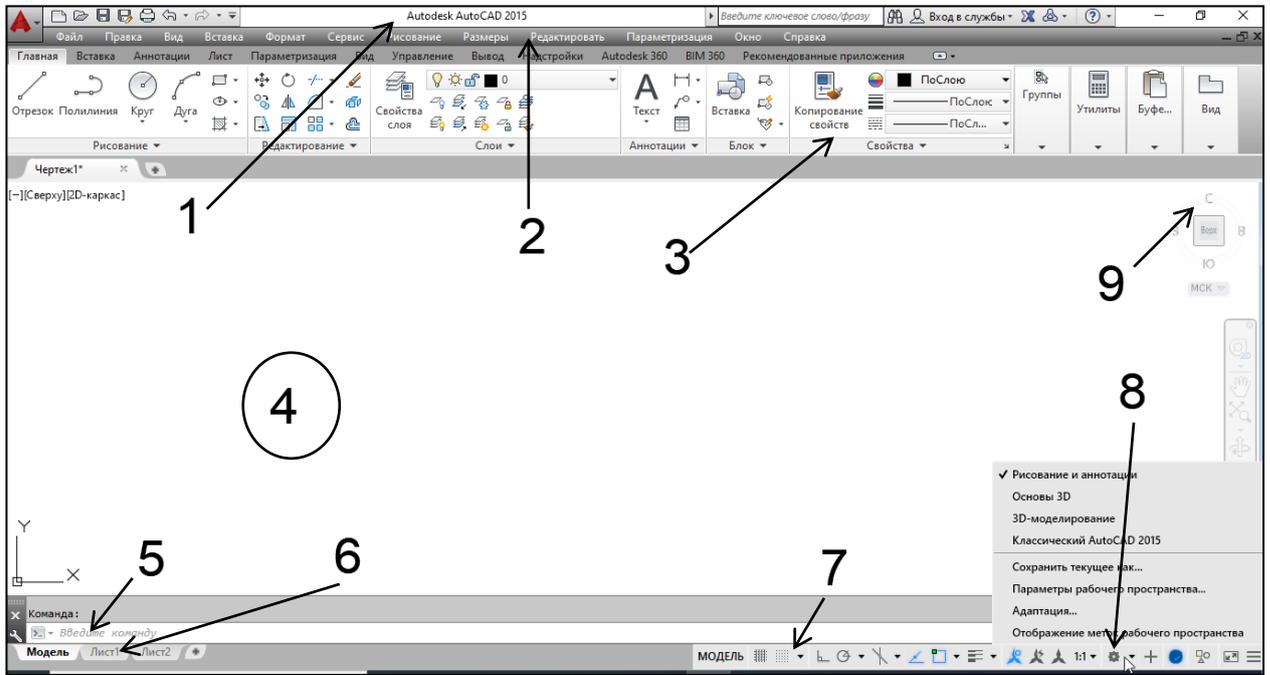


Рисунок 2.4 - Графическое окно AutoCAD

1. Заголовок окна содержит наименование программы, имя файла, а также кнопки работы с окном. По умолчанию AutoCAD присваивает вновь создаваемому файлу имя Чертеж1.dwg.

2. Системное (падающее меню) содержит следующие группы команд:

- Файл (File) – работа с файлами;
- Правка (Edit) – редактирование частей графического поля;
- Вид (View) – работа с изображением: управление параметрами дисплея, масштабирование и панорамирование изображения, установка точки зрения;
- Вставка (Insert) – работа с блоками, ссылками, полями, растровыми изображениями;
- Формат (Format) – работа со стилями, свойствами объектов, установка единиц измерения и границ чертежа;
- Сервис (Tools) – настройка рабочей среды AutoCAD, работа с пользовательской системой координат;
- Рисование (Draw) – команды черчения;
- Размеры (Dimension) – простановка размеров и управление раз-

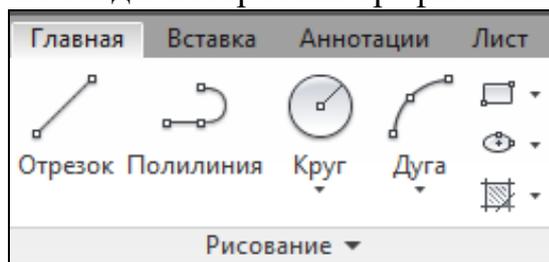
мерным стилем;

- Редактирование (Modify) – команды редактирования элементов чертежа;
- Окно (Windows) – параметры расположения окон, панелей инструментов и файлов;
- Справка (Help) - справочник по AutoCAD.

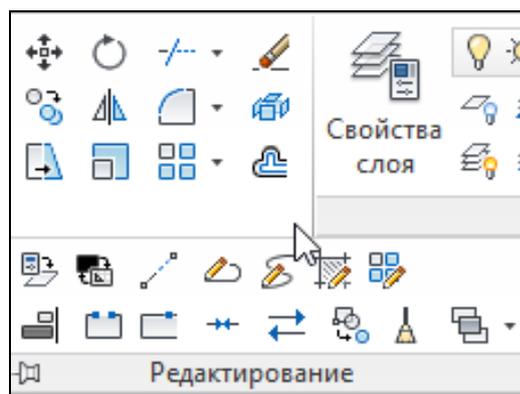
3. Лента – набор различных команд, сгруппированных во вкладки.

На вкладке **Главная** находятся следующие панели:

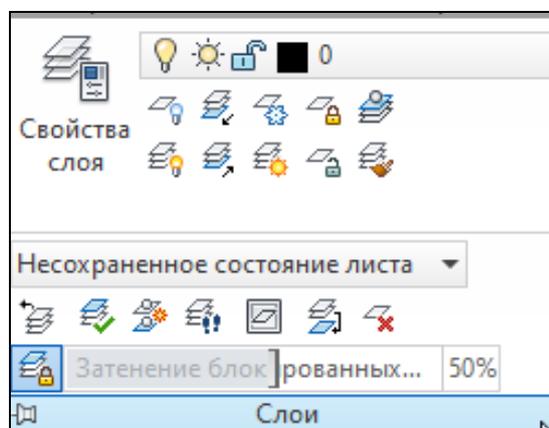
- Рисование – команды построения графических примитивов.



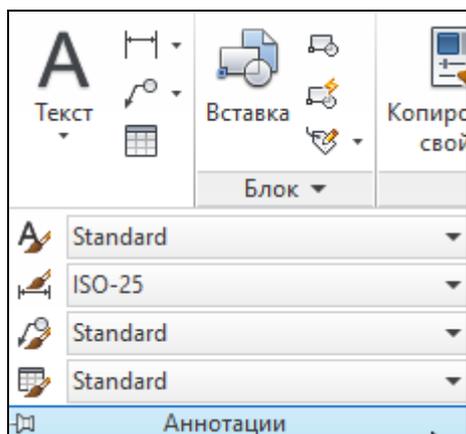
- Редактирование – команды редактирования графических примитивов.



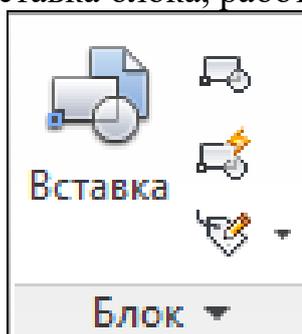
- Слой – создание и управление слоями.



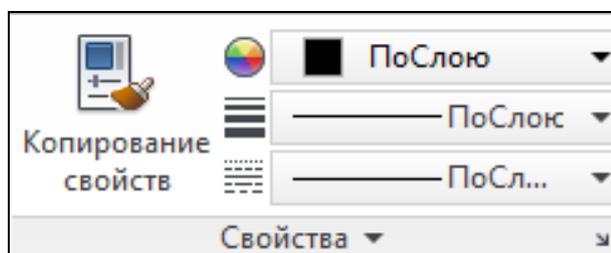
- Аннотации – содержит команды для работы с текстом, размерами, мультывыноской, таблицами и стилями.



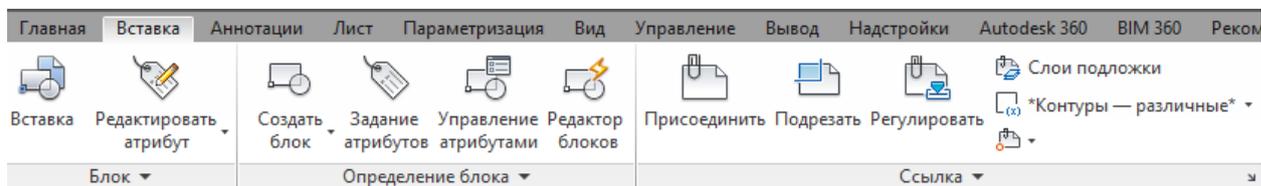
- Блок – создание и вставка блока, работа с атрибутами блока.



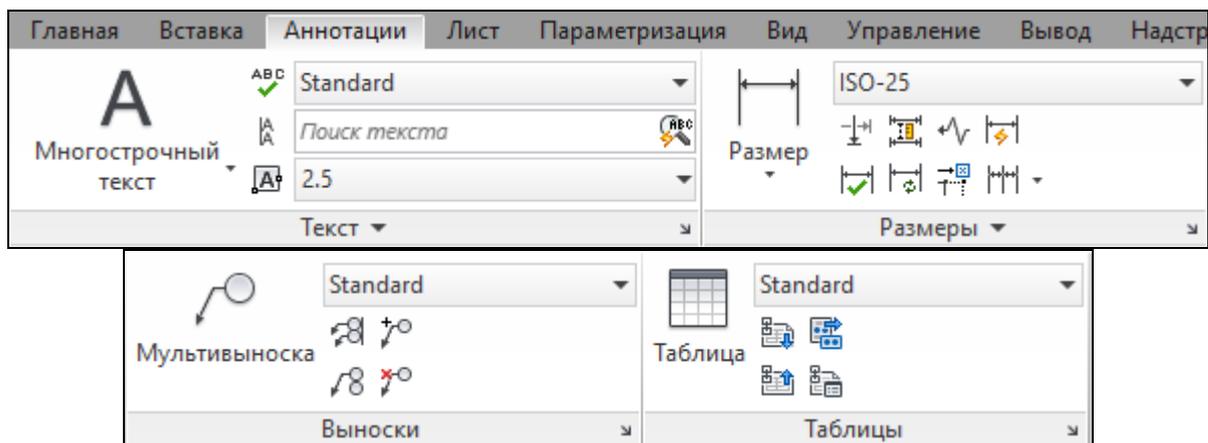
- Свойства – установка цвета, веса и типа линий.



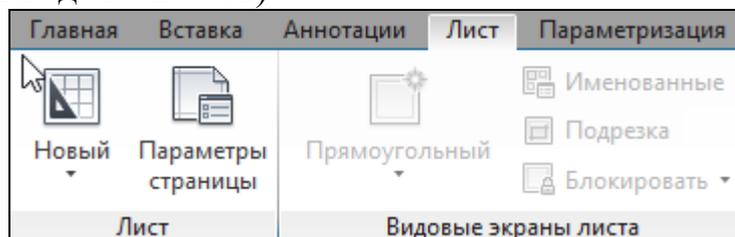
На вкладке **Вставка** находятся команды для работы с блоками и ссылками.



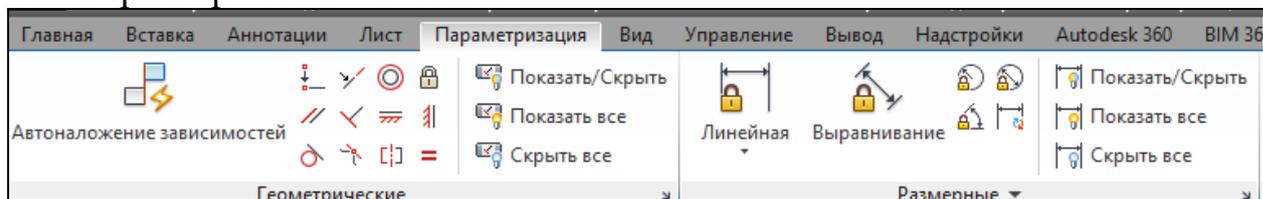
На вкладке **Аннотации** находятся команды для работы с текстом, размерами, выноской и таблицами.



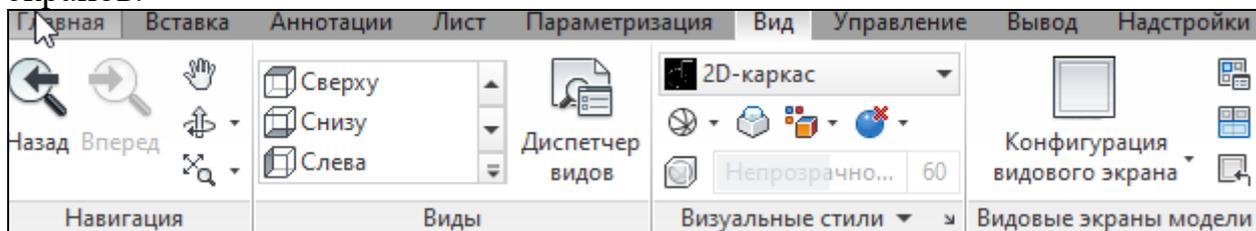
На вкладке **Лист** находятся команды для работы в пространстве Листа (создание и модификация листов и видовых экранов, установка параметров страницы для вывода на печать).



Вкладка **Параметризация** содержит команды наложения геометрических и размерных зависимостей.



Вкладка **Вид** – команды управления визуальными стилями, навигацией. Содержит команды установки различных видов и создания видовых экранов.



4. **Графическое поле** – основная рабочая зона, где осуществляются графические построения.

5. **Вкладки модели и листа** - в пространстве модели (на вкладке **Модель**) выполняется разработка и рисование двумерных рисунков и трехмерных моделей, пространство листа (вкладки **Лист**) используется для окончательной компоновки и подготовки рисунка к выводу на печать.

6. Командная строка служит для ввода команд и ведения диалога с системой AutoCAD.

7. Строка состояния содержит координаты (X,Y,Z) положения курсора в зоне рисования окна (при плоском черчении координата Z всегда равна 0), и кнопки, которые управляют вызовом команд режимов рисования (ШАГ, СЕТКА и т.д.).

При возникновении проблем ("зависание" программы, перезагрузка системы) можно восстановить файл в Диспетчере восстановления чертежей (рис. 2.5) на вкладке Главная – Утилиты.

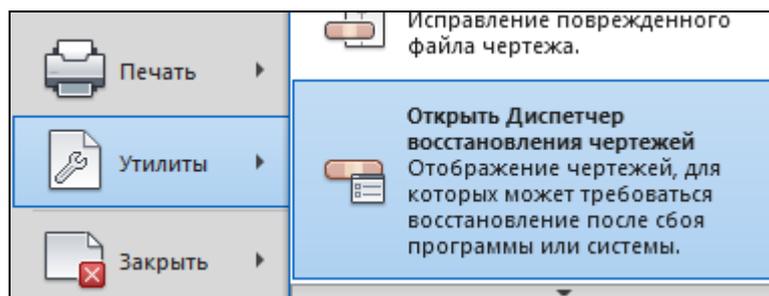


Рисунок 2.5 – Вызов диспетчера восстановления чертежей

Контекстно-зависимое меню

Контекстно-зависимое меню – это перечень действий, возможных к исполнению в данный момент времени над данным объектом. Оно открывается нажатием на правую кнопку мыши, курсор при этом находится на объекте.

Выход из AutoCAD

Для завершения работы и выхода из системы AutoCAD имеется несколько способов:

1. Кнопка x (закрытие окна).
2. Пункт Выход (Exit) падающего меню Файл (File).
3. Горячие клавиши Ctrl+Q.

2.1.2 Команды системы AutoCAD

Настройка графического окна программы

Для настройки графического окна можно, щелкнув правой кнопкой мышки по экрану, в контекстном меню выбрать Параметры (рис.2.6).

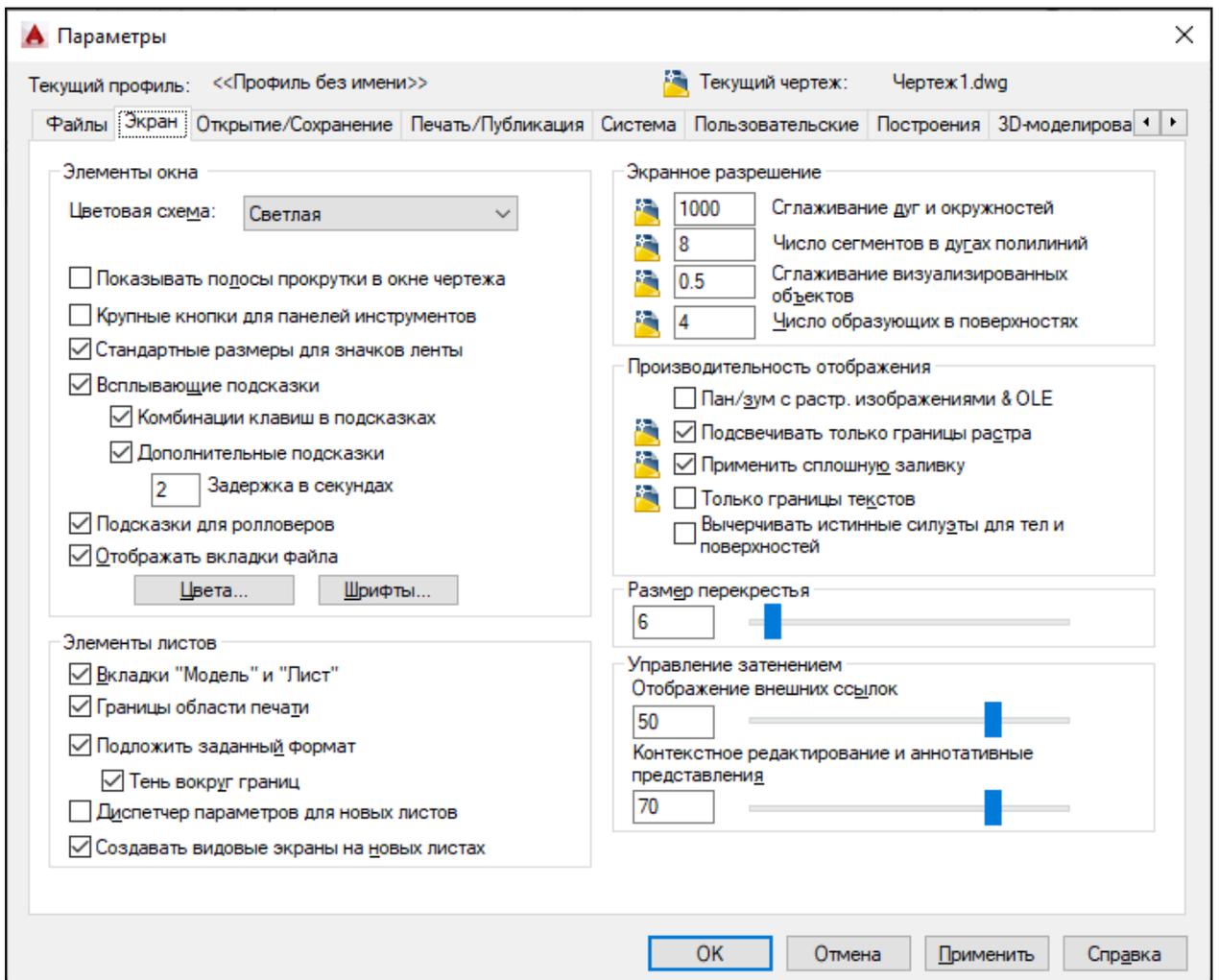


Рисунок 2.6 – Параметры настройки системы

Для удобной работы в системе AutoCAD можно настроить в диалоговом окне Параметры следующие опции:

- 1) фон области рисунка: Экран - Цвет (выбрать требуемый цвет) – Принять (Display - Colors);
- 2) размер курсора: Экран - Размер перекрестья (Display-Crosshair size);
- 3) цвет автопривязки: Построения - Цвет автопривязки (Drafting-Colors);
- 4) установить максимально возможный размер области рисования AutoCAD: для этого необходимо нажать клавиши CTRL+0 (ноль). Это сочетание клавиш уберет строку заголовка и все панели инструментов. Чтобы вернуть нормальный вид окна, повторно нажать CTRL+0.

Использование текстового окна

В текстовом окне отображается полный протокол команд текущего сеанса работы. Для перемещения по текстовому окну служит полоса прокрутки на правом краю окна. Для вызова текстового окна необходимо нажать клавишу F2.

Система AutoCAD создана для интерактивной (диалоговой) работы с

пользователем. Весь диалог с системой идет на языке команд. В ответ на приглашение Команда: (Command:) в командной строке необходимо ввести команду, далее последуют дополнительные запросы команды.

Способы задания команд

Команды задаются следующими способами:

- набрать на клавиатуре (и нажать "Enter");
- выбрать из системного (падающего) меню;
- щелкнуть по соответствующей пиктограмме левой клавишей мыши на панели инструментов;
- выбрать из контекстно-зависимого меню соответствующую команду.

После ввода команды AutoCAD выдает запросы, в ответ на которые необходимо ввести дополнительную информацию: числовое значение (например, расстояние, угол и т.д.), опцию или точку.

Если в ответ на запрос "Команда: " нажать клавишу "Enter" или "Пробел", то AutoCAD повторит вызов предыдущей команды.

Выполнение команды завершается нажатием одной из клавиш: "Enter", "Пробел" или "Esc".

Точность единиц измерения

Все установки типа и формата представления линейных и угловых единиц измерения выполняются в диалоговом окне Единицы рисунка, которое открывается командой Формат - Единицы (Format – Units). Это окно содержит два раздела для задания формата и точности линейных и угловых единиц измерения (рис.2.7). Нажав на кнопку Направление, можно выбрать расположение нулевого угла, от которого начинается отсчет углов.

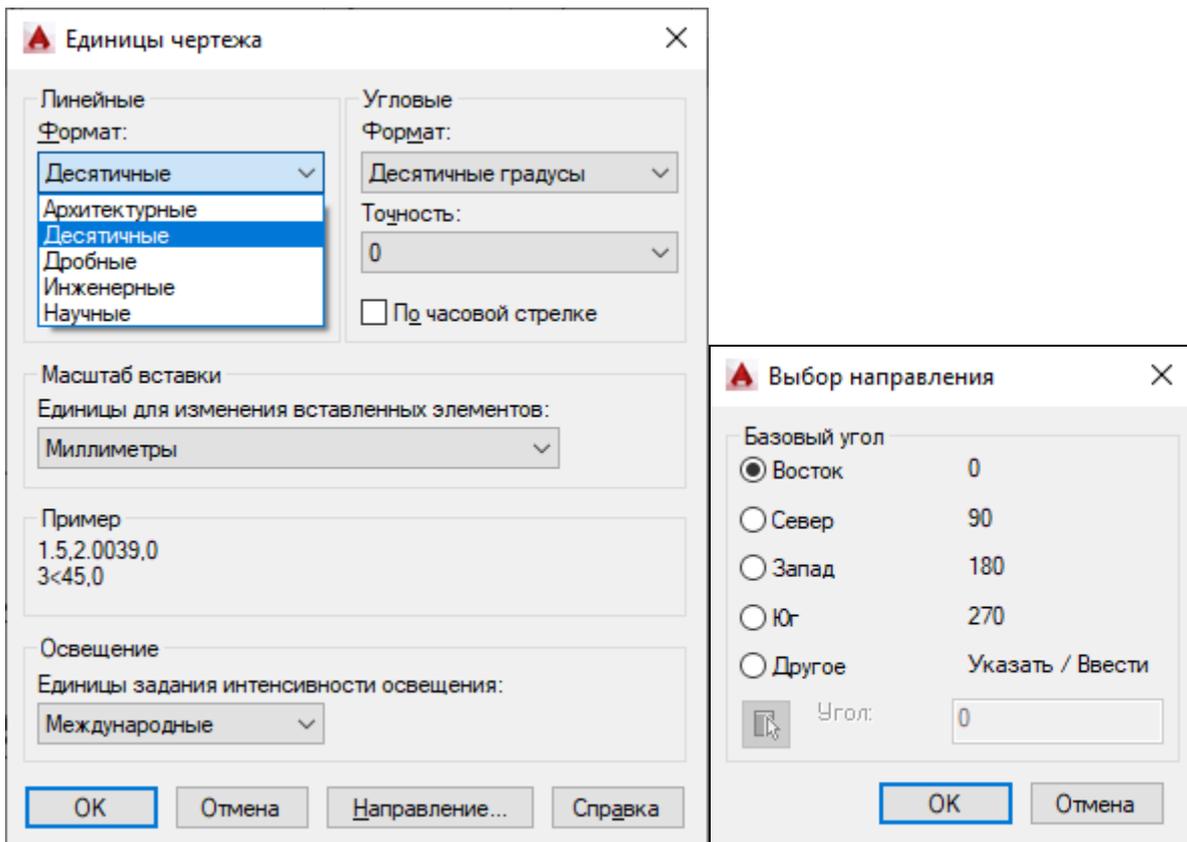


Рисунок 2.7 – Установка линейных и угловых единиц измерения

По умолчанию положительный угол откладывается против часовой стрелки от положительного направления оси X (направление на восток). Отрицательный угол откладывается по часовой стрелке (рис. 2.8).

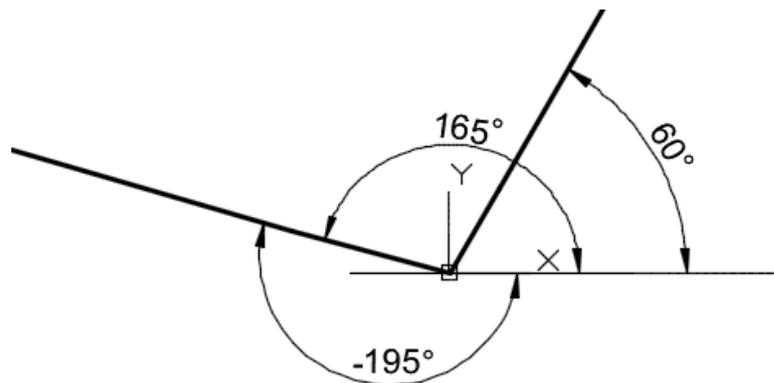


Рисунок 2.8 – Построение отрезков под углом

Работа с изображением

Зумирование.

При работе с чертежом часто возникает необходимость увеличить изображение на экране или уменьшить его. Это позволяет сделать команда Вид - Зумирование - В реальном времени или кнопка на стандартной панели. Команда имитирует изменение фокусного расстояния при рассмотрении объектов. Она делает объекты расположенными ближе или дальше, но не

меняет размеры чертежа. Курсор при зумировании принимает форму увеличительного стекла. Если нажать кнопку мыши и, не отпуская ее, перемещать курсор вверх, произойдет увеличение видимых размеров объекта, а при перемещении курсора вниз – уменьшение видимых размеров объекта. Если после нажатия кнопки выбора устройства указания перемещать курсор от нижней границы окна рисунка до его верхней границы, то увеличение составит 200%.

Вызов команды осуществляется несколькими способами:

- a) выбрать на Стандартной панели кнопку зумирования ;
- b) строка меню Вид - Зумирование - В реальном времени;
- c) контекстное меню: щелкнуть правой кнопкой мыши в области рисования и выбрать "Зумирование";
- d) скроллинг на мышке (прокрутка "колесика" вперед – увеличивает объекты, назад – уменьшает).

Если при прокрутке скроллинга "исчез" весь чертеж, то для его отображения выполните команду Вид – Зумирование – Показать все.

Для завершения зумирования следует нажать ENTER или ESC.

Команда Зумирование содержит следующие параметры (рис.2.9):

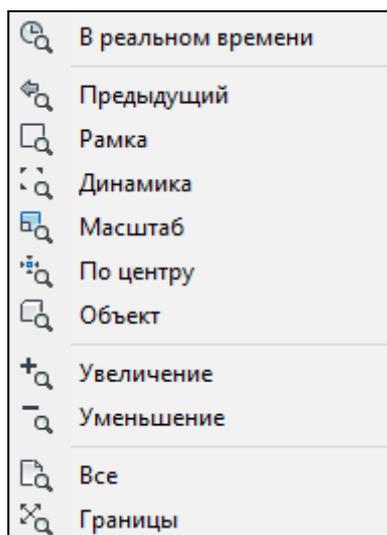


Рисунок 2.9 - Команда Зумирование

Рамка - Показ области рисунка, заключенной между заданными пользователем противоположными углами прямоугольной рамки.

Динамика - Зумирование изображения с использованием динамически изменяемой рамки вида. Рамка вида представляет собой изображение границ экрана, который можно уменьшать и увеличивать, а также перемещать по рисунку.

Масштаб - Зумирование с заданием масштаба. Для увеличения изображения задается коэффициент масштабирования больше единицы, для уменьшения – меньше единицы. Если ввести значение с символом **x** (ла-

тинское) на конце, коэффициент будет исчисляться относительно текущего изображения. Например, ввод 0.5x приведет к тому, что каждый объект будет выведен на экран с размером вдвое меньшим, чем его текущий размер.

Центр - Зумирование от указанной центральной точки нового изображения. Сначала вводится центральная точка, затем увеличение или высота. Меньшее значение высоты приводит к увеличению, а большее - к уменьшению изображения.

Объект - Показ одного или нескольких выбранных объектов максимально увеличенными и расположенными по центру области рисования. Объекты можно выбирать до или после вызова команды.

Увеличение изображения.

Уменьшение изображения.

Все - Показ всего рисунка на экране. В плане с помощью параметра "Все" выполняется увеличение до лимитов сетки или его текущих границ (в зависимости от того, что больше).

Границы - Выводит с наибольшим возможным коэффициентом экранного увеличения все объекты рисунка так, чтобы они заполнили графическую область. Команда Границы дублируется двойным нажатием на скроллинг мыши.

Панорамирование.

Для плоскопараллельного перемещения объекта используются линейки прокрутки окна или команда Панорамирование.

Вызов команды осуществляется несколькими способами:

- a) выбрать на Стандартной панели кнопку  ;
- b) Вид - Зумирование - Панорамирование;
- c) контекстное меню: щелкнуть правой кнопкой мыши в области рисования и выбрать "Панорамирование";
- d) нажав скроллинг на мышке и, не отпуская его, перемещать в нужном направлении. С помощью этой опции пользователь может просматривать различные участки рисунка, перемещая курсор в нужном направлении.

Курсор принимает форму кисти руки  курсор панорамирования . При перемещении курсора по экрану в том же направлении будет перемещаться и изображение. При отпускании кнопки выбора устройства указания панорамирование прекращается. Для завершения панорамирования следует нажать ENTER или ESC.

Редактирование чертежа

Преобразование чертежа производится с помощью команд редактирования. Большинство из них предлагает последовательно указать объекты для работы. Выбор объектов можно производить следующими способами:

– **Прицел** – это квадрат с перекрестьем. Курсор-прицел наводят на объект и щелкают левой мышкой. Выбранные объекты подсвечиваются и становятся пунктирными.

– **С помощью простой рамки.** Если поместить курсор слева от выбираемого объекта, охватывая появившейся сплошной (простой) рамкой объекты, то все объекты, полностью попавшие в рамку, будут выбраны.

– **С помощью секущей рамки.** Если поместить курсор справа от выбираемого объекта, охватывая появившейся пунктирной (секущей) рамкой объекты, то будут выбраны объекты, как полностью попавшие в рамку, так и пересеченные границами рамки.

Для отказа от предварительного выбора объектов – ESC или ENTER.



Если на указанном выше рисунке выбирать объекты простой рамкой (слева направо), то выбраны будут только отрезок и окружность. Если рамку рисовать справа налево, то будут выбраны отрезок, дуга и окружность.

Редактирование чертежа с помощью ручек.

Изменить свойства объекта можно с помощью "ручек". Если выбрать объекты прицелом или рамкой, то непрерывная линия объекта становится пунктирной и в характерных местах возникнут синие квадраты. Это и есть "ручки". Число ручек зависит от вида объекта.

Если щелкнуть левой клавишей мыши по какой-то из ручек, то выбранная ручка выделяется, т.е. меняет синий цвет на красный. Если выбрать крайнюю ручку и указать ее новое положение, то объект растянется вслед за перемещаемой ручкой. Если выбрать среднюю точку и указать новое положение, то весь объект переместится без изменения геометрии.

2.1.3 Ввод координат и настройка параметров чертежа

Системы координат.

В AutoCad по умолчанию используется мировая система координат. По умолчанию начало координат (пиктограмма пересечения осей X и Y) находится в левом нижнем углу графической области экрана. AutoCad – графический редактор, который работает в векторном формате, и для создания объектов необходимо не рисовать их на экране, а вводить только опорные точки. Опираясь на эти точки, программа сама построит объект. Например, для построения окружности достаточно ввести координаты центра окружности и значение радиуса.

В AutoCAD используются следующие системы координат:

а). Система абсолютных координат.

Абсолютные координаты всегда отсчитываются от начала координат (0,0), т.е. от точки пересечения координатных осей. Например, координаты 16,4.2 указывают на точку, удаленную от начала координат по оси X на 16 единиц и по оси Y - на 4.2 единицы.

б). Система относительных координат.

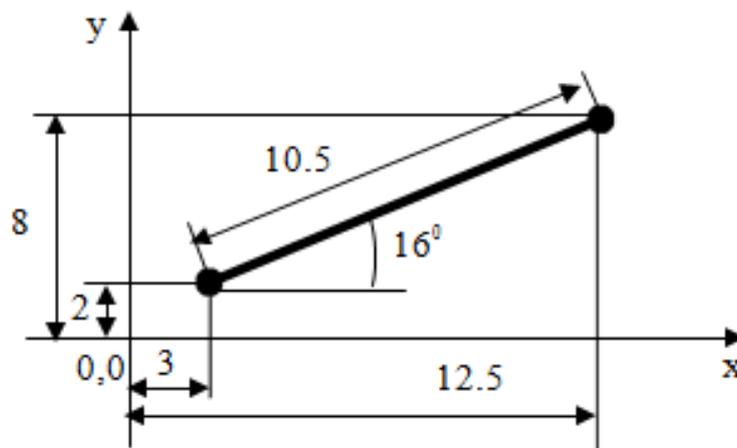
Относительные координаты отсчитываются от последней введенной точки. Для обозначения относительных координат используется знак @ перед числовыми значениями. Например, координаты @3,4 указывают на точку, удаленную от предыдущей точки по оси X на 3 единицы и по оси Y - на 4 единицы.

с). Полярная система координат.

В полярной системе координаты точки представляют собой значения расстояния и угла, отделяемых друг от друга символом открывающей угловой скобки (<). Абсолютные полярные координаты отсчитываются от начала координат (0,0). Относительные полярные координаты отсчитываются от последней введенной точки. Для обозначения относительных координат используется знак @ перед числовыми значениями. Например, для указания точки, находящейся на расстоянии одной единицы относительно предыдущей точки и располагающейся под углом 45 градусов, нужно ввести @1<45.

По умолчанию в AutoCAD возрастание величин углов происходит при движении против часовой стрелки. Для задания направления по часовой стрелке от нулевого угла следует вводить отрицательное значение угла. Например, 1<315 эквивалентно 1<-45.

Пример. Построение отрезка с помощью ввода координат в командной строке в разных системах координат (рис. 2.10).



Система абсолютных координат	Система относительных координат
Команда: <u>line</u> Первая точка: 3,2 Следующая точка или [<u>Замкнуть</u> / <u>Отменить</u>]: 12.5,8	Команда: <u>line</u> Первая точка: 3,2 Следующая точка или [<u>Замкнуть</u> / <u>Отменить</u>]: @9.5,6
Относительные полярные координаты	
Команда: <u>line</u> Первая точка: 3,2 Следующая точка или [<u>Замкнуть</u> / <u>Отменить</u>]: @10.5<16	

Рисунок 2.10 – Пример ввода координат

Координаты точки задаются следующими способами:

1. Указанием курсора на графическом поле чертежа (выбор точки курсором производится щелчком левой мыши).

2. Набором координат с клавиатуры.

При вводе с клавиатуры значения координат X и Y разделяются запятой. Дробная часть числа отделяется от целой части точкой. Координата X определяет расстояние от начала координат по горизонтали, Y- по вертикали. Значения расстояний могут быть положительными или отрицательными. Началом координат считается точка пересечения координатных осей, имеющая координаты (0,0).

3. С помощью объектных привязок.

Объектная привязка это способ задания точек с помощью привязки к имеющимся объектам. Этот способ позволяет точно указать такие точки, как, например, середина отрезка или дуги, центр окружности, пересечения линий и т.д.

4. По направлению.

Ввод точки по направлению предполагает задание направления черчения курсором с помощью режимов рисования, а расстояние от предыдущей точки задается набором числа с клавиатуры.

Границы чертежа.

В начале работы следует установить границы текущего чертежа в соответствие с размерами чертежа и предполагаемого формата вывода на печать.

Границы чертежа (Лимиты) – это прямоугольная область плоскости ХУ мировой системы координат, задаваемая двумя точками (левой нижней и правой верхней) в абсолютных координатах.

Для задания границ чертежа используется команда Формат – Лимиты.

Все запросы отображаются в командной строке:

- Левый нижний угол или [Вкл/Откл] <0.0000,0.0000>: задаются абсолютные координаты левого нижнего угла формата (по умолчанию 0,0).

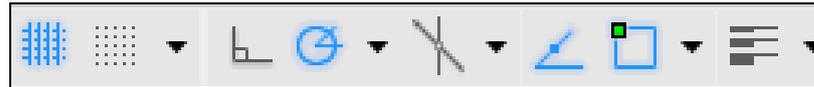
- Правый верхний угол <420.0000,297.0000>: задаются абсолютные

координаты правого верхнего угла формата

Если в командной строке ввести опцию Вкл, то включается контроль за границами чертежа, т.е. точки объектов можно вводить только в пределах лимитов чертежа.

Режимы рисования.

Для удобства, точности и быстроты формирования чертежа в AutoCAD предусмотрены специальные режимы рисования, расположенные в строке состояния. Режим считается действующим, если соответствующая ему кнопка подсвечивается голубым цветом.



Для настройки режимов рисования необходимо щелкнуть правой кнопкой мыши по любой кнопке режимов и выбрать в контекстном меню Параметры. При этом откроется диалоговое окно Режимы рисования (рис.2.11), содержащее 4 вкладки:

- Шаг и сетка
- Отслеживание
- Объектная привязка
- Динамический ввод

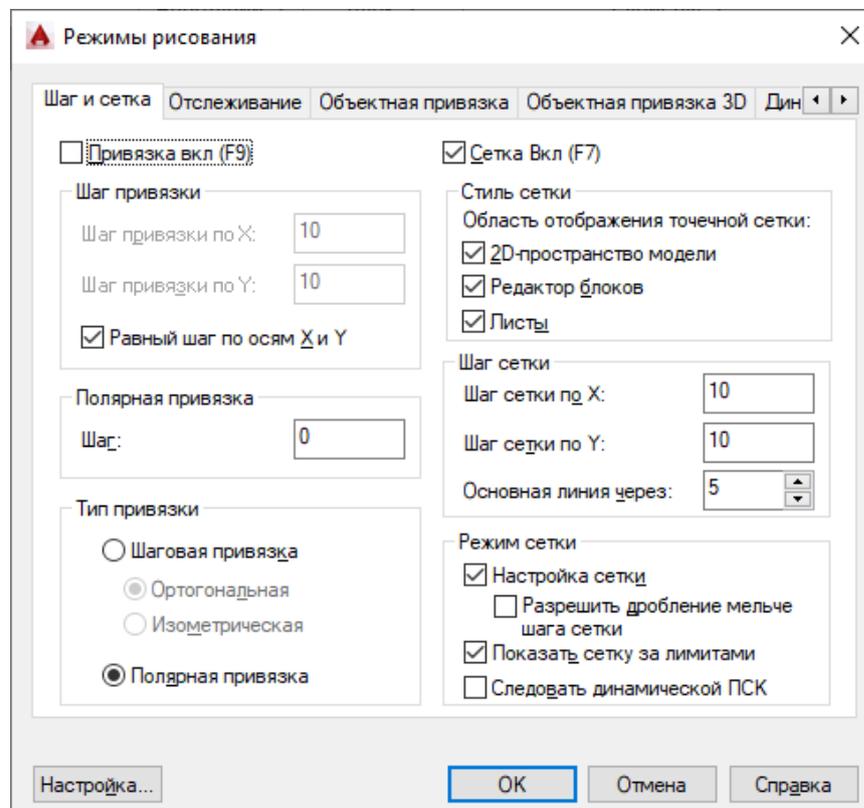


Рисунок 2.11 – Режимы рисования

Видимая сетка (GRID)

Режим, при котором на экране отображаются точки сетки с определенным интервалом между ними. Сетка служит для удобства черчения по узлам сетки и для облегчения визуального восприятия рисунка в пределах лимитов. Она не является объектом рисунка и не выводится на печать. F7 – функциональная клавиша включения/выключения сетки.

Если интервал ячеек видимой сетки слишком мал, то сетка не отображается на экране, и выдается сообщение: *Сетка слишком плотна для изображения.*

Чтобы отобразить сетку, необходимо увеличить расстояние между узлами сетки в диалоговом окне Режимы рисования.

Шаговая привязка (SNAP).

При этом режиме курсор движется с заданным шагом, появляется пунктирная линия и рисование привязано к узлам сетки. F9 – функциональная клавиша вызова режима шаговой привязки.

Шаговая привязка может быть ортогональной или изометрической.

Ортогональное рисование (ORTHO).

Часто создаваемые чертежи имеют большое число взаимно перпендикулярных и параллельных линий. При включенном режиме "Орто" строить объект можно только параллельно осям координат. F8 – функциональная клавиша быстрого вызова.

Объектная привязка (OSNAP).

Режим объектной привязки позволяет повысить точность построений и значительно упрощает процесс построения объектов.

При вводе точек можно использовать объекты, имеющиеся на чертеже. Вместо ввода значений координат можно задавать точки, указывая курсором характерные точки имеющихся объектов (конечные точки и середина отрезков, центр круга и т.д.). По умолчанию при перемещении курсора над объектной привязкой на объекте отображаются маркер и подсказка о назначении привязки. F3 – функциональная клавиша включения/отключения привязки.

Установка фиксации объектных привязок производится в третьей вкладке Объектная привязка диалогового окна Режимы рисования (рис. 2.12). Для фиксации объектной привязки необходимо поставить "галочку" в соответствующем поле.

Объектные привязки можно установить в раскрывающемся списке при нажатии на кнопку Объектная привязка .

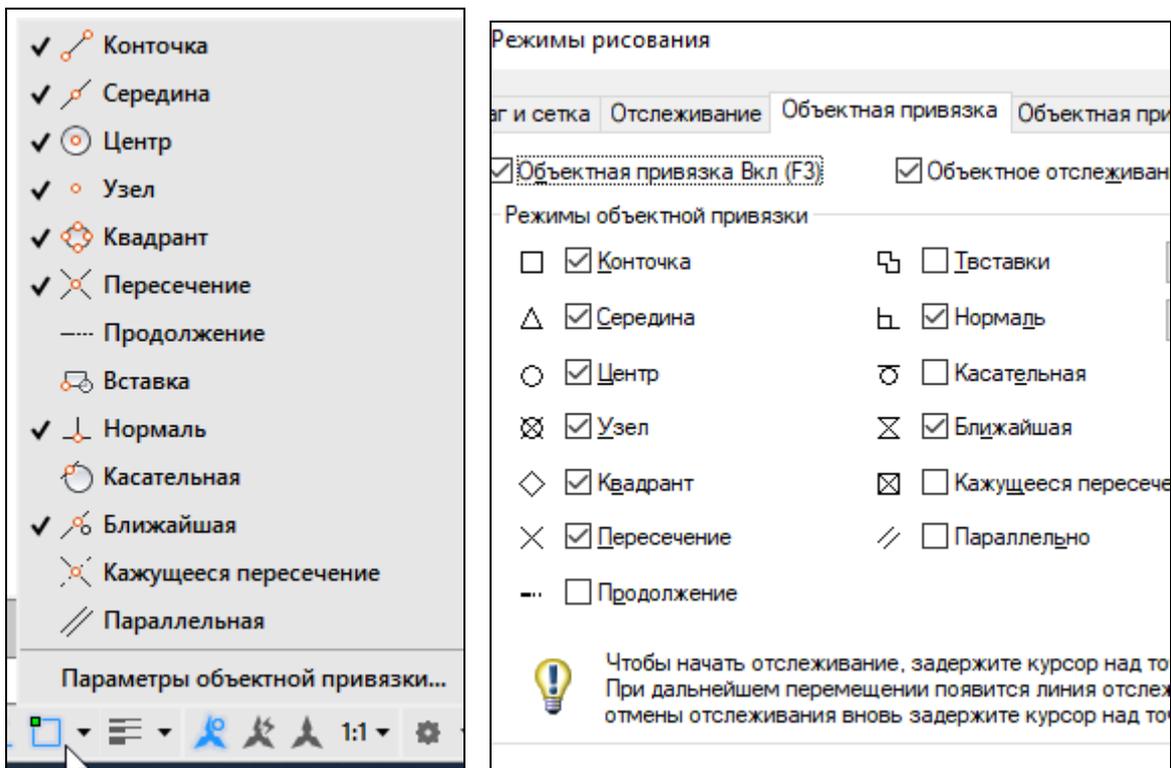


Рисунок 2.12 – Объектная привязка

Полярное отслеживание (POLAR)

Полярное отслеживание - режим, при котором рядом с курсором отображаются его полярные координаты вдоль отслеживаемого угла (пунктирная линия (трассирующая) указывает направление угла), кратного заданному угловому шагу. Включить/выключить режим полярного отслеживания можно с помощью функциональной клавиши F10.

Настроить режим полярного отслеживания можно при помощи второй вкладки Отслеживание (рис. 2.13) диалогового окна Режимы рисования или в раскрывающемся списке при нажатии на кнопку Полярное отслеживание

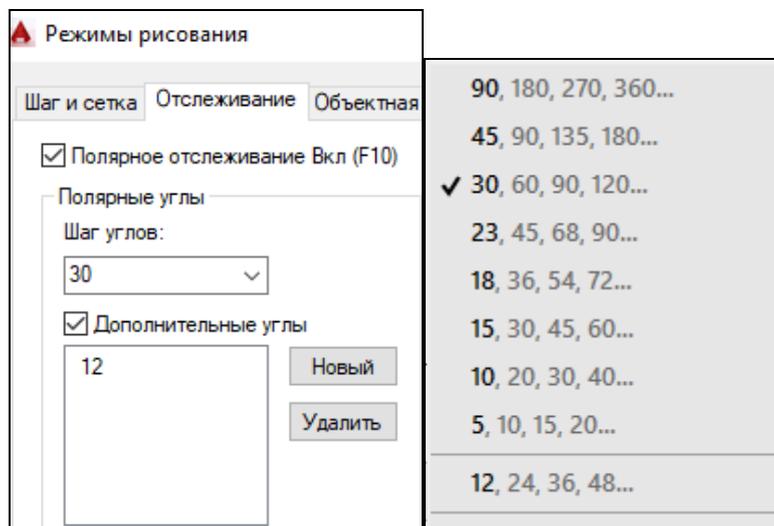


Рисунок 2.13 – Полярное отслеживание

По умолчанию шаг углов равен 90^0 . Это обозначает, что через каждые 90^0 будет появляться пунктирная линия, указывающая направление угла.

Пример. Необходимо нарисовать линию длиной 18 единиц под углом 67^0 .

Для этого на вкладке Полярное отслеживание, щелкнув по кнопке Новый, впишите дополнительный угол 67 и нажмите ОК. При вызове команды Отрезок двигайте мышкой по экрану (не щелкая) пока не появится трассирующая пунктирная линия и подсказка с отображением угла 67^0 возле курсора (рис. 2.14). После появления пунктирной линии на экране наберите на клавиатуре 18 и нажмите Enter. Будет нарисована линия длиной 18 единиц под углом 67^0 .



Рисунок 2.14 – Построение линии под углом 67^0

При необходимости можно изменить шаг углов, вместо 90^0 поставить 15^0 . И тогда через каждые 15^0 на экране будет отражаться трассирующая линия.

Объектное отслеживание (OTRACK)

Режим объектного отслеживания позволяет задавать точку, отслеживая расстояние и угол относительно точки, указанной фиксированной объектной привязкой. Объектное отслеживание облегчает выбор точек, которые лежат на линиях отслеживания, проходящих через характерные точки объектов. Захваченная точка помечается маркером в виде маленького знака "плюс" (+). После захвата точки по мере передвижения курсора появляются вертикальные, горизонтальные или полярные линии отслеживания (пунктирные линии), проходящие через данную точку. Для использования объектного отслеживания необходимо наличие включенных режимов объектной привязки.

Включить/выключить действие объектного отслеживания можно при помощи функциональной клавиши F11, или нажав на кнопку .

Пример. Построить прямоугольник 100×70 , используя объектную привязку. Включить режим ОРТО. При вызове команды Отрезок первую точку указать произвольно. Направить курсор вправо и набрать на клавиатуре 100, затем направить курсор вверх и набрать 70. При построении третьего отрезка направить курсор на начало первого отрезка, пунктирная линия покажет след этой точки, и на пересечении двух пунктирных линий

щелкнуть мышкой (рис. 2.15). Соединить третью и первую точки, используя привязку.

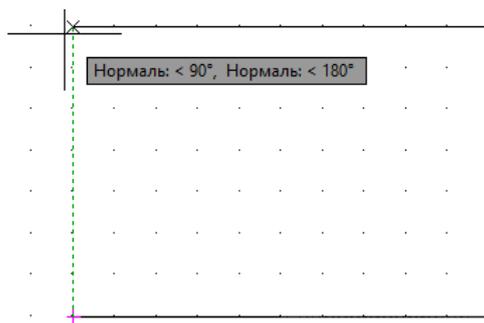


Рисунок 2.15 – Использование объектной привязки

Режим отображения веса (толщины) линии (LWT)

Кнопка Вес включает/выключает режим отображения веса (толщины) элементов чертежа. Вес линии – это ширина, с которой линия будет выводиться на печать.

Динамический ввод (DYN)

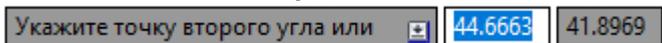
Функция динамического ввода обеспечивает командный интерфейс в области курсора, позволяющий удерживать фокус в области построения.

Когда параметр "Динамический ввод" включен, сведения, отображаемые в подсказках, рядом с курсором динамически обновляются по мере перемещения курсора. Когда команда активна, подсказки обеспечивают место для ввода, осуществляемого пользователем. Настроить режим динамического ввода можно при помощи четвертой вкладки Динамический ввод (рис. 2.16) диалогового окна Режимы рисования.

Включение и отключение функции динамического ввода происходит при нажатии на кнопку ДИН. Чтобы временно ее отключить, нажмите клавишу F12. Функция динамического ввода содержит три компонента: ввод с помощью мыши, ввод размеров и динамические подсказки.

Ввод с помощью мыши

Когда ввод с помощью мыши включен и команда активна, положение курсоров в форме перекрестия отображается в виде координат в подсказке возле курсора. Для перехода от одного значения координат к другому используется клавиша ТАВ. Значения координат можно вводить и в подсказке, и в командной строке



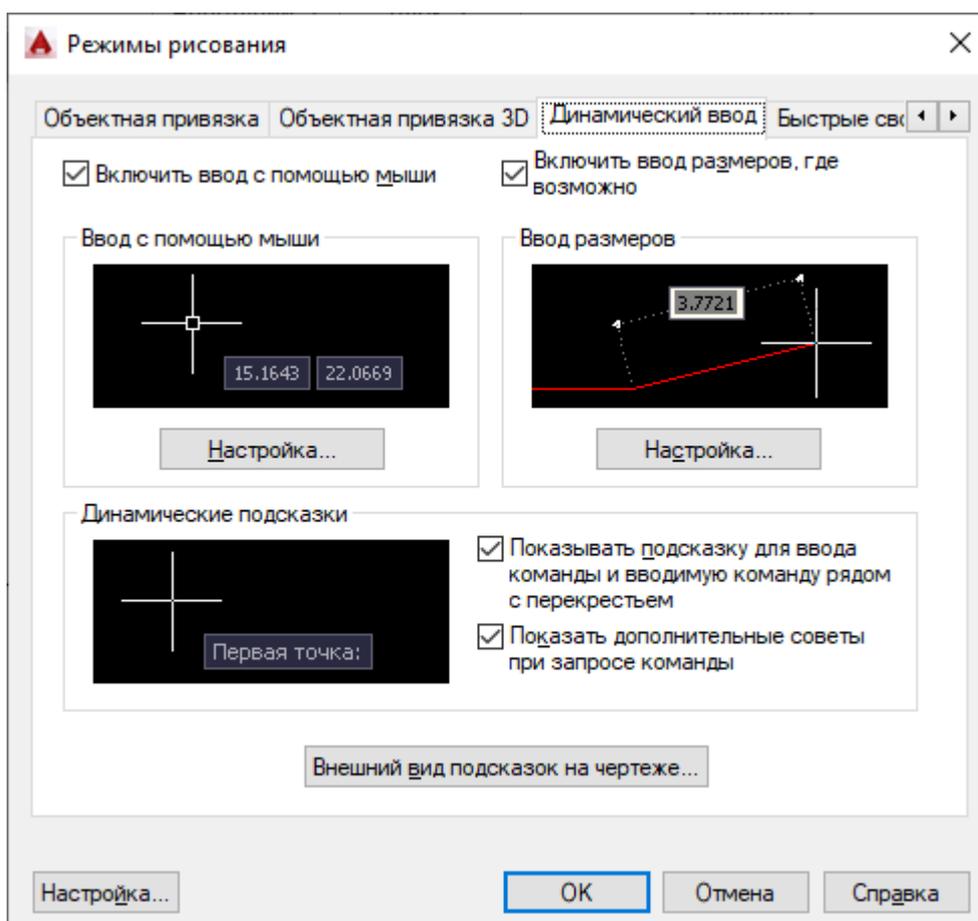


Рисунок 2.16 – Настройка динамического ввода

Ввод размеров

Когда функция ввода размеров включена, в ответ на запросе второй точки подсказки отображают линейные и угловые значения. Значения в размерных подсказках изменяются по мере перемещения курсора.

После ввода первого размера и нажатия клавиши TAB в поле отображается значок блокировки, а курсор ограничен введенным значением. 

Динамические подсказки

Когда функция динамических подсказок включена, запросы отображаются в подсказке возле курсора (рис. 2.17). Ответ можно вводить и в подсказке, и в командной строке. Нажмите клавишу СТРЕЛКА ВНИЗ для просмотра и выбора параметров. Нажмите клавишу СТРЕЛКА ВВЕРХ для отображения последних введенных команд.

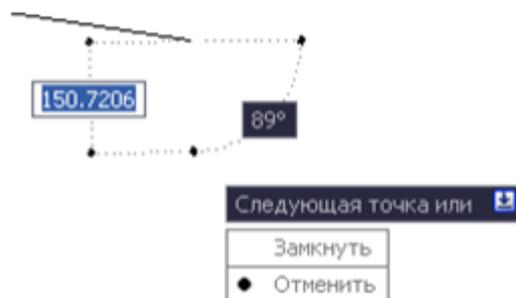


Рисунок 2.17 – Динамические подсказки

2.1.4 Графические примитивы как основа изображений

Структура команд AutoCAD

В AutoCAD имеется множество команд с различной структурой. В зависимости от того, как команды строят диалог с пользователем, их можно разделить на следующие типы:

Сложные команды, которые после вызова предлагают уточнить режим выполнения. Такие режимы называются опциями. Они располагаются в строке запроса команды после служебного слова "или" в квадратных скобках и отделены друг от друга наклонной прямой [... /.../...]. Чтобы выбрать опцию достаточно щелкнуть мышкой по названию или ввести в командной строке только выделенную большую букву, которая может находиться не в начале, а в середине названия опции. Регистр значения не имеет. Например, при построении эллипса в командной строке появляется запрос:

Команда: `_ellipse`

Конечная точка оси эллипса или [Дуга/Центр]: для вызова опции необходимо набрать на клавиатуре "d" или "c" и подтвердить выбор клавишей "Enter".

2. Команды, работающие с предварительной настройкой, т.е. использующие "стиль". По умолчанию команды работают со стилем Standard. Можно создать несколько собственных стилей, по мере необходимости делая один из них текущим. Стиль создается заранее. Примером команды со стилем является команда Текст. В ней от выбранного стиля зависят шрифт, высота и наклон букв.

3. Команды, работающие с помощью диалоговых окон. В диалоговом окне предлагается выбрать все параметры настройки команды и в любой последовательности. Примером может служить команда Штриховка.

Команды построения графических примитивов.

Чертеж в AutoCAD формируется из набора имеющихся в системе графических примитивов. К ним относятся: отрезки, дуги, окружности, прямоугольники, многоугольники и эллипсы.

Графический примитив – это элементарный геометрический объект, создаваемый одной командой. Из совокупности графических прими-

тивов в дальнейшем формируются сложные геометрические фигуры.

Команды построения примитивов находятся в системном меню Рисование либо на панели Рисование.

 **Точка** (Point) – фиксация точки. Объекты-точки рекомендуется использовать в качестве геометрических опорных узлов для объектной привязки и относительных смещений.

Вызов команды: Рисование – Точка – Одиночная (Несколько).

Запросы: Команда: `_point`

Текущие режимы точек: `PDMODE=0 PDSIZE=0.0000` Укажите точку: *указать точку любым известным способом* Завершение команды - Esc.

Режим `PDMODE` отвечает за вид отображения точки, режим `PDSIZE` – за ее размер. В указанном месте появится точка. Форма и размер точки определяются ее стилем. Стиль точки можно определить командой `Формат – Отображение точек` (рис. 2.18).

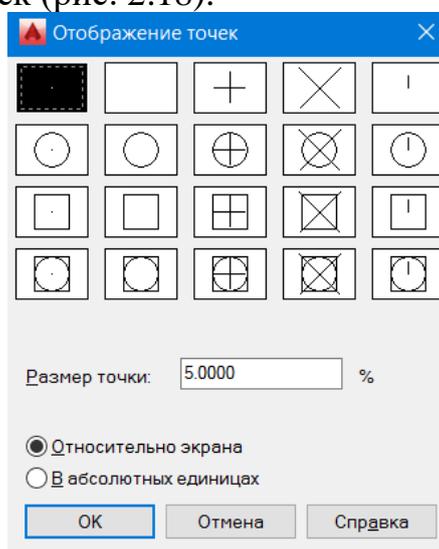


Рисунок 2.18 – Отображение точек

 **Отрезок** (Line) – создание ломаной линии, состоящей из отдельных отрезков.

Каждый линейный сегмент из набора сегментов может редактироваться отдельно. Построение выполняется точным указанием конечных точек каждого отрезка. Пользователь может:

- вводить значения координат конечной точки с использованием либо абсолютных, либо относительных координат;

- задать объектную привязку относительно имеющегося объекта. Например, в качестве одной конечной точки отрезка можно задать центр окружности либо середину любой линии;

- использовать шаговую привязку.

Запросы:

Команда: `_line` Первая точка: *указать точку любым известным способом*

Следующая точка или [Замкнуть/Отменить]: *указать точку или вы-*

брать одну из опций, указанных в квадратных скобках

Завершение команды - нажатие одной из клавиш: Esc, пробел или Enter.

Опции:

Отменить - удаление последнего сегмента последовательности отрезков (отмена предыдущего шага).

Замкнуть - построение последнего замыкающего отрезка, конец которого соединяется с первой введенной точкой. После чего последовательность отрезков приобретает вид замкнутого многоугольника. Опцию можно использовать после построения последовательности из двух или более отрезков.

Для повторного вызова последней команды можно нажать клавиши ENTER или ПРОБЕЛ.



Прямая (Xline) – создание бесконечных линий, проходящих через указанную точку.

Линии, не имеющие концов в обоих направлениях, называются прямыми. Бесконечные линии можно использовать в качестве вспомогательных при построении объектов. Эти линии часто строят на отдельном слое, который перед выводом на плоттер можно заморозить или отключить. Обычно, направление задается по двум точкам.

Запросы:

Команда: `_xline` Укажите точку или [Гор/Вер/Угол/Биссект/ Отступ]: *указать точку, которая будет являться серединой прямой или выбрать опцию Через точку: указать вторую точку, определяющую направление прямой*

Завершение команды - клавиши Esc, пробел или Enter.

Опции:

Гор - построение горизонтальной бесконечной прямой, проходящей через указанную точку. Прямая строится параллельно оси X

Вер - построение вертикальной бесконечной прямой, проходящей через указанную точку. Прямая строится параллельно оси Y.

Угол - построение бесконечной линии, проходящей под заданным углом через заданную точку.

Биссектр - построение бесконечной прямой, проходящей через указанную вершину угла и делящей его пополам. Прямая располагается в плоскости, определяемой тремя точками.

Отступ - построение бесконечной прямой параллельно выбранному линейному объекту. При выборе опции Отступ появляется следующий запрос:

Величина смещения или [Точка] <текущая>: *задать величину смещения, ввести *t* или нажать ENTER*

Величина смещения - задание расстояния, на котором должна располагаться бесконечная прямая относительно выбранного объекта.

Точка - построение бесконечной прямой, проходящей через за-

данную точку параллельно выбранному линейному объекту.

 **Дуга** (Arc) – создание дуги.

По умолчанию команда предлагает построить дугу по трем точкам (рис. 2.19).

При построении дуги надо помнить, что она строится против хода часовой стрелки, за исключением случая задания конечной точки с помощью угла.

Запросы:

- Команда: `_arc` Начальная точка дуги или [Центр]: *указать первую точку т.1*

- Вторая точка дуги или [Центр/Конец]: *указать вторую точку относительно середины отрезка т.2*

- Конечная точка дуги: *указать конечную точку т.3*

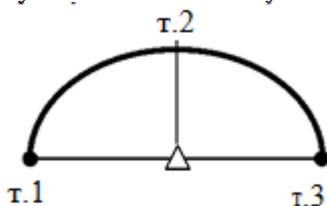


Рисунок 2.19 – Построение дуги по трем точкам

Последовательно выбирая опции можно построить дугу любым удобным для пользователя способом.

Существует несколько способов построения дуги. Рассмотрим чаще встречающиеся варианты.

Построение дуги по началу, центру и концу

Если известны начальная, центральная и конечная точки, то построение дуги можно начать с указания центральной точки. Центральная точка совпадает с центром окружности, частью которой является создаваемая дуга (рис. 2.20).

Команда: `_arc` Начальная точка дуги или [Центр]: *ц* Центр дуги: т.1

Начальная точка дуги: т.2

Конечная точка дуги или [Угол/Длина хорды]: т.3

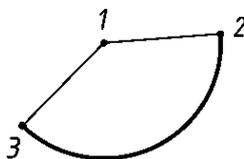


Рисунок 2.20 - Построение дуги по началу, центру и концу

Построение дуги по началу, центру и углу

Если известны начальная точка, центр дуги и центральный угол, то можно воспользоваться опцией "Начало, центр, угол" или "Центр, начало, угол" (рис.2.21).

Команда: `_arc` Начальная точка дуги или [Центр]: *т.1* Вторая

точка дуги или [Центр/Конец]: u

Центр дуги: $m.2$

Конечная точка дуги или [Угол/Длина хорды]: y Центральный угол: 90

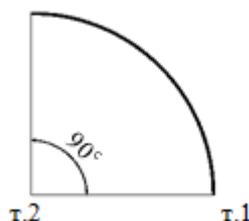


Рисунок 2.21 - Построение дуги по началу, центру и углу

Построение дуги по началу, концу и направлению/радиусу.

Если известны начальная и конечная точки, то можно воспользоваться методом "Начало, конец, направление" или "Начало, конец, радиус".

Радиус задается с клавиатуры или путем перемещения курсора в требуемом направлении с последующим вводом точного значения расстояния.

Если курсор перемещается вверх от начальной и конечной точек, то строится вогнутая дуга, как показано в данном случае. При перемещении курсора вниз строится выпуклая дуга.



Прямоугольник (Rectang) – создание замкнутых полилиний в форме прямоугольника.

Запросы: Команда: `_rectang`

Первый угол или [Фаска/Уровень/Сопряжение/Высота/Ширина]: Второй угол или [Площадь/Размеры/поВорот]:

Опции:

Фаска - задает размеры фаски для прямоугольника. Уровень - задает уровень для прямоугольника.

Сопряжение - задает радиус сопряжения для прямоугольника. Высота - задает трехмерную высоту для прямоугольника. Ширина - задает ширину полилинии для прямоугольника. Введенное значение становится текущей шириной полилинии, используемой при последующих вызовах команды ПРЯМОУГ.

Площадь - создает прямоугольник с использованием значений площади, а также значений длины или ширины.

Поворот - создает прямоугольник под заданным углом поворота. Размеры - построение прямоугольника по заданным значениям длины и ширины.

Существует несколько способов построения прямоугольников.

1.Построение прямоугольника по двум противоположным углам (рис.2.22).

Стороны прямоугольника параллельны осям X и Y текущей ПСК.

Команда: `_rectang`

Первый угол: 35, 25 (т.1 – начальная точка прямоугольника) Второй

угол: @100, 60 (т.2 – размер прямоугольника от точки 1)

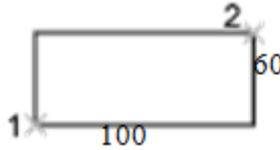


Рисунок 2.22 - Построение прямоугольника по двум противоположным углам

2. Построение прямоугольника по заданным значениям длины и ширины.

Команда: `_rectang`

Первый угол: *135,125 (т.1 – начальная точка прямоугольника)* Второй угол или [Площадь/Размеры/поВорот]: *p*

Длина прямоугольника `<10.0000>`: 100 Ширина прямоугольника `<10.0000>`: 60

Второй угол или [Площадь/Размеры/поВорот]: *указать мышкой вторую точку для определения ориентации прямоугольника относительно первой точки, т. е. указать положение прямоугольника.*

3. Построение прямоугольника со скругленными углами

Команда: `_rectang`

Первый угол или [Фаска/Уровень/Сопряж./Высота/Ширина]: *c* Радиус сопряжения прямоугольников `<0.0000>`: 10

Первый угол или [Фаска/Уровень/Сопряжение/Высота/Ширина]: *указать точку для первого угла*

Второй угол или [Площадь/Размеры/поВорот]: *любым способом указать размер прямоугольника, например @100,60.*

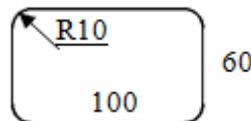


Рисунок 2.23 - Построение прямоугольника со скругленными углами

4. Построение прямоугольника со скошенными углами.

Команда: `_rectang`

Первый угол или [Фаска/Уровень/Сопряж./Высота/Ширина]: *f* Длина первой фаски прямоугольников `<10.0000>`: 20

Длина второй фаски прямоугольников `<10.0000>`: 20

Первый угол или [Фаска/Уровень/Сопряжение/Высота/Ширина]: *указать точку для первого угла*

Второй угол или [Площадь/Размеры/поВорот]: *любым способом указать размер прямоугольника, например @100,60*

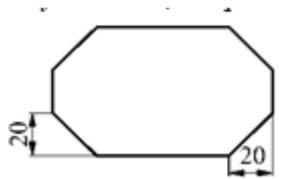


Рисунок 2.24 - Построение прямоугольника со скошенными углами



Окружность (Circle) – создание окружности.

Запросы:

- Команда: `_circle` Центр круга или [3Т/2Т/ККР (кас кас радиус)]: *укажите точку центра окружности или выберите опцию для построения окружности другим способом*

- Радиус круга или [Диаметр]: *по умолчанию задается радиус*

Опции:

2Т – построение окружности по двум точкам на диаметре;

3Т – построение окружности по трем точкам, лежащим на окружности;

ККР – построение окружности по двум касательным и радиусу.

Построение окружности по центру и радиусу (или диаметру)

1. Радиус круга или [Диаметр]: *задайте значение радиуса или введите d .*

2. Диаметр круга: 50.

Построение окружности, касающейся двух объектов (ккр). Включается режим объектной привязки «Касательная».

Команда: `_circle` Центр круга или [3Т/2Т/ККР (кас кас радиус)]: ККР

- Выбрать первый объект, которого касается окружность. - Выбрать второй объект, которого касается окружность. - Задать радиус круга.

Построение окружности по трем точкам (рис.2.25).

Команда: `_circle` Центр круга или [3Т/2Т/ККР (кас кас радиус)]: 3т

- Первая точка круга: т.1, середина одной стороны

- Вторая точка круга: т.2, середина второй стороны

- Третья точка круга: т.3, середина третьей стороны

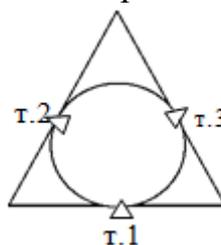


Рисунок 2.25 - Построение окружности по трем точкам

Построение окружности по двум конечным точкам ее диаметра Команда: `_circle` Центр круга или [3Т/2Т/ККР (кас кас радиус)]: 2т

- Выбрать первую точку диаметра. - Выбрать вторую точку диаметра.

- Первая конечная точка диаметра круга: т.1
- Вторая конечная точка диаметра круга: т.2



Эллипс (Ellipse) – создание эллипсов.

Построение эллипсов производится путем задания двух осей. Длинная ось эллипса называется его большой осью, короткая - малой осью (рис.2.26).

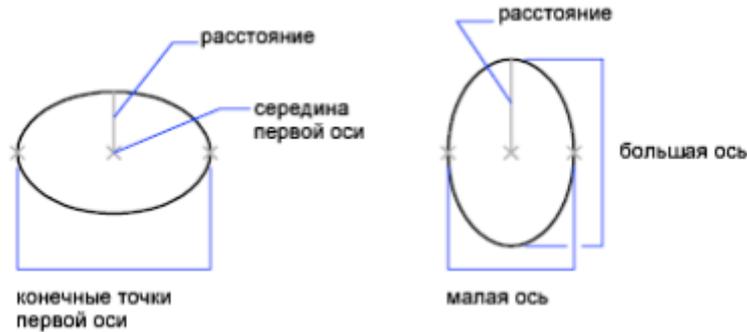


Рисунок 2.26 – Построение эллипса

Запросы: Команда: `_ellipse`

- Конечная точка оси эллипса или [Дуга/Центр]: *указать первую точку одной оси или выбрать одну из опций*
- Вторая конечная точка оси: *указать вторую точку этой же оси*
- Длина другой оси или [Поворот]: *указать половину другой оси*

Опции:

Центр – Построение эллипса по указанной точке центра.

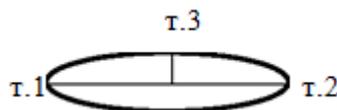
Дуга – Построение эллиптической дуги. Угол поворота первой оси задает поворот всей эллиптической дуги

Поворот - отношение длин осей эллипса задается путем поворота круга относительно первой оси. Чем больше угол поворота (он должен лежать в диапазоне от 0 до 89,4 градусов), тем больше отношение малой оси к большой. Если угол равен 0, строится круг.

Построение эллипса по конечным точкам одной оси и половине длины другой оси

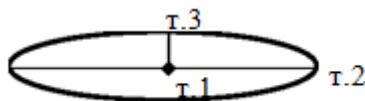
Команда: `_ellipse`

- Конечная точка оси эллипса или [Дуга/Центр]: *указать точку т.1 одной оси*
- Вторая конечная точка оси: *указать точку т.2 этой же оси*
- Длина другой оси или [Поворот]: *указать т.3 или ввести расстояние, равное половине малой оси*



Команда: `_ellipse`

- Конечная точка оси эллипса или [Дуга/Центр]: *указать точку т.1 одной оси*
- Вторая конечная точка оси: *указать точку т.2 этой же оси*
- Длина другой оси или [Поворот]: *указать т.3 или ввести расстояние равное половине малой оси*



Многоугольник (Polygon) – создание равносторонних многоугольников.

С помощью команды Многоугольник можно создавать замкнутые полилинии с числом сторон равной длины от 3 до 1024..

Запросы:

Команда: `_polygon` Число сторон <б>: *ввести число сторон*

- Укажите центр многоугольника или [Сторона]: *указать центр или выбрать опцию*
- Задайте опцию размещения [Вписанный в окружность/Описанный вокруг окружности] <В>: *задать опцию*
- Радиус окружности: *указать радиус*

Методы создания многоугольников:

1.

остроение вписанного (описанного) многоугольника

Команда: `_polygon` Число сторон <б>: *введите число сторон*

- Укажите центр многоугольника или [Сторона]: *укажите центр многоугольника*
- Задайте опцию размещения [Вписанный в окружность/ Описанный вокруг окружности] <В>: *введите в для построения многоугольника, вписанного в круг с заданным центром или о для построения многоугольника, описанного вокруг круга.*
- Радиус окружности: *задайте радиус*

2. Построение многоугольника с заданной стороной

- Команда: `_polygon` Число сторон <б>: *введите число сторон*
- Укажите центр многоугольника или [Сторона]: *укажите параметр с.*
- Первая конечная точка стороны: *укажите начальную точку одной стороны многоугольника (т.1)*
- Вторая конечная точка стороны: *укажите конечную точку этой же стороны или введите значение длины стороны.*

2.1.5 Сложные примитивы. Работа с OLE-объектами

Команды построения сложных примитивов



Полилиния (Pline) – создание связанной последовательности сегментов.

Полилиния представляет собой последовательность сегментов; все эти сегменты являются единым объектом. Полилинии используются, если предполагается работа с набором сегментов как с целым. Полилинии могут состоять из линейных и дуговых сегментов, а также из любых их сочетаний. У полилиний можно изменять ширину и кривизну сегментов (такие операции не возможны для простых отрезков).

Запросы: Команда: `_pline`

- Начальная точка: *указать начальную точку* Текущая ширина Полилинии равна 0.0000

- Следующая точка или [Дуга/Полуширина/длИна/Отменить/ Ширина]: *указать следующую точку или ввести одну из опций.*

Для выхода из команды нажать Esc или ENTER.

Опции:

Дуга - полилинии могут иметь дуговые сегменты. Каждый из таких сегментов строится из конечной точки предыдущего сегмента. Дуги описываются заданием угла, центра, направления или радиуса. Кроме того, дугу можно построить указанием второй и конечной точек.

Ширина – задается ширина следующего создаваемого сегмента полилинии. При указании значения 0 строится тонкий сегмент с нулевой шириной. Любые ненулевые положительные значения ширины приводят к созданию широких линий, которые отображаются закрашенными или в виде контуров, в зависимости от состояния режима закрашки. Сегменты могут сужаться или расширяться, если значения ширины в начальной и конечной точках различны. Эти опции доступны после указания начальной точки при построении полилинии. Начальные и конечные точки широких сегментов полилинии располагаются на центральных осевых линиях сегментов.

Полуширина – задается расстояние от осевой линии широкой полилинии до ее края.

длИна – длина линейного сегмента.

Замкнуть - построение замкнутой полилинии для создания многоугольника. Для замыкания полилинии нужно указать начальную точку последней построенной стороны объекта, ввести z (Замкнуть) и нажать ENTER.

Отменить – отмена предыдущего ввода.

Линейный – построение линейного сегмента.

Построение полилинии из линейных и дуговых сегментов

1. Команда: `_pline`
2. Начальная точка: *указать начальную точку*
3. Следующая точка или [Дуга/Полуширина/длИна/Отменить/ Ширина]: *указать следующую точку или ввести одну из опций.*
4. Указать следующую точку сегмента:

- о для включения режима отрисовки дуг ввести д (Дуга).
- о для возврата в режим построения линейных сегментов ввести л (Линейный).

5. При необходимости построить последующие сегменты полилинии.

6. Для завершения построения нажмите ENTER или введите з, если нужно замкнуть полилинию.

Создание широкой полилинии

1. Команда: `_pline`
2. Начальная точка: *указать начальную точку*
3. Следующая точка или [Дуга/Полуширина/длина/Отменить/ Ширина]: *ввести ш (Ширина)*
4. Начальная ширина <0.0000>: 5 5. Конечная ширина <5.0000>:
6. Продолжать указание конечных точек для последующих сегментов.
7. Для завершения построения нажмите ENTER или введите з, если нужно замкнуть полилинию.

Построение полилинии, состоящей из линейных и дуговых сегментов разной ширины.

1. Команда: `_pline`
2. Начальная точка: *указать начальную точку*
3. Следующая точка или [Дуга/Полуширина/длина/Отменить/ Ширина]: ш
4. Начальная ширина <0.0000>: 9 5. Конечная ширина <9.0000>: 0
6. Следующая точка или [Дуга/Полуширина/длина/Отменить/ Ширина]: *указать вторую точку линейного сегмента*
7. Следующая точка или [Дуга/Полуширина/длина/Отменить/ Ширина]: д(дуга)
8. Конечная точка дуги или [Угол/Центр/Замкнуть/ Полуширина /Линейный/Радиус/Вторая/Отменить/Ширина]: ш
9. Начальная ширина <0.0000>: 0 10. Конечная ширина <9.0000>: 9
11. Конечная точка дуги: *указать конечную точку*
12. Конечная точка дуги или [Угол/Центр/Замкнуть/Направление/Линейный/Радиус/Вторая/Отменить/Ширина]: л (*линейный*)
13. Следующая точка или [Дуга/Замкнуть/Полуширина/длина/ Отменить/Ширина]: ш
14. Начальная ширина <10.0000>: 9 15. Конечная ширина <10.0000>: 0
16. Следующая точка или [Дуга/Замкнуть/Полуширина/длина/ Отменить/Ширина]: *указать конечную точку*
17. Для завершения построения нажмите ENTER



Сплайн (Spline) - гладкая кривая, проходящая вблизи (в пределах заданной погрешности) некоторого набора точек.

Сплайны применяются при рисовании кривых произвольной формы; например, горизонталей в географических информационных системах. Сплайны строятся путем задания координат определяющих точек. Сплайны могут быть замкнутыми; при этом совпадают как конечная и начальная точки, так и направления касательных в них.

Запросы: Команда: `_spline`

- Первая точка или [Объект]: *указать начальную точку* - Следующая точка: *указать следующую точку*

- Следующая точка или [Замкнуть/Допуск] <касательная в начале>: *для выхода из команды нажать несколько раз ENTER.*

Опции:

Первая точка - создает сплайн по заданным точкам с использованием математики неоднородных рациональных B-сплайнов (NURBS).

Объект - преобразование сглаженных сплайнами полилиний в эквивалентные им сплайны. Если значение системной переменной DELOBJ = 1, то исходные объекты будут удалены из рисунка, если DELOBJ = 0 - оставлены в рисунке.

Следующая точка - использование этой опции позволяет продолжать ввод точек, добавляя к кривой сплайна новые сегменты, до тех пор, пока не будет нажата клавиша ENTER. Для отмены последней введенной точки следует ввести отменить. После нажатия ENTER отобразится запрос на указание направления касательной в начале сплайна.

Замкнуть - замыкает кривую сплайна, задавая конечную точку совпадающей с точкой начала, и сглаживает кривую в точке соединения.

Касательная в начале - задание направления касательной в начале (и, далее, в конце) сплайновой кривой. Указанное направление касательной определяет наклон сплайна в начальной (конечной) точке.

Команды редактирования сложных примитивов

Команды редактирования сложных примитивов находятся в системном меню Редактирование-Объект-Полилиния (Сплайн) либо на панели Рисование-2.

Редактирование полилиний



ПОЛРЕД - команда редактирования полилинии.

Вызвать команду редактирования полилинии можно, выбрав меню "Редактировать" Объект "Полилиния" или выбрав полилинию для редактирования, щелкнуть правой кнопкой мыши в области рисования и выбрать "Редактирование полилинии".

Запросы: Команда: `_redit`

- Выберите полилинию или [Несколько]: *Выбрать одну полилинию для редактирования. Если необходимо выбрать две и более линий, вводим параметр "Несколько", выделяем необходимые линии и завершаем выбор клавишей "Enter".*

Если выбранный объект является линией или дугой, отобразится следующий запрос: Выбранный объект - не полилиния.

- Преобразовать отрезки и дуги в полилинии [Да/Нет]?
<Д>Ввести д или н или нажать клавишу ENTER

При ответе **да** такой объект преобразуется в двумерную полилинию, состоящую из одного сегмента. Если значение системной переменной PEDITACCEPТ равно 1, то такой запрос не выдается, и выбранный объект автоматически преобразуется в полилинию.

- Задайте опцию [Замутить/Добавить/Ширина /Вершина/ СГладить/СПлайн/Убратъ сглаживание/Типлин/Отменить]: *выбрать необходимую опцию*

Для выхода из команды нажать Esc или ENTER.

Опции редактирования:

Замкнуть - для замыкания разомкнутой полилинии.

Добавить - для объединения смежных отрезков, дуг или полилиний. Позволяет объединять несоприкасающиеся полилинии, если их конечные точки лежат на расстоянии, не превышающем заданное.

Если ранее было выбрано несколько объектов с помощью параметра "Несколько", отобразится следующий запрос:

Значение допуска или [Способ]: *вести расстояние или с Способ* - задание способа объединения выбранных полилиний.

Удлинить – объединение выбранных полилиний удлинением или обрезкой сегментов до ближайших конечных точек.

Добавить – объединение выбранных полилиний добавлением прямолинейного сегмента между ближайшими конечными точками.

Оба - объединение выбранных полилиний удлинением или обрезкой (если возможно). Если это невозможно сделать с помощью этих операций, то объединение выбранных полилиний производится добавлением прямолинейного сегмента между ближайшими конечными точками.

Ширина - для задания новой единой ширины для всей полилинии.

Вершина - для редактирования вершин.

СГладить - для сглаживания полилинии дугами, проходящими через все вершины.

СПлайн - для сплайновой аппроксимации полилинии. Вершины выбранной полилинии используются как контрольные точки кривой. Эта кривая носит название полилинии, сглаженной сплайном. Она проходит через первую и последнюю контрольную точки, если исходная полилиния не была замкнута. Кривая как бы подтянута к другим точкам, но не обязательно проходит через них.

Если исходная полилиния содержит дуговые сегменты, то при формировании каркаса сплайна они спрямляются.

Убратъ сглаживание - для удаления всех дополнительных вершин, вставленных при выполнении операций "Сгладить" и "Сплайн", и выпрямление всех сегментов полилинии.

Типлин - для включения или отключения непрерывной генерации

образца типа линий.

Отменить - для отмены действий и возврата полилинии в исходное состояние, которое было на момент вызова команды ПОЛРЕД.

Пример. Дана фигура, состоящая из 6 отрезков. Два отрезка не соединены между собой на расстоянии 8,5 единиц (рис. 2.27).

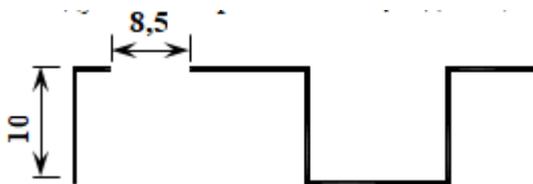


Рисунок 2.27 – Редактирование полилинии

Необходимо преобразовать ее в одну полилинию, а затем отредактировать, используя необходимые опции.

1. Опция **Добавить** – объединение смежных отрезков в одну полилинию

Команда: `_pedit`

Выберите полилинию или [Несколько]: *вводим опцию "H", выбираем отрезки и завершаем выбор клавишей "Enter"*.

Преобразовать отрезки и дуги в полилинии? [Да/Нет]: *Да* Задайте опцию [Замкнуть/ Добавить/ Ширина /Вершина/

СГладить/ СПлайн/ Убрать сглаживание/Типлин/ Отменить]: *Добавить*

Способ добавления = *Удлинить*

Значение допуска или [Способ] `<0.0000>`: *10* Для выхода из команды нажмите Esc или ENTER.

2. Опция **СПлайн** – сплайновая аппроксимации полилинии.

3. **СГладить** - для сглаживания полилинии дугами, проходящими через все вершины.

Редактирование сплайна



РЕДСПЛАЙН - редактирование сплайнов или сглаженных сплайнами полилиний.

Вызвать команду редактирования сплайна можно, выбрав меню "Редактировать" Объект "Сплайн" или выбрать сплайн для редактирования, щелкнуть правой кнопкой мыши в области рисования и выбрать «Редактирование сплайна».

Команда **РЕДСПЛАЙН** автоматически преобразует сглаженные сплайнами полилинии в настоящие сплайны.

Запросы:

Команда: `_splinedit`

- Выберите сплайн: *при выборе сплайна (сглаженной сплайном поли-*

линии) его определяющие (управляющие) точки выделяются ручками.

- Задайте опцию [Определяющие/Замкнуть/Перенести/Исправить/Обратно/Отменить]: *выбрать необходимую опцию*

Для выхода из команды нажать Esc или ENTER.

Опции редактирования:

Определяющие – редактирование определяющих сплайн данных, используя следующие опции:

Добавить - добавление в сплайн определяющих точек. После выбора точки РЕДСПЛАЙН подсвечивает ее и следующую за ней; новая точка размещается между двумя этими точками.

Открыть - размыкает замкнутый сплайн.

Замкнуть – замыкает разомкнутый сплайн и сглаживает кривую в точке соединения. Если начальная и конечная точки сплайна имели одинаковое положение, эта опция просто сглаживает кривую в обеих точках. Если выбранный сплайн замкнут, опция «Замкнуть» заменяется на «Разомкнуть».

Перенести – Переносит определяющую точку в новое положение.

Исправить – изменение формы сплайна, используя следующие опции:

Добавить управляющую точку - увеличивает число управляющих точек на участке сплайна. РЕДСПЛАЙН добавляет управляющую точку вблизи точки, выбранной на сплайне, между двумя другими управляющими точками, действующими на эту часть сплайна.

Повысить порядок - увеличивает порядок сплайна. Ввод большего (по сравнению с текущим) значения порядка ведет к увеличению числа управляющих точек. Максимальное значение порядка равно 26.

Обратно – изменяет направление сплайна на противоположное.

Отменить – отменяет последнюю операцию редактирования.

Работа с OLE-объектами

Технологией внедрения объектов (OLE - Object Linking Embedding) называется процесс вставки в исходный документ фрагмента документа другого приложения с сохранением или без сохранения связей с исходным оригиналом.

OLE-объекты не могут быть выбраны одновременно с обычными графическими примитивами AutoCAD. Они вставляются как картинка.

Если OLE-объект содержит текст, то AutoCAD переводит его кегль в измерительные единицы, принятые в AutoCAD. Когда вы вставляете OLE-объект в чертеж, его масштаб подгоняется под размер страницы и выражается в измерительных единицах этого чертежа.

Вставка в чертеж AutoCad фрагмента документа другого приложения осуществляется командой Объект из главного меню Файл (рис. 2.28).

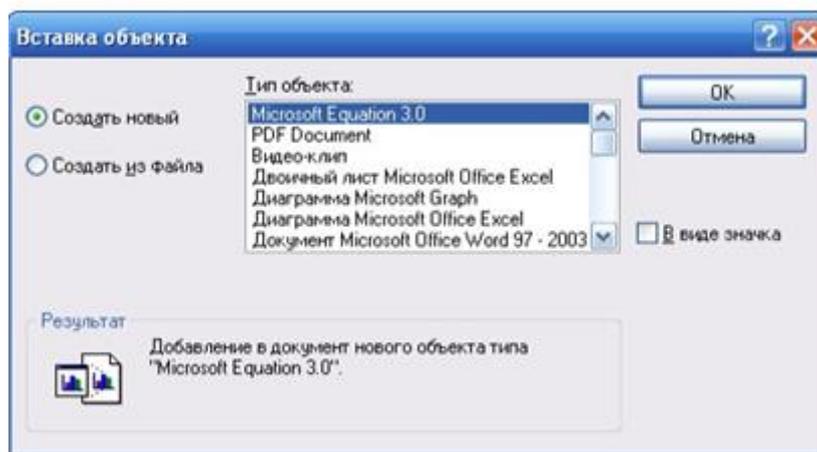


Рисунок 2. 28 – Вставка в AutoCad другого приложения

В центральной части диалогового окна в списке Тип объекта (Object Type) располагается список документов всех приложений Windows, установленных на компьютере и поддерживающих OLE-технологии (технологии связывания и внедрения).

Данная технология предусматривает два варианта вставки одного документа в другой: связывание и внедрение.

При связывании создается ссылка на другой файл документа, который можно редактировать, не выходя из основного документа.

При внедрении в рисунок вставляется содержимое другого документа, но связь с файлом-источником утрачивается.

В обоих случаях на время создания или редактирования вставленного документа в отдельное окно загружается его собственное приложение.

В левой части открытого диалогового окна Вставка объекта (Insert Object) можно выбрать одну из кнопок:

- 1) Создать новый (Create New)
- 2) Создать из файла (Create from File).

Первый вариант означает, что создается совершенно новый документ, который будет внедрен внутрь рисунка AutoCAD без создания файла. Выбор переключателя Создать из файла означает, что нужно будет указать файл с вставляемым документом.

Пояснить этот процесс можно на конкретных примерах.

Пример 1. Требуется в чертеж AutoCAD вставить математическую формулу, путем внедрения объекта "Microsoft Equation 3.0".

1. Вызываем команду Вставка - Объект.
2. В поле Тип объекта выбираем "Microsoft Equation 3.0" и указываем переключатель Создать новый.
3. В открывшемся редакторе формул набираем исходную формулу. Затем при помощи команды Файл – Выход и возврат в рисунок закрываем редактор формул.
4. Редактор формул закрывается и в чертеже AutoCAD появляется

математическая формула, обведенная рамкой.

Редактирование вставленной в AutoCAD формулы:

1. Щелкнув по формуле 2 раза, открыть Редактор формул.
2. Внести необходимые изменения, сохранить и закрыть Редактор формул командой Файл – Выход и возврат в рисунок.

Все внесенные изменения отразятся в рисунке AutoCAD.

Пример2. Требуется вставить в AutoCAD растровое изображение (фотографию). Рассмотрим два способа вставки изображения.

I способ.

1. Вставляем скопированный рисунок при помощи команды Вставка главного меню Файл либо вызываем в командной строке команду pasteclip.

2. В командной строке появляются запросы:

- Точка вставки <0,0>: *указать мышкой точку вставки*
- Масштаб <1>: *вести масштаб. Значение больше 1 увеличивает изображение, меньше 1 – уменьшает.*
- Угол поворота <0>: *вести значение угла поворота изображения*

II способ.

1. Растровое изображение можно вставить в AutoCAD при помощи команды «Вставка - Растровое изображение». При этом загружается окно, в котором выбираем необходимый файл и нажимаем кнопку Открыть.

2. Затем загружается диалоговое окно «Растровое изображение», в котором необходимо указать следующие параметры:

- Задание пути: *указывается один из типов путей к папке:*

Полный - полный путь к файлу растрового изображения

Без пути - задается только имя файла изображения.

Файл растрового изображения должен быть сохранен в одной папке с файлом текущего рисунка.

- Точка вставки: *вести координаты либо установить флажок «Указать на экране»*

- Масштаб: *задать масштаба или установить флажок «Указать на экране»,*

- Угол поворота: *задать угол поворота изображения.*

2.1.6 Основные свойства объектов

Слои являются основным средством упорядочивания на чертеже. Они позволяют сгруппировать информацию и упрощают управление различными свойствами, такими как типы линий, цвета и др. Слои напоминают лежащие друг на друге прозрачные листы кальки. Наложение этих листов друг на друга позволяет получить единое изображение.

Слои позволяют группировать однотипные объекты. Например, такие объекты, как вспомогательные линии, тексты, размеры и основные надписи можно разместить на отдельных слоях.

Послойная организация объектов позволяет:

- Легко подавлять и включать отображение объектов слоя на видо-вых экранах.
- Разрешать, запрещать и настраивать вывод объектов на печать.
- Назначать цвет одновременно всем объектам слоя.
- Задавать тип и вес линий по умолчанию для всех объектов слоя.
- Разрешать или запрещать редактирование объектов слоя.

Создание слоя

При создании нового рисунка автоматически создается слой, названный 0, которому присваиваются белый цвет, непрерывный тип линии Continuous, вес (толщина) линии Default. Этот слой не может быть удален и переименован. Кроме того, существует системный слой Defpoints, содержание которого не выводится на печать.

Управление установками свойств слоев осуществляется в диалоговом окне Диспетчер свойств слоев. Оно загружается из падающего меню Формат – Слой или на вкладке Главная – Слои (рис. 2.29).



Рисунок 2.29 – Панель Слои

Для создания нового слоя необходимо выбрать команду Свойства слоя. При этом открывается Окно диспетчера свойств слоев, которое состоит из двух панелей. На левой панели выводится дерево слоев, на правой – список слоев, относящихся к выбранному элементу дерева. Если в дереве выбрана строчка Все, то на правой панели выводятся все слои рисунка (рис. 2.30).

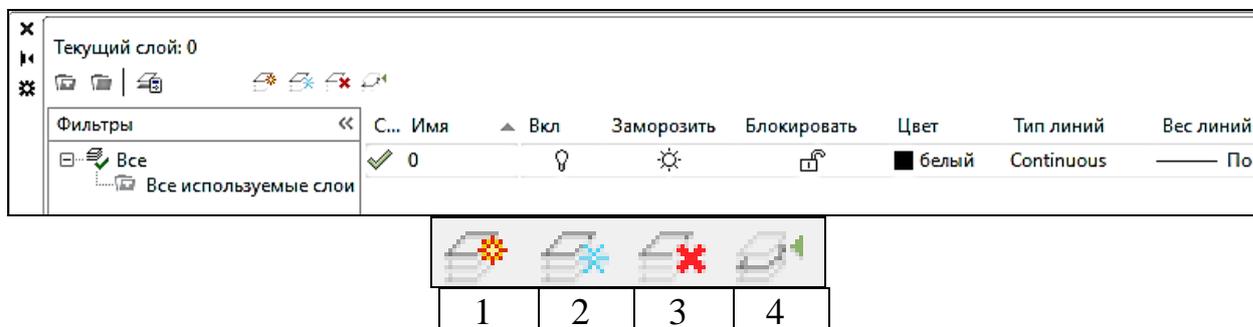


Рисунок 2.30 – Диспетчер свойств слоев

Опции команды:

- 1 – создание нового слоя;

2 - замораживание нового слоя на всех видовых экранах;

3 – удалить слой. Можно удалять только неиспользуемые слои. Нельзя удалить 0 слой и Defpoints, текущий слой.

Для создания нового слоя необходимо щелкнуть мышью по пиктограмме Создать слой (Alt+Д), находящейся в верхней части диалогового окна Диспетчера слоев. Создается слой, по умолчанию названный Слой1. Далее все новые слои автоматически именуются Слой2, Слой3 и т. д. - в порядке их создания. Чтобы присвоить слою уникальное имя, необходимо двойным щелчком мыши по текущему названию активизировать поле ввода текста, а затем набрать имя с клавиатуры и нажать клавишу Enter.

Если при создании нового слоя выделен один из имеющихся слоев, то новый слой наследует его свойства. При необходимости свойства нового слоя можно изменить.

Слои обладают следующими свойствами:

Статус – отображает состояние слоя: пустой, используемый или текущий.

Имя – вывод и редактирование имени выбранного слоя. Не следует давать имя слоя из нескольких букв или цифр понятных только разработчику чертежа, имя слоя должно нести смысловую нагрузку. Для использования фильтра по именам слоев (сортировки), слоям необходимо назначать одинаковые префиксы. За префиксом следует уточнение назначения слоя, например, 1этаж_стены, 1этаж_проводка и т.д.

В диспетчере слоев слои автоматически сортируются по названиям в алфавитном порядке.

Вкл (откл) - включение (отключение) видимости слоя. Если у слоя в этом столбце стоит значок "желтая лампочка", то слой считается включенным, если значок "синяя лампочка" - то выключенным. При включении или отключении слоев чертеж не регенерируется (перерасчет формул). AutoCAD не отображает на экране объекты,

расположенные на невидимых слоях, и не выводит их на плоттер. **Замороженный** - если у слоя в этом столбце стоит значок "желтое солнце", то слой считается размороженным, если значок "синяя снежинка" - замороженным. Объекты, расположенные на слое, который замораживается, становятся временно невидимыми, до тех пор, пока слой не будет разморожен. Любой слой, кроме текущего, может быть заморожен. Операция размораживания одного или более слоев приводит к регенерации чертежа и отнимает больше времени, чем простое включение и отключение слоев.

Блокировка - примитивы в заблокированном слое отображаются, но их нельзя редактировать. Блокированный слой можно сделать текущим, рисовать в нем, замораживать и применять к его примитивам команды справок и объектную привязку.

Цвет – список цветов, доступных для присвоения выбранному слою.

Тип линий - список типов линий. В окне Выбор типа линий отме-

титель нужную линию и нажать ОК для возврата в диалоговое окно диспетчера слоев. Для загрузки нужного типа линии необходимо щелкнуть по кнопке Загрузить. В открывшемся окне Загрузка типов линий (рис. 2.31) выбрать нужный тип линии и щелкнуть по кнопке ОК.

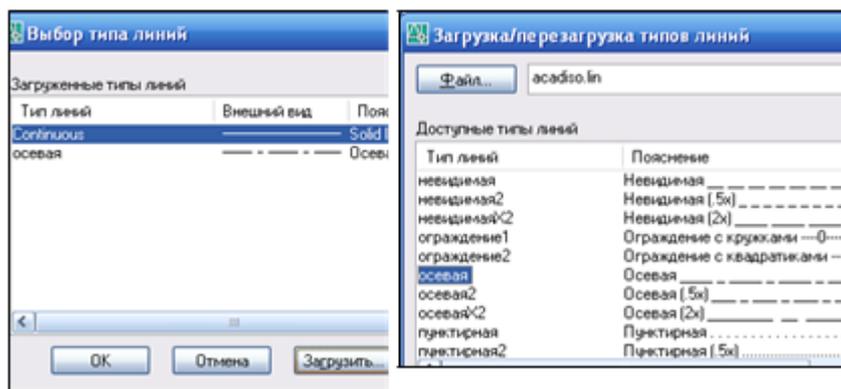


Рисунок 2.31 – Загрузка типов линий

Вес линий - вывод списка фиксированных весов линий (толщина линий при выводе на печать). На графическом экране примитивы отображаются без толщины линий, если кнопка режима ВЕС выключена.

Стиль печати - стиль печати для заданного слоя. При выводе на печать с помощью стилей печатей можно задавать различные типы соединений и окончания линий для различных весов линий (закругление или скос).

Печать – разрешение (запрет) вывода слоя на печать. Запрет печати удобно использовать для слоев, содержащих вспомогательные элементы построений. При этом перед выводом рисунка на печать такие слои не требуется отключать.

Пояснение – текстовое пояснение к слою.

Панель инструментов Слои предназначена для оперативного управления состоянием слоев. Она состоит из трех кнопок и одного выпадающего списка слоев.

Чтобы установить текущим слой, на котором находится выделенный объект, необходимо щелкнуть на кнопку Сделать слой объекта текущим.

В диспетчере свойств слоев можно выбрать два вида фильтра:

1. Фильтр слоев по свойствам определяется в диалоговом окне "Свойства фильтра слоев", где пользователь может выбрать любые из свойств слоев, определяющих фильтр.

Фильтр слоев ограничивает отображение имен слоев в окне "Диспетчера свойств слоев". Для фильтрации слоев по именам можно использовать специальные символы. Например, если необходимо отобразить только те слои, имена которых начинаются с символов mech, то надо ввести mech*.

Набор слоев в фильтре по свойствам может меняться при изменении свойств слоев.

2. Групповой фильтр слоев включает только те слои, которые были явно занесены туда пользователем. Если свойства слоев, назначенных данному фильтру, были изменены, то, несмотря на это, данные слои будут продолжать входить в этот фильтр.

Инвертирование фильтра слоев. Пользователь может инвертировать фильтр слоев. Например, если все элементы плана участка располагаются на слоях, имена которых содержат слово Участок, то для отображения всех остальных слоев необходимо создать фильтр, который отфильтровывает слои по строке *участок*, и установить флажок "Инвертировать фильтр".

Свойства объекта

Панель текущих свойств определяет свойства для вновь создаваемых примитивов (рис. 2.32).

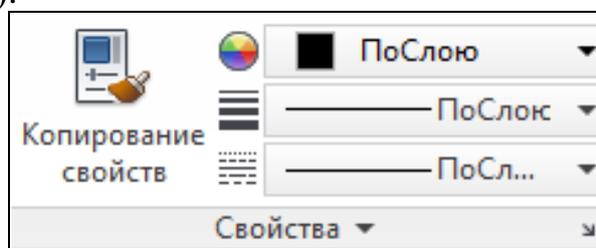


Рисунок 2.32 – Панель Свойства

Каждое свойство отображается в поле со списком. Чтобы назначить свойство текущим, необходимо открыть список и выделить необходимое значение.

Если в строке текущий цвет, тип или вес линии имеют значение По слою, то это означает, что они соответствуют настройкам цвета, типа и веса линии текущего слоя.

Универсальная команда PROPERTIES (ОКНОСВ) дает возможность редактирования любых свойств объектов, а также элементов геометрии (рис.2.33).

Команде соответствуют пункты Сервис – Палитра –Свойства или Редактирование – Свойства, горячие клавиши – Ctrl+1.

Существует возможность быстрого вызова этого окна. Если в рисунке нет выделенных объектов, то двойной щелчок по объекту выводит на экран окно PROPERTIES (СВОЙСТВА). Если в рисунке был выделен объект (не тот, по которому будет выполнен двойной щелчок) или объекты, то двойной щелчок также откроет окно.

Содержимое окна PROPERTIES (СВОЙСТВА) существенно зависит от того, какие объекты в данный момент выделены в рисунке.

Если нет выбранных объектов, то в раскрывающемся списке в верх-

ней части окна написано No selection (Ничего не выбрано) и окно показывает текущие установки рисования. Если в графическом экране отметить объект, то содержимое окна PROPERTIES (СВОЙСТВА) изменится и будет показывать свойства выбранного примитива, а раскрывающийся список укажет имя примитива. Когда выбраны сразу несколько объектов, то окно показывает только свойства, присутствующие у всех этих объектов (например, не у всякого примитива есть радиус, площадь).

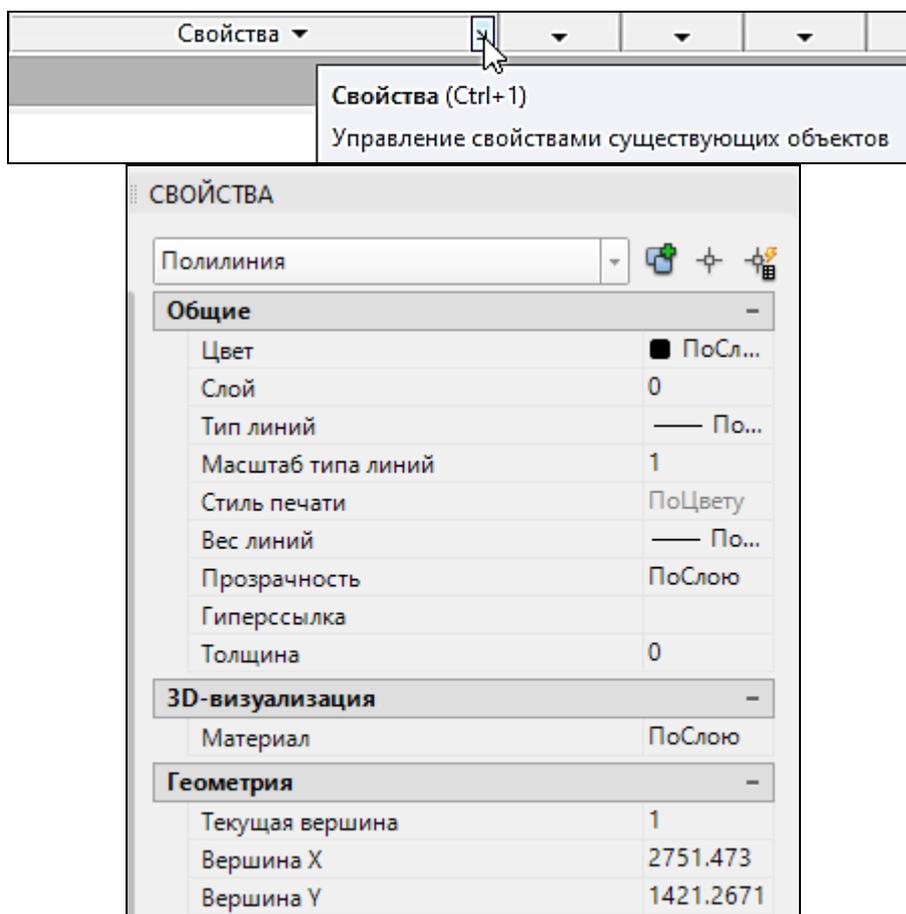


Рисунок 2.33 – Вызов и состав панели Свойства

Центральная часть окна PROPERTIES (СВОЙСТВА) оформлена в виде таблицы, имеющей вкладки. Список вкладок зависит от того, что в данный момент выбрано.

При выборе объекта в строке свойств автоматически отображается принадлежность к слою, цвет, тип и вес линии выделенного объекта. Выделив объект и выбрав из списка другое свойство, можно присвоить новое свойство этому объекту.

Пример. Нарисована осевая линия и т.к. маленький размер объекта, то не видны штрихи у осевой линии. Выделяем осевую линию, щелкаем по ней правой мышкой, и в контекстном меню выбираем Свойства. В открывшемся окне на вкладке Общие в строке Масштаб типа линии (по умолчанию = 1) набираем 0.5, т.е. изменяем масштаб осевой линии, штрихи будут в 2 раза меньше.

Каждый графический примитив обладает набором свойств: принадлежность к слою, цвет, тип линии, вес (толщина) линии, геометрические характеристики, масштаб типа линий и т.д. Под заглавием панели отображается название объекта. В зависимости от выбранного объекта отображаются различные характеристики.

Импортирование слоев из других чертежей осуществляется в диспетчере слоев (рис. 2.34) или в центре управления.

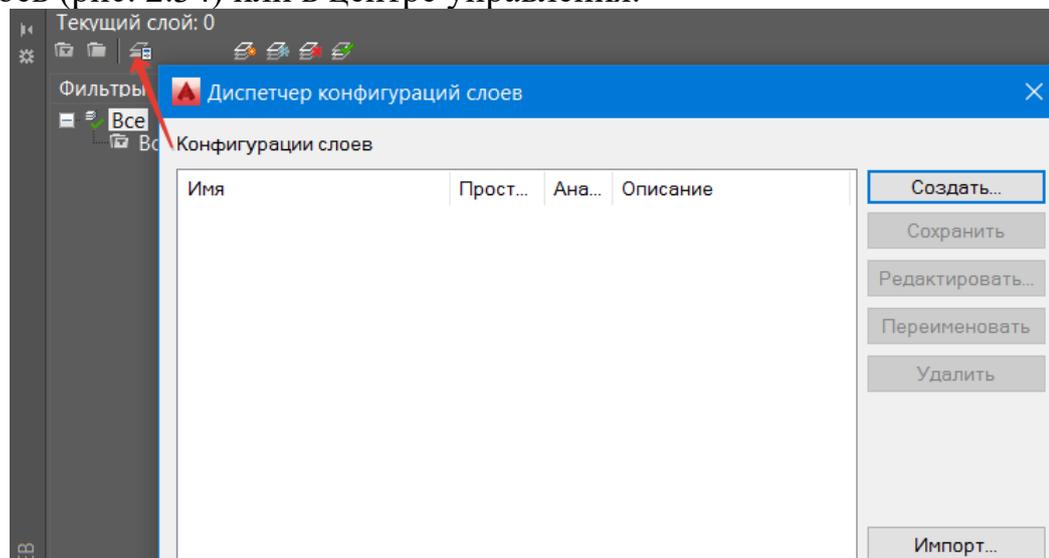


Рисунок 2.34 - Импортирование слоев из других чертежей

Центр управления

В практической деятельности очень важно иметь возможность использовать созданные ранее наработки. Центр управления AutoCAD позволяет использовать фрагменты чертежей многократно. Центр управления позволяет быстро находить, просматривать, вызывать и применять специфические компоненты, такие как блоки, слои, типы линий и пр. Причем эти операции осуществляются из открытых или закрытых файлов AutoCAD, независимо от того, где они находятся: на локальных или удаленных дисках. Достаточно просто выбрать нужные компоненты и перетащить их в текущий рисунок.

Центр управления AutoCAD загружается из падающего меню Сервис – Палитры – Центр управления (рис. 2.35), горячие клавиши - Ctrl+2.

AutoCAD делит окно Центр Управления на две части и показывает:

- в левой части древовидную структуру папок и файлов;
- в правой части - состав выбранного элемента.

Окно содержит следующие вкладки:

1. Folders (Папки) показывает диски и папки компьютера, для выбора файла и просмотра его элементов;
2. Open Drawings (Открытые рисунки) — показывает в левой части окна элементы открытого рисунка, а в правой части — имена этих эле-

ментов (имена типов линий);

3. History (Журнал) — показывает журнал последних операций в Центре управления.

4. DC Online (ЦУ Online) — доступ в Интернете в прямом режиме к библиотекам стандартных элементов на сайте Autodesk.

5. Search (Поиск), ищет файл по заданным признакам (слоям, блокам, например, по первой букве файла Ф*);

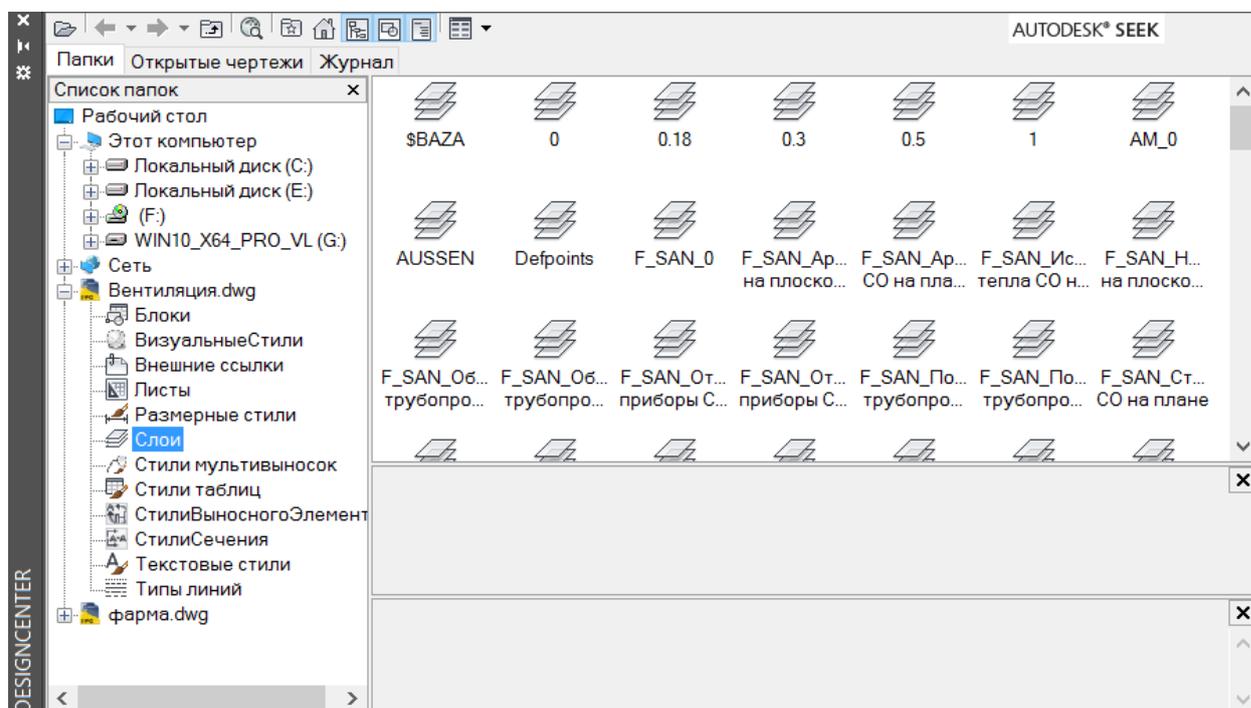


Рисунок 2.35 - Центр управления AutoCAD

На вкладках Открытые рисунки или Папки щелчок по значку "плюс" открывает содержимое рисунка или папки.

В состав рисунка входят следующие элементы:

- блоки (blocks);
- размерные стили (dimstyles);
- слои (layers);
- листы (layouts);
- типы линий (linetypes);
- текстовые стили (textstyles);
- внешние ссылки (xrefs).

Любые элементы, отображаемые в окне, могут быть перенесены в текущий рисунок простым перетаскиванием с помощью мыши. Достаточно просто выбрать мышкой нужные компоненты в правой части окна и перетащить их в текущий рисунок на графический экран. Копирование можно выполнить также двойным щелчком по выбранному элементу.

2.2 Редактирование объектов, размерные стили

2.2.1 Команды редактирования

Команды редактирования можно вызвать посредством соответствующих кнопок панели инструментов Редактирование (рис. 2.36).

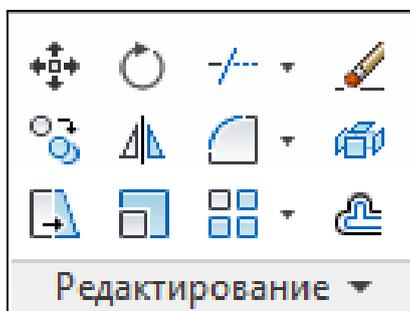


Рисунок 2.36 – Панель Редактирование

Команды изменения положения объектов

✚ ПЕРЕНЕСТИ (MOVE) - перемещение объектов на заданное расстояние в указанном направлении, не изменяя их размеров (рис.2.37).

Запросы в командной строке:

Команда: `_move`

– Выберите объекты: *Выбрать объекты любым способом и нажать ENTER по завершении выбора объектов*

– Базовая точка или [Перемещение]<Перемещение>: *Указать базовую точку*

– Вторая точка или <считать перемещением первую точку>: *Укажите новую точку или задайте в командной строке расстояние переноса.*

Опции:

Перемещение - две указанные точки задают направляющий вектор, который определяет, на какое расстояние и в каком направлении должны быть перемещены выбранные объекты.

Пример 1. Переместить прямоугольник к середине отрезка.

Команда: `_move`

– Выберите объекты: *Выбрать прямоугольник и нажать ENTER*

– Базовая точка или [Перемещение]<Перемещение>: *Указать середину нижнего основания прямоугольника*

– Вторая точка: *Указать середину отрезка*

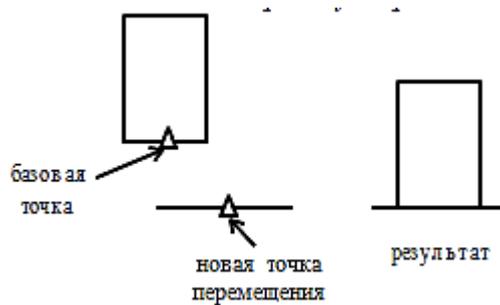


Рисунок 2.37 – Команда Перенести

ПОВЕРНУТЬ (ROTATE)  - поворот объектов вокруг базовой точки

Запросы:

Команда: `_rotate`

Сначала команда информирует о режимах измерения углов. Текущие установки отсчета углов в ПСК: $ANGDIR=против\ ч/с$, $ANGBASE=0$

– Выберите объекты: *выбрать объекты любым способом и нажать ENTER по завершении выбора объектов*

– Базовая точка: *указать точку, вокруг которой необходимо выполнить поворот объектов*

– Угол поворота или [Копия/Опорный угол] $<0>$: *ввести значение угла поворота (при положительном значении угла объект поворачивается против часовой стрелки, в противном случае – по часовой стрелке) или выбрать опцию.*

Опции:

Копия – исходный объект остается без изменений, а поворачивается лишь его копия.

Опорный угол – поворот по опорному углу. В ответ на следующие запросы указывается начальный угол и новый угол (возможно указание угла двумя точками на экране, в этом случае берется угол наклона линии, проходящей через эти две точки).

Опорный угол $<последний\ опорный\ угол>$: *Задать угол либо вводом значения или указанием двух точек*

Новый угол или [Точки] $<последний\ новый\ угол>$: *Задать новый абсолютный угол вводом значения или указанием двух точек*

Пример 2. Совмещение диагонали прямоугольника при повороте его вокруг одной из вершин с направлением к точке т.3 (рис.2.38).

Команда: `_rotate`

- Выберите объекты: *выбрать прямоугольник и нажать ENTER.*
- Базовая точка: *указать точку т.1*
- Угол поворота или [Копия/Опорный угол] $<0>$: **О**
- Опорный угол: *указать точку т.1, а затем точку т.2*
- Новый угол: *указать точку т.3*

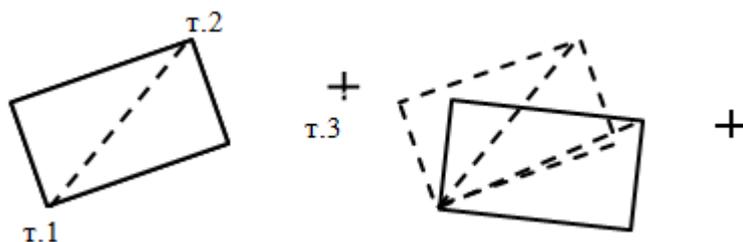


Рисунок 2.38 – Команда Поворот

Команды копирования объектов



КОПИРОВАТЬ (COPY) - копирование и отображение объекта в заданной области экрана.

Запросы:

Команда: `_copy`

– Выберите объекты: *выбрать объекты любым способом и нажать ENTER по завершении выбора объектов*

– Базовая точка или [Перемещение]: *указать базовую точку, относительно которой будет перемещаться объект или выбрать опцию*

– Укажите вторую точку или [Выход/Отменить]: *указать новое положение базовой точки или задать в командной строке расстояние переноса.*

Каждое смещение определяется относительно исходной базовой точки. Запрос на новое положение базовой точки выдается циклически и будет повторяться до тех пор, пока не нажать ENTER.

Опции:

Перемещение - две указанные точки задают направляющий вектор, который определяет, на какое расстояние и в каком направлении должны быть перемещены копируемые объекты.

Отменить – отмена предыдущего ввода.

Выход – завершение команды.

Пример 3.3. Копирование набора объектов с использованием привязок и указанием расстояния переноса (рис. 2.39).

Команда: `_copy`

– Выберите объекты: *выбрать объекты и нажать ENTER* – Базовая точка: *указать точку т.1*

– Укажите следующую точку или [Выход/Отменить]: *вторую точку указать координатой @-30,40*

– Укажите следующую точку или [Выход/Отменить]: *указать начало отрезка*

– Укажите следующую точку или [Выход/Отменить]: *указать середину отрезка*

– Укажите следующую точку или [Выход/Отменить]: *указать конец отрезка*

– Для выхода из команды нажать Enter.

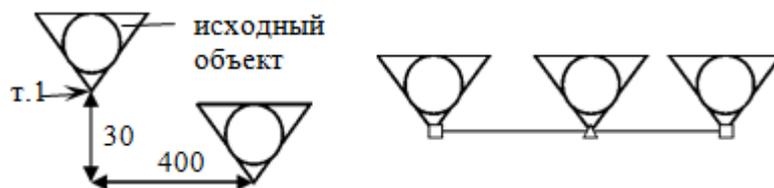


Рисунок 2.39 – Команда Копировать

 **ПОДОБИЕ (OFFSET)**. Команда применяется для построения параллельных отрезков, полилиний, дуг и т.д.

При вызове команды задается расстояние или точка, через которую должен пройти переносимый объект.

С помощью команды **ПОДОБИЕ** можно создать новый объект, подобный выбранному объекту. Подобные круги или дуги имеют диаметр, больший или меньший радиуса исходного, в зависимости от того, как задано смещение.

Если точка смещения расположена внутри периметра замкнутого объекта, то построенный объект будет меньше исходного, в противном случае – больше.

Команда не допускает предварительного выбора объектов.

Запросы:

Команда: `_offset`

- Укажите расстояние смещения или [Через/Удалить/Слой] <Через>: *ввести расстояние смещения или выбрать опцию*

- Выберите объект для смещения или [Выход/Отменить] <Выход>: *выберите один объект, к которому нужно построить параллельную копию, при этом нельзя использовать выбор объектов рамкой*

- Укажите точку, определяющую сторону смещения, или [Выход/Несколько/ Отменить] <Выход>: *указать точку курсором в направлении смещения*

Запрос на выбор точки выдается циклически и будет повторяться до тех пор, пока не нажать ENTER.

Опции:

Через - построение объекта, проходящего через заданную точку.

Удалить - удаляет исходный объект после смещения.

Слой - определяет положение смещенных объектов: на текущем слое или на слое исходного объекта.

Несколько - включает режим многократного смещения, при котором смещение повторяется с учетом текущего расстояния смещения.

Отменить – отмена предыдущего ввода. **Выход** – завершение команды.

Для построения подобного объекта путем задания смещения (рис. 2.40) необходимо:

1. Вызвать команду Подобие.
2. Задать с клавиатуры расстояние смещения: R .
3. Выбрать исходный объект: *в первом случае – разомкнутую полилинию, во втором – замкнутый контур (эллипс).*
4. Указать точку, определяющую сторону смещения: *в первом случае щелкнуть мышкой справа или слева от полилинии, во втором – внутри или вне эллипса.*
5. Выбрать следующий исходный объект или нажать ENTER для завершения команды.

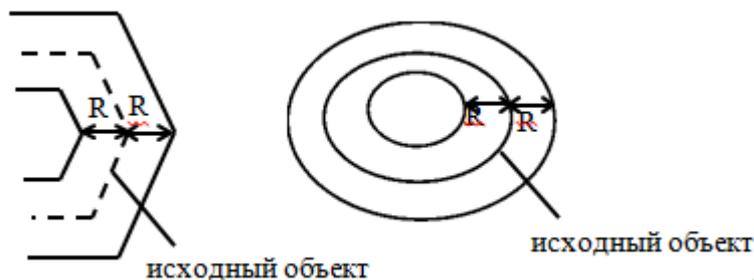


Рисунок 2.40 – Команда Подобие

Для построения подобного объекта, проходящего через точку необходимо:

1. Вызвать команду Подобие.
2. Указать расстояние смещения или [Через/Удалить/Слой] <Через>: *выбрать опцию Через.*
3. Выбрать объект для смещения или [Выход/Отменить] <Выход>: *выбрать исходный объект.*
4. Указать точку, через которую проходит объект, или [Выход/Несколько/Отменить]: *выбрать опцию Несколько*
5. Через точку или [Выход/Отменить] <следующий объект>: Указать точку, через которую должен проходить новый объект или ввести в командной строке значение смещения. Новое значение смещения будет отсчитываться от последнего построенного смещением объекта. *Ввести расстояние смещения 20*
6. Через точку или [Выход/Отменить] <следующий объект>: *указать середину отрезка*
7. Через точку или [Выход/Отменить] <следующий объект>: *80*
8. Выбрать следующий объект или нажать ENTER для завершения команды.



ЗЕРКАЛО (MIRROR) – зеркальное отображение объектов.

Зеркальное отображение позволяет создавать зеркальные копии объектов относительно заданной оси.

Функция зеркального отражения помогает быстро создавать сим-

метричные объекты, так как достаточно построение лишь половины объекта с последующим зеркальным отражением вместо построения целого объекта.

Запросы:

Команда: `_mirror`

- Выберите объекты: *Выбрать объекты любым способом; по окончании выбора нажать ENTER*

- Первая точка оси отражения: *Укажите точку 1*

- Вторая точка оси отражения: *Укажите точку 2*

- Удалить исходные объекты? [Да/Нет] <Н>: *Ввести **д** или **н** или нажать клавишу ENTER для выбора опции Нет*

Опции:

Да – помещает отраженное зеркально изображение в рисунок и стирает исходные объекты.

Нет – помещает отраженное зеркально изображение в рисунок и сохраняет исходные объекты.

Для управления свойствами отражения текстовых объектов используется системная переменная `MIRRTXT`. По умолчанию она включена (присвоено значение 1), и зеркальное отражение текстов осуществляется точно так же, как и для других объектов. Если для `MIRRTXT` установлено значение "отключено" (0), в этом случае отображенный текст будет ориентирован и выровнен точно так же, как и исходный.

Пример 3.4. Зеркальное отображение объекта (рис. 2.41).

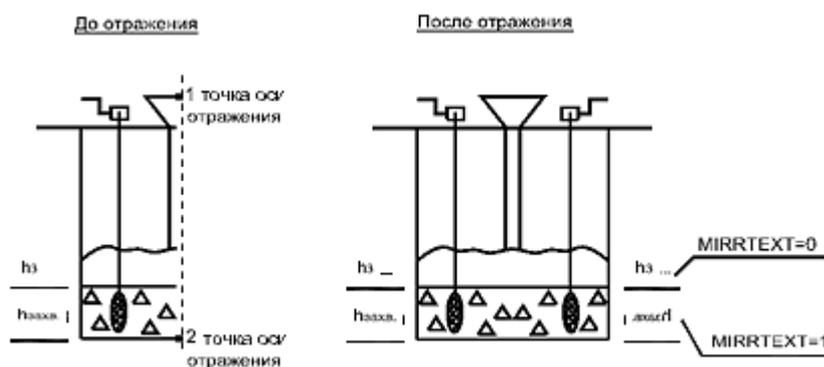


Рисунок 2.41 – Команда Зеркало

МАССИВ (ARRAY) – копирование набора объектов массивом.

Массив - получение нескольких копий выбранных объектов при размещении их в прямоугольной или круговой области. Размножение массивом является более быстрым методом для создания нескольких копий объекта с заданным интервалом, чем выполнение обычного копирования.

С помощью команды Прямоугольный массив (рис.3.7) создается прямоугольный массив путем размножения выбранных элементов с заданным количеством рядов и столбцов, образующих прямоугольную матрицу. По умолчанию строится массив 3x4.

Построение прямоугольного массива

- В окне "Массив" выбрать "Прямоугольный массив".
- В командной строке появится запрос Выбор объектов. При этом диалоговое окно временно закрывается, и AutoCAD предлагает выбрать объекты. После завершения выбора объектов нажать ENTER.
- В полях "Строки" и "Столбцы" ввести количество рядов и столбцов. Если в массиве только один ряд, необходимо задать более одного столбца, соответственно, если один столбец – задать более одного ряда.
- Задать горизонтальное и вертикальное расстояния между объектами одним из следующих способов:
 - В поле "Между" ввести значения расстояний между первым и вторым рядами. Для добавления рядов вниз необходимо задать отрицательное расстояние. В поле "Всего" указывается расстояние между первым и последним рядами.
 - В поле "Между столбцами" ввести значения расстояний между первым и вторым столбцами. Для добавления столбцов влево необходимо задать отрицательное значение. В поле "Всего" указывается расстояние между первым и последним столбцами (2.42).
- Нажать кнопку "Закреть массив".

 Прямоугольный	 Столбцы: 4	 Строки: 3	 Уровни: 1
	 Между: 1040.4312	 Между: 648.2637	 Между: 1
	 Всего: 3121.2936	 Всего: 1296.5274	 Всего: 1
Тип	Столбцы	Строки	Уровни

Рисунок 2.42 – Команда Прямоугольный массив

Построение кругового массива

С помощью команды Круговой массив (рис. 2.43) создается массив путем указания центральной точки, вокруг которой строятся копии выбранных объектов

1. На панели Редактирование выбрать Массив, указав "Круговой массив".
2. В командной строке появится запрос "Выбор объектов". Диалоговое окно временно закрывается, и AutoCAD предлагает выбрать объекты. После завершения выбора объектов нажать ENTER.
3. Далее на запрос "Укажите центральную точку массива" указать центр массива.
4. На панели Элементы указать число элементов одним из способов:
 - ввести число в поле "Количество элементов";
 - в поле "Между" указать угол между элементами;
 - в поле "Угол заполнения" указать угол между первым и последним элементами.
5. Аналогичным образом указать число строк и уровней.
6. Для поворота элементов в массиве нажать кнопку Поворот эле-

ментов массива.

– Образец динамически обновляется при изменении какого-либо параметра. Нажать кнопку "Закреть массив".

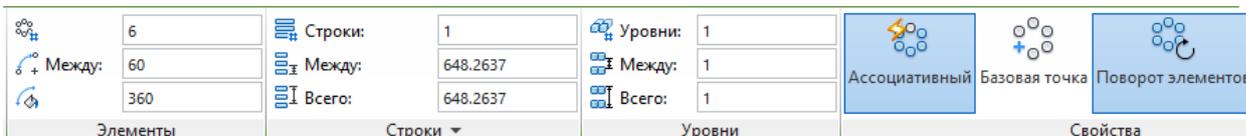


Рисунок 2.43 – Команда кругового массива

Построение массива по траектории

Равномерно распределяет экземпляры объекта вдоль траектории или ее части.

1. На панели Редактирование выбрать Массив, указав "Массив по траектории".

2. В командной строке появится запрос "Выбор объектов". Диалоговое окно временно закрывается, и AutoCAD предлагает выбрать объекты. После завершения выбора объектов нажать ENTER.

3. Следующим появляется запрос на выбор траектории.

4. Открывается панель Массива по траектории (рис. 2.44), на которой указываем число элементов одним из способов:

- поле "Число элементов". Происходит автоматический расчёт количества элементов на основе длины криволинейной траектории и расстояния между элементами;

- поле "Между". Указывается интервал между элементами;

- поле "Всего". Указывается интервал между первым и последним элементами.

5. Аналогичным образом указываются количество строк и уровней.

6. Выровнять элементы. Указывается, следует ли выравнивать каждый элемент по касательной к траектории. Выравнивание выполняется относительно ориентации первого элемента.

7. Нажать кнопку "Закреть массив".

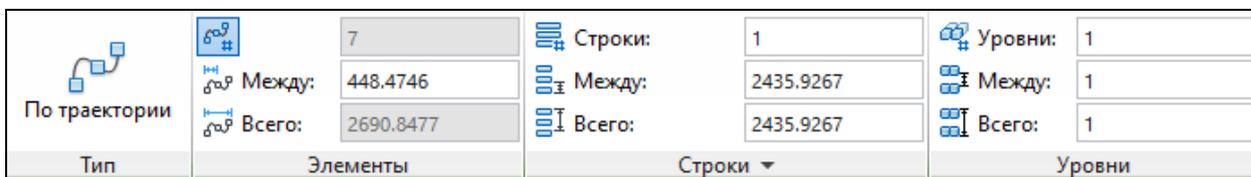


Рисунок 2.44 – Построение массива по траектории

Команды удаления



СТЕРЕТЬ(ERASE) – удаление набора объектов.

Запрос:

Команда: _erase

- Выберите объекты: *выбрать объекты любым способом и нажать ENTER по завершении выбора объектов*
Выбранные объекты удаляются из рисунка.

 **РАЗОРВАТЬ В ТОЧКЕ (BREAK)** – разбиение объекта на две части. Из одного объекта получаются два объекта без зазора между ними.

Запрос:

- Команда: `_break` Выберите объект: *указать мышкой исходный объект.*
- Первая точка разрыва: *указать точку разрыва.*

 **РАЗОРВАТЬ (BREAK)** – разрыв выбранного объекта между двумя точками.

Можно создать разрывы в большинстве геометрических тел, кроме следующих объектов: блоки, размеры, мультилинии, области.

Запрос:

Команда: `_break`

Выберите объект: *указать объект любым способом* (по умолчанию точка указания объекта считается первой точкой разрыва).

- Вторая точка разрыва или [Первая точка]: *при выборе объекта с помощью указывающего устройства программа выбирает точку, которая считается первой точкой разрыв. Укажите вторую точку разрыва или введите параметр п.*

Опции:

Первая точка – изменение координат первой точки разрыва, указанной при выделении объекта. При этом появляются новые запросы на указание первой и второй точек разрыва.

Первая точка разрыва: *указать первую точку разрыва* Вторая точка разрыва: *указать вторую точку разрыва*

Разбиение замкнутой фигуры выполняется против часовой стрелки, что должно учитываться при указании точек разбиения.

Пример 3.5. Разбить окружность в двух точках (рис. 2.45). В первом случае точки разрыва выбраны слева направо против часовой стрелки, во втором – точки выбраны справа налево по часовой стрелке.

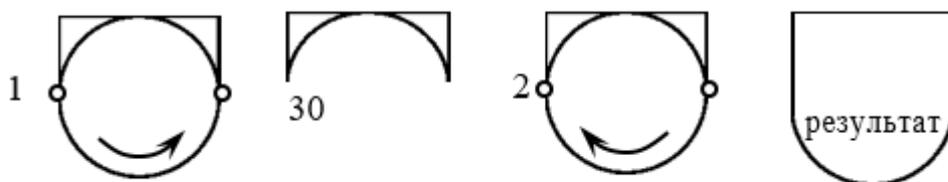


Рисунок 2.45 – Применение команды Разорвать

 **ОБРЕЗАТЬ (TRIM)** - отсечение части примитива по режущим

кромкам. Команда позволяет удалить часть объекта, выступающую за указанные границы. Для построения границы объекта можно задать одну или несколько границ отсечения. Объекты, выбираемые для определения граничных или режущих кромок, не обязательно должны пересекаться с обрезаемыми объектами (рис. 2.46).

Запросы:

Команда: `_trim`

Текущие установки: Проекция=ПСК, Кромки=Без продолжения

- Выберите режущие кромки: *выбрать один или несколько объектов, определяющие режущие кромки, и нажать ENTER, или нажать ENTER для выбора всех отображенных объектов в качестве потенциальных режущих кромок*

- Выберите обрезаемый объект (+Shift - удлиняемый) или [Линия выбора/ Проекция/Кромка/удалить/Отменить]: *выбрать объект для отсечения или опцию. По завершении выбора нажать ENTER.*

Опции:

Обрезаемый объект - задание обрезаемого объекта. Запрос для выбора объектов обрезки повторяется, что позволяет выполнять обрезку нескольких объектов. Нажать ENTER для завершения команды.

Один и тот же объект может быть одновременно и режущей кромкой, и обрезаемым объектом. Например, на рисунке окружность является режущей кромкой для прямых линий, и в то же время обрезается сама (рис.3.11).

Выбор с помощью клавиши Shift для удлинения - удлиняет выбранные объекты, а не обрезает их. С помощью этого параметра можно легко переключаться между обрезкой и удлинением.

Линия - выбор всех объектов, пересекающих линию выбора. Линия выбора - это последовательность временных сегментов линии, заданная с помощью двух или более точек линии выбора. Линия выбора не образует замкнутый контур.

Проекция - задает режим проецирования, который используется при обрезке. Без проецирования - выполняется обрезка только тех объектов, которые пересекаются с режущими кромками в 3М пространстве.

Кромка - определяет способ обрезки объекта: по растянутой кромке другого объекта или только до объекта, который пересекает его в трёхмерном пространстве.

Удалить - удаляет выбранные объекты. С помощью этого параметра удобно стирать ненужные объекты, не выходя из режима команды ОБРЕЗКА.

Отменить - отмена последнего действия команды ОБРЕЗАТЬ.

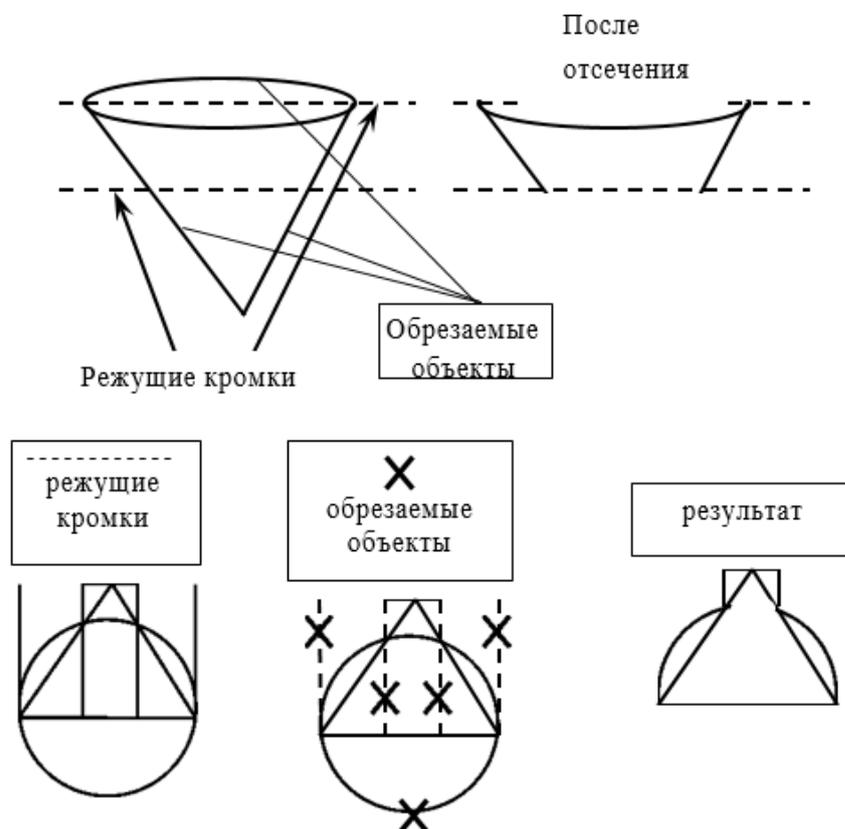


Рисунок 2.46 – Применение команды ОБРЕЗАТЬ

Команды изменения формы объектов

 **СОЕДИНИТЬ (JOIN)** - соединяет объекты для формирования одного целого объекта. Объект, к которому необходимо присоединить подобные объекты, называется исходным объектом. Объекты, которые необходимо присоединить, должны находиться в одной плоскости. Можно также создать замкнутые круги и эллипсы из дуг и эллиптических дуг. Можно соединять следующие объекты: дуги, отрезки, полилинии, сплайны.

Дополнительные ограничения для объектов:

Линия - объекты линии должны быть коллинеарными (т.е. лежать на одной бесконечной линии), но между ними должны быть зазоры

Полилинии - между объектами не может быть зазоров, и они должны лежать в одной плоскости, параллельной плоскости ХУ ПСК.

Дуги должны лежать на одной воображаемой окружности, но между ними должны быть зазоры. Параметр "Замкнуть" преобразовывает исходную дугу в окружность.

Сплайны должны быть смежными и лежать в одной плоскости.

Запрос:

Команда: `_join` Выберите исходный объект: *указать устройством указания объект, к которому необходимо присоединить объекты.*

- Выберите объекты для объединения с источником: *Выбрать один или несколько объектов для присоединения к исходному объекту*

В зависимости от того, какой выбран исходный объект, отобразятся разные подсказки в командной строке.

 **УДЛИНИТЬ (EXTEND)** - удлинение объектов до пересечения с другими объектами.

Удлинение и обрезка объектов схожи по технике выполнения. Эта команда позволяет удлинить объекты так, чтобы они заканчивались точно на граничных кромках, определенных другими объектами. Данную команду можно рассматривать как противоположную TRIM. Команда не применима к замкнутым полилиниям. При выполнении команды сначала задаются граничные линии.

Запросы: Команда: `_extend`

Текущие установки: Проекция=ПСК, Кромки=Без продолжения

Выберите граничные кромки ...

- Выберите объекты или <выбрать все>: *выбрать один или несколько объектов, определяющие граничные кромки, и нажать ENTER, или нажать ENTER для выбора всех отображенных объектов в качестве потенциальных граничных кромок*

- Выберите удлиняемый (+Shift - обрезаемый) объект или [Линия выбора/перечеркивание/Проекция/Кромка/Отменить]: *выбрать удлиняемые объекты или, удерживая SHIFT, выбрать обрезаемый объект.*

Опции команды Удлинить аналогичны опциям команды Обрезать.

 **РАСТЯЖНУТЬ (STRETCH)** - растягивание объектов. Команда позволяет изменять форму и размеры выделенной части объекта.

Объекты, которые частично находятся в текущей рамке, растягиваются. Объекты, которые полностью включены в текущую рамку, или выделены отдельно, перемещаются, а не растягиваются.

Запросы:

Команда: `_stretch`

Выберите растягиваемые объекты текущей рамкой.

- Выберите объекты: *выбрать объекты текущей рамкой или текущим многоугольником и нажать ENTER.*

- Базовая точка или [перемещение] <перемещение>: *Указать базовую точку или координаты точки перемещения*

- Вторая точка или <считать перемещением первую точку>: *Указать точку перемещения или нажать ENTER для перемещения в точку с ранее указанными координатами*

Если при выделении объекта текущей рамкой, объект целиком попадает в рамку, то команда Растянуть перемещает его в новое положение. Конечные точки объекта, находящиеся за пределами текущей рамки, не перемещаются (рис. 2.47), в то время как все другие точки выделенного объекта изменяют свое положение.

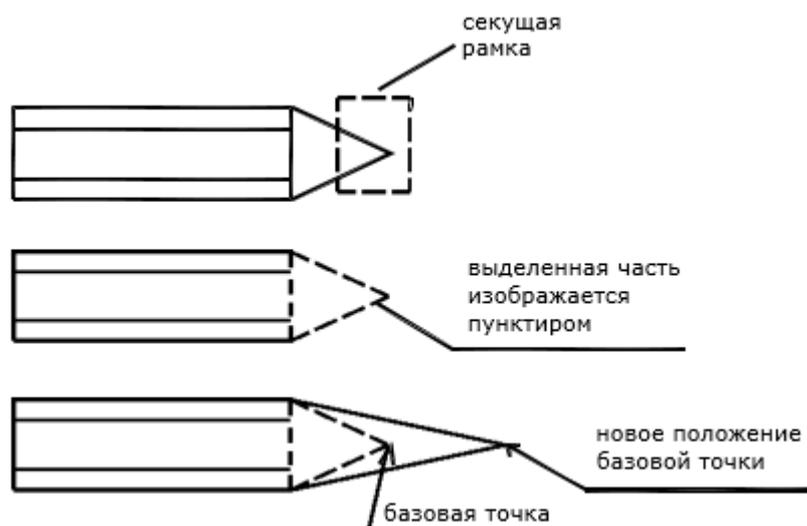


Рисунок 2.47 – Команда Растянуть

 **МАСШТАБ (SCALE)** – масштабирование объектов.

Команда применяется для пропорционального изменения размеров объектов чертежа. В зависимости от значения масштабного коэффициента объекты увеличиваются или уменьшаются. Масштабирование осуществляется относительно базовой точки, положение которой на экране остается неизменным.

Запросы:

Команда: `_scale`

- Выберите объекты: *выбрать объекты любым способом и нажать ENTER по завершении выбора*
- Базовая точка: *указать базовую точку*
- Масштаб или [Копия/Опорный отрезок] `<1.0000>`: *задать масштаб или ввести опцию*

Опции:

Масштаб – задает число, на которое будут умножены все размеры выбранных объектов. Значения, большие 1, увеличивают объекты. Значения в пределах от 0 до 1 уменьшают объекты.

Копия – создание копии выбранных объектов для масштабирования

Опорный отрезок – вычисление масштабного коэффициента. Позволяет задавать масштаб объекта относительно другой заданной длины.

Масштабирование объектов с помощью опорного отрезка

При масштабировании с помощью опорного отрезка используется имеющееся расстояние как основа для нового размера. При этом задаются длина опорного отрезка и его новая длина. Например, одна из сторон объекта имеет 4.8 единицы в длину. Необходимо таким образом масштабировать объект, чтобы она увеличилась до 7.5 единиц. Тогда первая длина является опорной, а вторая новой.

Масштабирование по опорной длине можно применять ко всему рисунку. Это может понадобиться, например, если были неверно заданы единицы рисунка. Для этого нужно выбрать в рисунке все объекты, а затем, задав опцию "Опорный отрезок", выбрать две точки и указать новое расстояние между ними. Все объекты рисунка масштабируются соответственно.

Команда: `_scale`

- Выберите объекты: *выбрать объекты* - Базовая

точка: *указать базовую точку*

- Масштаб или [Копия/Опорный отрезок] <1.0000>: *вести опцию "О"*

- Длина опорного отрезка: *вести начальное значение длины, начиная с которого изменяется масштаб выбранных объектов*

- Вторая точка: *указать вторую точку начальной длины*

- Новая длина или [Точки]: *вести конечное значение новой длины или ввести т, чтобы определить длину двумя точками.*

Пример 3.6. Дан прямоугольник и два отрезка. Масштабировать прямоугольник так, чтобы он поместился между двух отрезков (рис. 2.48).

Команда: `_scale`

- Выберите объекты: *выбрать прямоугольник и нажать ENTER*

- Базовая точка: *указать базовую точку - т.1*

- Масштаб или [Копия/Опорный отрезок]: *вести опцию "О"*

- Длина опорного отрезка: *указать первую точку начальной длины т.1;*

- Вторая точка: *указать вторую точку начальной длины т.2;*

- Новая длина: *указать вторую точку новой длины т.3.*

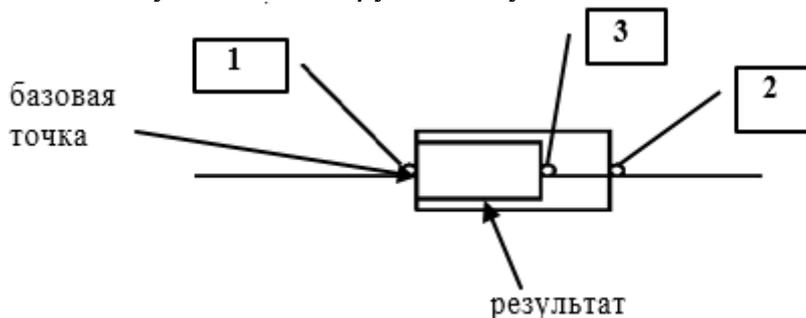


Рисунок 2.48 – Масштабирование с опорным отрезком

 **ФАСКА (CHAMFER)** - построение фасок в местах пересечения объектов. Если объекты намного короче длины фаски, то у объектов фаска не снимается (рис. 2.49).

Запросы:

Команда: `_chamfer`

(Режим С ОБРЕЗКОЙ) Параметры фаски: Длина1 = 0.0000, Длина2 = 0.0000

- Выберите первый отрезок или [Отменить/полИлиния/Длина/Угол/Обрезка/Метод/Несколько]: *выбрать первую кромку или ввести па-*

раметр

- Выберите второй отрезок или нажмите клавишу Shift при выборе, чтобы создать угол: *выбрать вторую кромку*

Опции:

Отменить - отменяет предыдущую операцию в команде. **Полилиния** - построение фасок в каждой вершине вдоль всей полилинии. Линии фасок становятся новыми сегментами полилинии.

Длина - задание длин фасок (расстояний от выбранного пересечения до концов линии фаски). Если обе длины фаски равны нулю, команда ФАСКА удлинит или обрежет отрезки так, чтобы они заканчивались в одной и той же точке.

- Первая длина фаски <0.0000>: 20 - Вторая длина фаски <20.0000>:

Угол - задание в качестве параметров фаски одной из ее длин и величины угла (рис. 3.14).

- Первая длина фаски <текущая>: *ввести значение длины фаски* - Угол фаски с первым отрезком: *ввести значение угла*.

Обрезка - определяет, обрезают ли команда ФАСКА выбранные отрезки до конечных точек фаски.

Метод - задает использование командой ФАСКА двух длин или длины и величины угла для построения фаски.

- Метод построения [Длина/Угол] <текущий>:

Несколько - создает фаски для кромок нескольких наборов объектов. Команда ФАСКА будет отображать основную подсказку, а также подсказку "Выберите второй объект" до тех пор, пока не будет нажата клавиша ENTER для завершения команды.

Построение фаски с использованием опций Длина и Полилиния

Команда: `_chamfer`

(Режим С ОБРЕЗКОЙ) Параметры фаски: Длина1=0.0, Длина2 = 0.0 - Выберите первый отрезок или [Отменить/Полилиния/Длина/

Угол/Обрезка/Метод/Несколько]: *выбрать опцию Длина - Д*

- Первая длина фаски <0.0000>: 10 - Вторая длина фаски <0.0000>: 10

- Выберите первый отрезок или [Отменить/Полилиния/Длина/Угол/Обрезка/Метод/Несколько]: *выбрать опцию Полилиния - И*

- Выберите 2М полилинию: *указать полилинию*

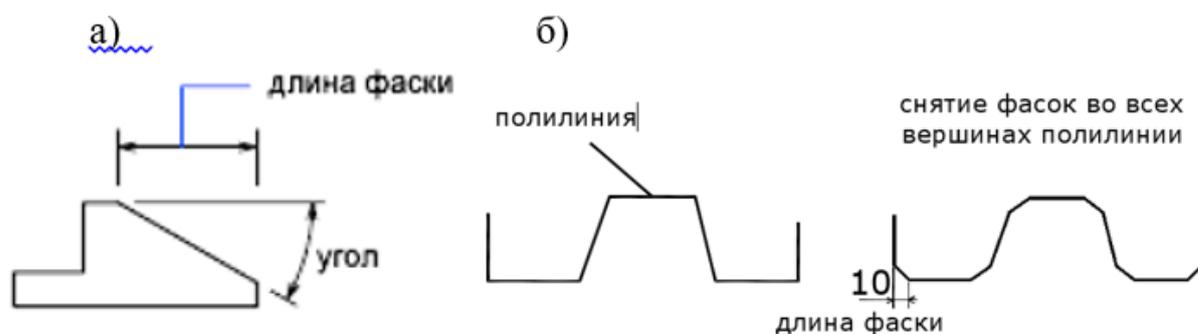


Рисунок 2.49 – Команда Фаска с использованием:
а – параметра Угол; б - параметров Длина и Полилиния

В результате все 6 сегментов полилинии были соединены фаской длиной 10 единиц.

 СОПРЯЖЕНИЕ (FILLET) – используется для скругления углов и сопряжение объектов. Сопрягает объекты дугой заданного радиуса.

Запросы:

Команда: `_fillet`

Текущие настройки: Режим = С ОБРЕЗКОЙ, Радиус сопряжения = 0.0000

- Выберите первый объект или [отменить/полилиния/радиус/ Обрезка/Несколько]: *выбрать первый объект или ввести параметр*

- Выберите второй объект: *выбрать второй объект*

Опции:

Отменить - отменяет предыдущую операцию в команде.

Полилиния - строит дуги сопряжения во всех точках пересечения линейных сегментов полилинии. Дуги становятся новыми сегментами полилинии.

Радиус – радиусом сопряжения называется радиус дуги, соединяющей сопрягаемые объекты. Изменение радиуса действует только на выполняемые после этого сопряжения, оставляя неизменными существующие. Если радиус задать равным 0, то сопрягаемые объекты просто обрезаются или удлиняются до точки пересечения без построения дуги сопряжения.

Обрезка - определяет, обрезают ли СОПРЯЖЕНИЕ выбранные кромки по конечным точкам дуги сопряжения.

- Введите режим обрезки [С обрезкой/Без обрезки] <текущий>: *ввести параметр или нажать ENTER*

Несколько - скругление углов у нескольких наборов объектов. Команда СОПРЯЖЕНИЕ будет отображать основную подсказку, а также подсказку "Выберите второй объект" до тех пор, пока не будет нажата клавиша ENTER для завершения команды.

Пример 3.7. Дан шестиугольник с длиной стороны 60. Выполнить сопряжение его вершин радиусом 30 единиц (рис. 2.50).

Команда: `_fillet`

- Выберите первый объект или [Отменить/полилиния/радиус/ Обрезка/Несколько]: *выбрать параметр радиус – Д*
- Радиус сопряжения <0.0000>: 30
- Выберите первый объект или [Отменить/полилиния/радиус/ Обрезка/Несколько]: *выбрать параметр полилиния - И*
- Выберите 2М полилинию: *выбрать шестиугольник*

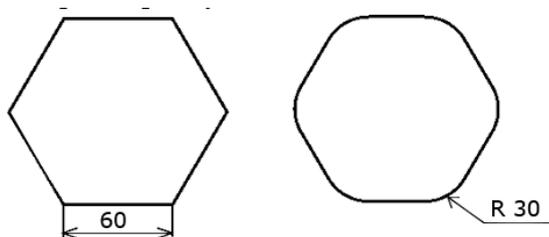


Рисунок 2.50 – Сопряжение вершин прямоугольника

 РАСЧЛЕНИТЬ - разбивка сложного объекта на составляющие его элементы.

2.2.2 Размеры и размерные стили

Для простановки размеров на чертежах используется панель Размеры вкладки Аннотации или панель Аннотация на вкладке Главная. AutoCAD предоставляет линейные, угловые, диаметральные, радиальные и другие размеры. При необходимости, стиль размера можно изменить или создать новый.

Команды нанесения размеров

 **ЛИНЕЙНЫЙ** – проставляет горизонтальные или вертикальные размеры.

Запросы:

Команда: `_dimlinear`

- Начало первой выносной линии или <выбрать объект>: *указать первую исходную точку выносной линии или выбрать объект*

- Начало второй выносной линии: *указать вторую точку выносной линии*

- Положение размерной линии или [Мтекст/Текст/Угол/ Горизонтальный/Вертикальный/Повернутый]: *указать точку в соответствии с ГОСТом или ввести опцию.*

Опции:

Мтекст - запуск редактора, используемого в месте редактирования, с помощью которого можно редактировать размерный текст.

Текст - предоставляет возможность отредактировать размерный текст в командной строке. Полученное значение размера отображается в угловых скобках. Редактирование текста в скобках (<>) приводит к изменению или удалению размерной величины, вычисленной программой. Для добавления текстовых элементов до и после размерного числа следует ввести текст до

и после угловых скобок соответственно.

Угол - изменяет угол поворота размерного текста.

Выбор объекта - автоматическое определение начальных точек первой и второй выносной линии после выбора объекта.

 **ПАРАЛЛЕЛЬНЫЙ** – проставляет линейный размер параллельно выбранному отрезку или двум указанным точкам.

Команда работает аналогично предыдущей.

 **РАДИУС** – используется для измерения радиуса выбранной дуги или окружности и отображения размерного текста, перед которым стоит символ радиуса.

Запрос:

Команда: `_dimradius`

- Выберите дугу или круг: *задать точку на дуге или окружности*
- Положение размерной линии или [Мтекст/Текст/Угол]: *задать точку, которая определяет положение выносной линии или ввести опцию*

Опции аналогичные опциям команды **Линейный**.

 **ДИАМЕТР** – проставляет размер диаметра окружности.

Команда: `_dimdiameter`

Выберите дугу или круг:

Положение размерной линии или [Мтекст/Текст/Угол]: Команда работает аналогично предыдущей.

 **УГЛОВОЙ** – проставляет угол между отрезками или угловой размер дуги.

Запрос:

Команда: `_dimangular`

- Выберите дугу, круг, отрезок или <указать вершину>: *выбрать дугу, круг или отрезок либо нажать ENTER для нанесения углового размера по трем точкам*

- Вторая конечная точка угла: *указать вторую точку сектора дуги, окружности или второй отрезок*

- Положение размерной дуги или [Мтекст/Текст/Угол]: *указывается положение размерной линии или вводится опция*

Опции аналогичны опциям предыдущих команд.

 **БАЗОВЫЙ** - проставляет размеры от базовой линии.

Размеры от общей базы — это последовательность размеров, отложенных от одной базовой линии.

 **ПРОДОЛЖИТЬ** – проставляет цепочку размеров

У размерных цепей начало каждого размера совпадает с концом предыдущего, т.е. каждый последующий размер имеет общую выносную линию с предыдущим.

Для нанесения размеров от общей базы или размерной цепи для изделия должен быть проставлен хотя бы один линейный, ординатный или угловой размер. Если не указать начальную размерную точку, то базовая линия или первая выносная линия размерной цепи располагаются на месте второй выносной линии последнего нанесенного размера (рис. 2.51).

Запросы:

- Выберите исходный размер: *выбрать линейный, ординатный или угловой размер*

Если размеры уже нанесены, программа не выводит этот запрос и использует последний созданный в текущем сеансе объект для нанесения размера.

Если выбран линейный или угловой размер, появляется следующий запрос:

- Указать исходную точку второй выносной линии или [Отменить/Выбрать] <Выбрать>: *указать точку, задать опцию или нажать клавишу ENTER для выбора исходного размера*

При необходимости, повторить указание начальных точек следующих выносных линий.

- Нажать дважды ENTER для завершения команды.

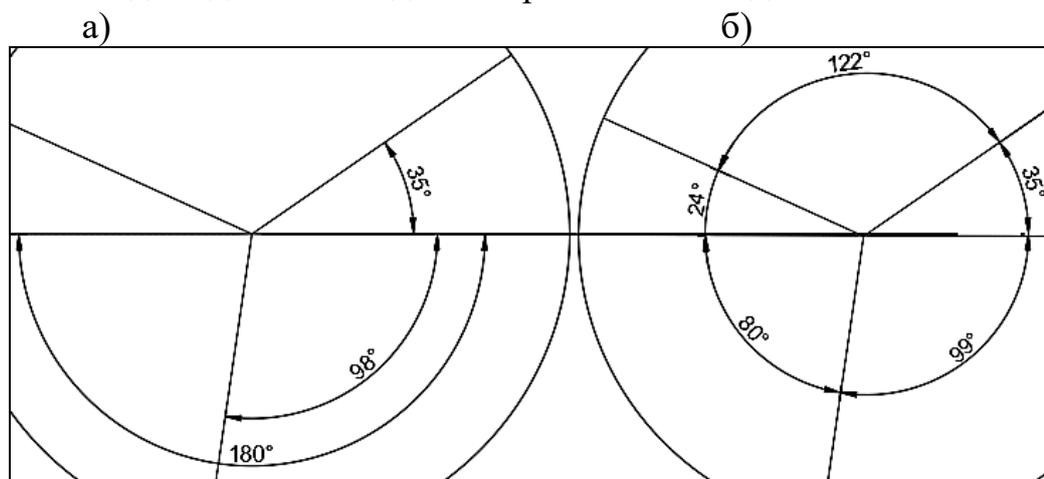


Рисунок 2.51 – Простановка размеров:

а – базовый; б – продолжить



ВЫНОСКА - используется для быстрого нанесения выносок и пояснений на выносках.

Выноска представляет собой прямую линию со стрелкой на конце, связанную с каким-либо элементом рисунка. Выноска находится на вкладке Главная – Аннотации ил на вкладке Аннотация - Выноски (рис. 2.52). Обычно выноска используется для указания, к какому объекту относится пояснительная надпись.

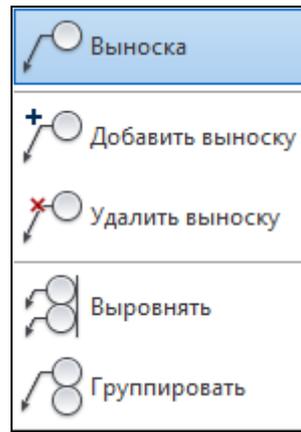


Рисунок 2.52 – Команда Мультивыноска

Запросы:

Команда: `_leader`

- Первая точка выноски или [Параметры] <Параметры>: ***m.1*** или Enter

- Следующая точка: ***m.2***

- Следующая точка: ***m.3***

Ширина текста <0>: Enter

Первая строка текста пояснения: *ввод текста* Следующая строка текста надписи: Enter

Опции:

Параметры – настраивание формата выносной стрелки и расположение текста выноски.

Открывается окно Параметры выноски, где можно изменить тип пояснения, формат стрелки и выровнять текст ссылки.

После закрытия окна Параметры продолжает выполняться команда Выноска, в которой указываются опорные точки и текст для выноски.

Текст и выносные линии не объединены в один объект. Редактировать их можно отдельно.

Стиль мультивыноски

Создать или редактировать мультивыноску можно при помощи команды стиль мультивыноски, находящейся на вкладке Главная – Аннотации – Стиль мультивыноски либо Аннотации – Выноски. В диспетчере мультивыносок для создания нового стиля необходимо нажать кнопку Создать и ввести новое имя для стиля. Откроется окно с тремя вкладками.

Первая вкладка Формат выноски (рис. 2.53а) позволяет выбрать тип выноски и стрелки, а также установить размер стрелки.

Вторая вкладка Структура выноски (рис. 2.53б) позволяет задать количество точек выноски и величину полки. В поле Масштаб находится переключатель Задать масштаб, который соответствует глобальному масштабу в размерном стиле.

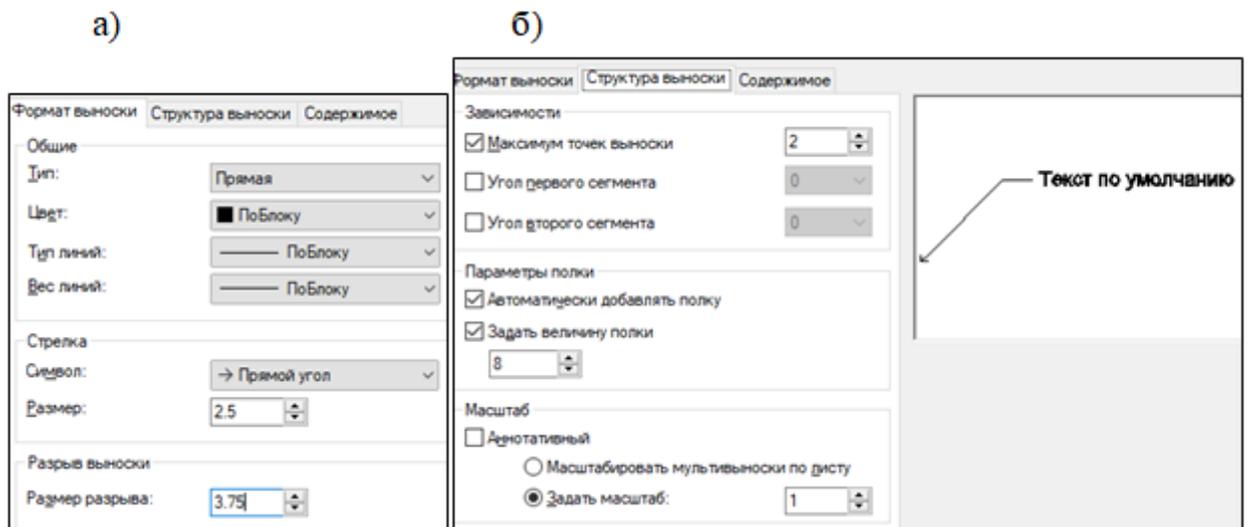


Рисунок 2.53 – Изменение стиля мультивыноски:

а – формат выноски; б – структура выноски

На третьей вкладке Содержимое (рис. 2.54) можно установить высоту текста и присоединение выноски по горизонтали и вертикали.

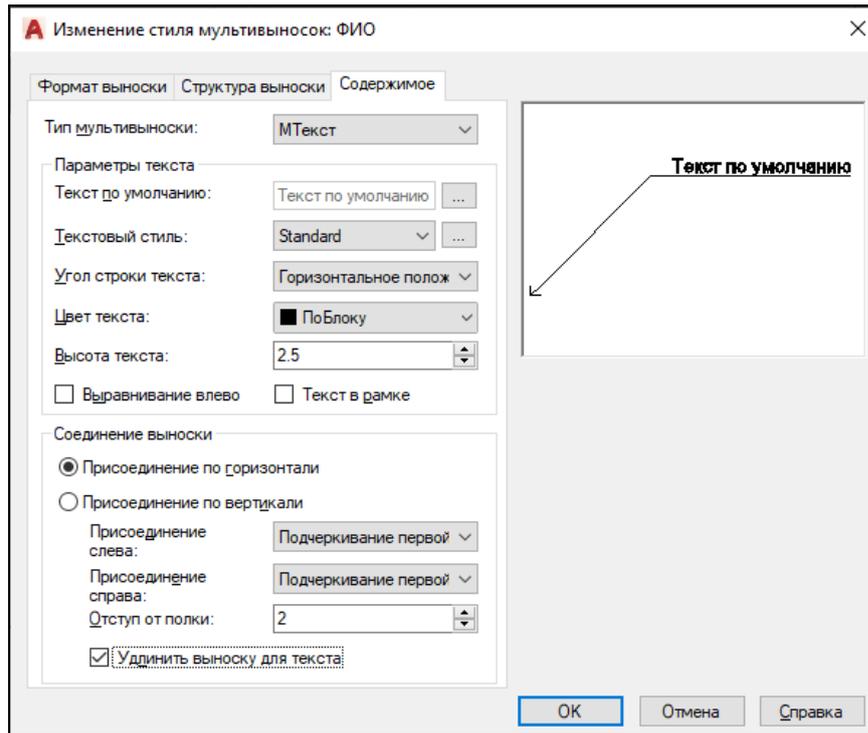


Рисунок 2.54 – Вкладка Содержимое стиля мультивыноски

Размерные стили

Для облегчения управления размерными переменными в AutoCAD предусмотрено понятие размерных стилей. Размерным стилем называется именованная группа установок размерных переменных, определяющая внешний вид проставляемого размера.

По умолчанию установлен размерный стиль ISO-25, но можно создавать собственные размерные стили.

Работа с размерными стилями выполняется с помощью команды  **Размерные стили**, которой соответствуют кнопка панелей **Стили** и **Размеры**, а также пункт **Размерные стили** падающего меню **Формат**.

Команда **Размерные стили** открывает диалоговое окно **Диспетчер размерных стилей** (рис. 2.55).



Рисунок 2.55 – Диспетчер размерных стилей

Вкладка Линии (рис. 2.56) содержит параметры, изменяющие цвет, тип и вес размерных линий. У выносных линий устанавливается отступ от объекта и величина удлинения за размерные линии.

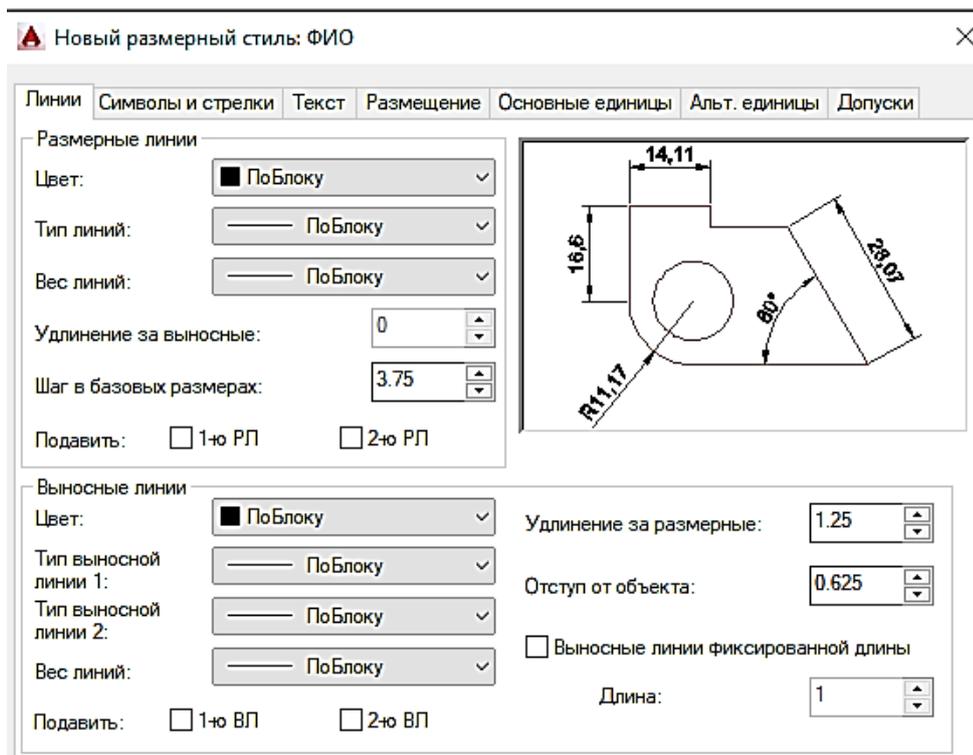


Рисунок 2.56 – Вкладка Линии

Вкладка Символы и стрелки позволяет изменять тип и размер стрелок, выносок (рис. 2.57).

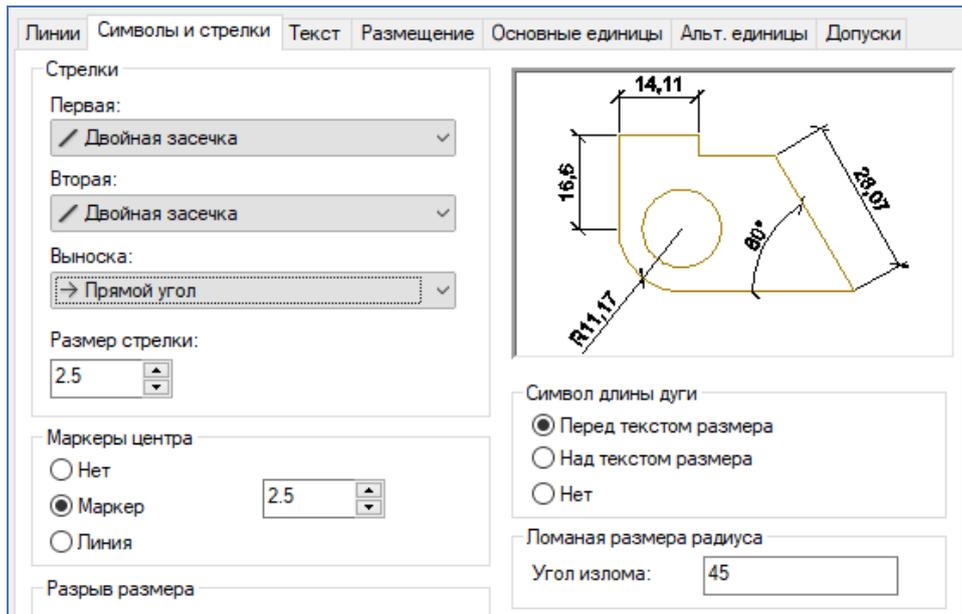


Рисунок 2.57 - Вкладка Символы и стрелки

Вкладка Текст (рис.2.58) позволяет изменить высоту текста, отступ от размерной линии. Устанавливаем ориентацию текста вдоль размерной линии. Текст можно размещать по вертикали по центру, над линией, снаружи, под линией. По горизонтали текст рекомендуется установить по центру.

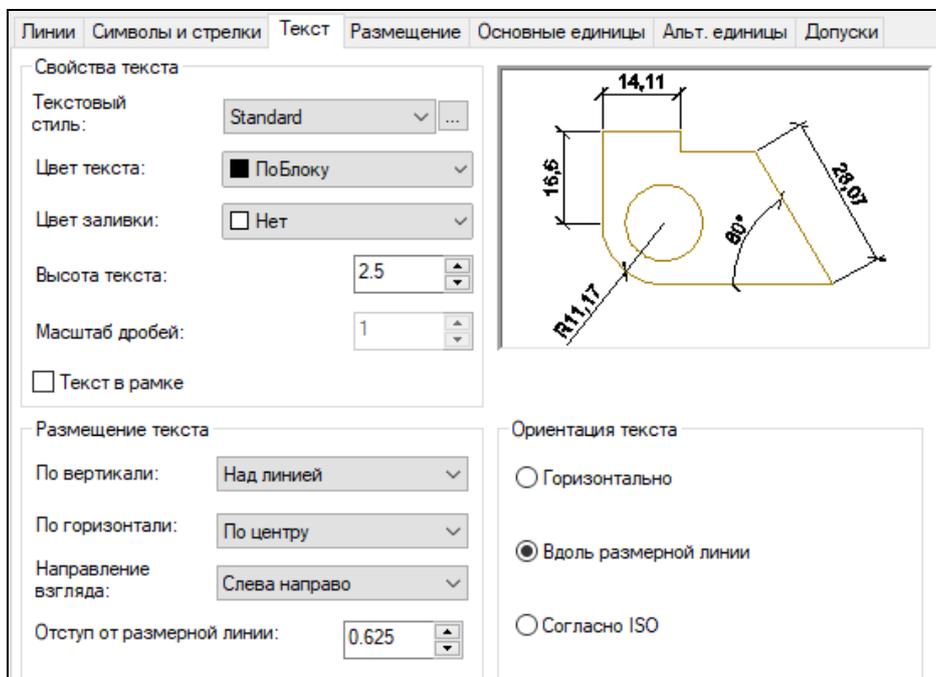


Рисунок 2.58 – Вкладка Текст

На вкладке Размещение (рис.2.59) находится Глобальный масштаб, влияющий на все элементы оформления размеров. Увеличивает, если больше 1, и уменьшает, если масштаб меньше 1, высоту (начертание) раз-

меров, но не изменяет их значения. Размерный текст и засечки изменяются пропорционально.

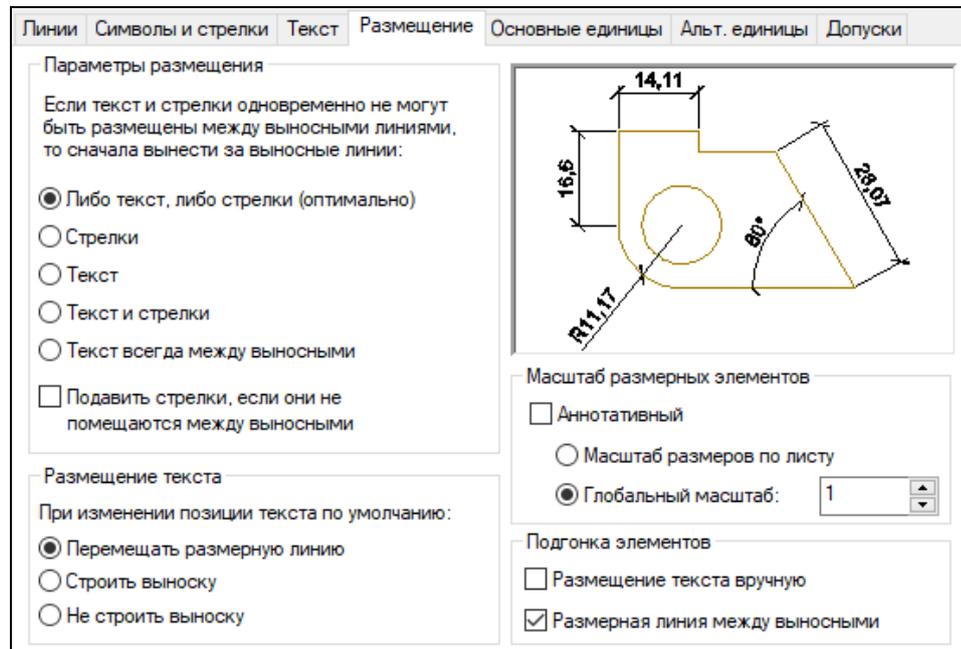


Рисунок 2.59 – Вкладка Размещение

На вкладке Основные единицы (рис. 2.60) можно установить формат и точность единиц измерений, подавить хвостовые нули.

Область Масштаб измерений содержит масштабный множитель **Масштаб**, на который умножаются все линейные размеры. Масштаб измерений, отличный от единицы, изменяет значения размеров.

Изменять положение размера можно с помощью ручек, увеличивая или уменьшая расстояние до объекта.

Передвигая в размере среднюю ручку в сторону, можно изменять положение размерного текста.

Изменить размеры можно и в контекстном меню. Щелкнув правой кнопкой мыши по размеру, вызываем контекстное меню, в котором можно указать:

- новое положение размерного текста;
- точность;
- размерный стиль.

Размеры также можно изменять на палитре Свойства.

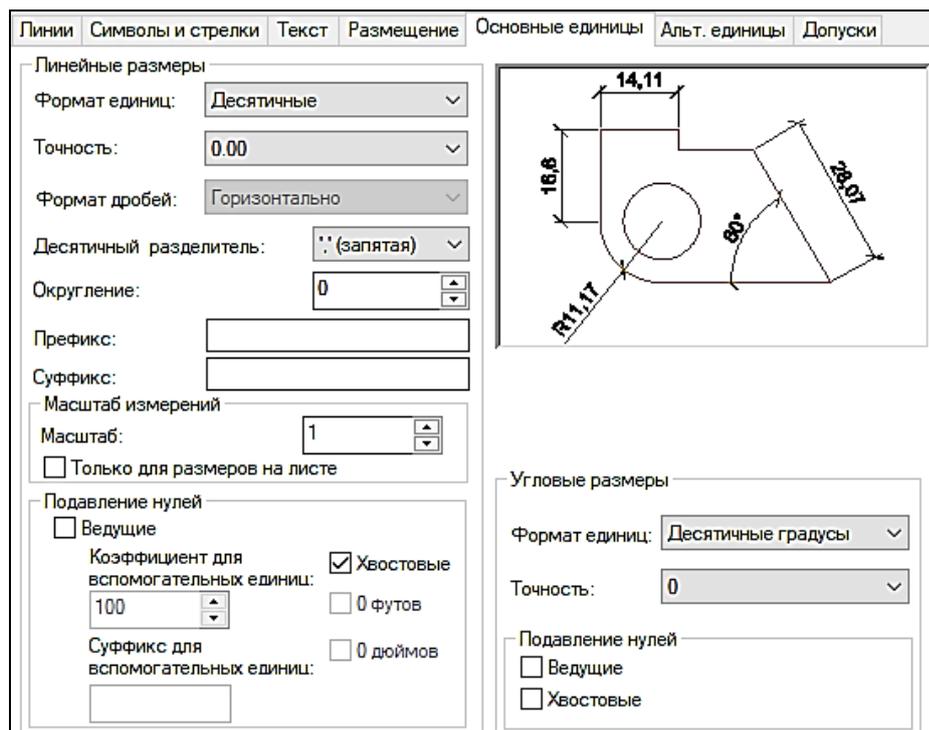


Рисунок 2.60 – Вкладка Основные единицы

2.2.3 Работа с текстом

Для нанесения надписей в чертеже используется однострочный и многострочный текст. Короткие фрагменты текста выполняются однострочным текстом, а длинные надписи с форматированием выполняются при помощи многострочного текста. Каждая текстовая надпись выполняется текущим текстовым стилем, который определяет различные параметры текста.

Создание и редактирование стиля текста

Совокупность параметров, определяющих внешний вид текста, называется стилем текста. Для создания стиля текста выполняется следующая последовательность действий:

1. Выбрать команду **Стиль текста** из меню **Формат** или активизировать кнопку **Текстовые стили**.
2. На экране появится одноименное диалоговое окно, в котором находятся все опции, применяемые для создания и редактирования стилей текста (рис. 2.61).
3. Нажать кнопку **Новый** и в открывшемся окне **Новый текстовый стиль** присвоить создаваемому стилю имя. Во избежание неудобства работы, рекомендуется назначать имена, несущие некоторую информацию о стиле текста.

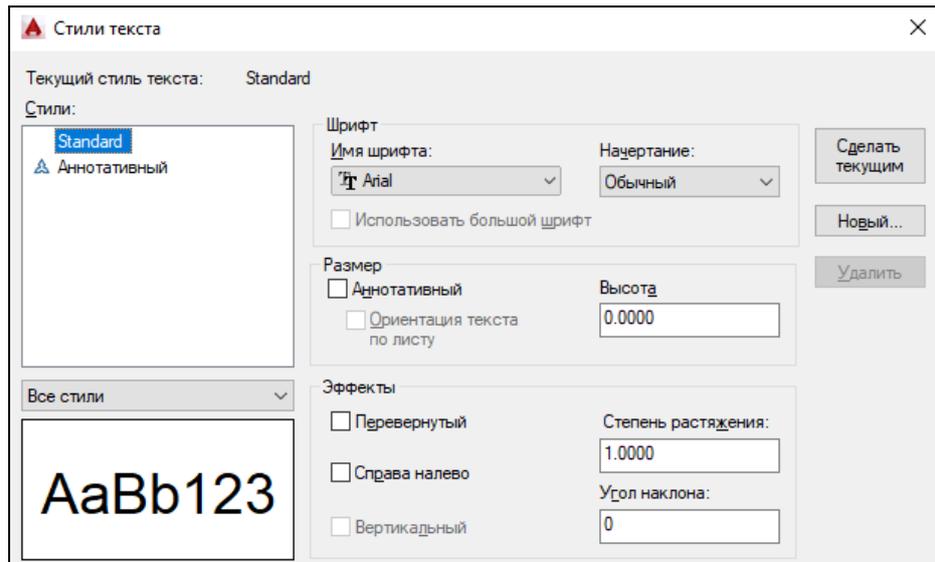


Рисунок 2.61 – Создание текстового стиля

4. В разделе Шрифт из списка необходимо выбрать указанный шрифт, способ его начертания и высоту. Если задать нулевую высоту, то при создании текста командная строка каждый раз будет запрашивать значение высоты. Имя шрифта содержит как шрифты true type, так и шрифты созданные специально для Autocad с расширением *.shx. Наиболее полно соответствует требованиям ЕСКД файл шрифта amk.shx. При его отсутствии рекомендуется при выполнении чертежей использовать шрифт simplex.shx или txt.shx. Для Shx-шрифтов полужирное и курсивное начертания не поддерживаются.

5. В разделе Эффекты можно выбрать варианты представления текста: перевернутый, справа налево, вертикальный. Здесь можно изменить угол наклона и степень растяжения символов.

6. В области Просмотр можно увидеть, как символы будут выглядеть на чертеже. Это удобный способ проверить, поддерживает ли выбранный шрифт кириллицу.

7. Для сохранения выбранных свойств текстового стиля необходимо нажать кнопку Применить.

Создание однострочного текста

Однострочный текст в Autocad создается с помощью команд dtext, которую можно ввести непосредственно в командную строку, или в падающем меню выбрать команду Рисование – Текст – Однострочный. После выполнения любого из этих действий в командной строке появится следующее сообщение:

Команда: `_dtext`

Текущий текстовый стиль: "стили1" высота текста: 2.5000

- Начальная точка текста или [Выравнивание/Стиль]: *вести начальную точку текста или выбрать одну из опций: Выравнивание*

строк текста или выбор Стиля текста

- Высота <2.5000>: *указать высоту текста*

- Угол поворота текста <0>: *указать угол поворота* - Введите текст:
ввести текст

Расстояние между строками устанавливается автоматически.

Опции:

Выравнивание - позволяет производить выравнивание строк текста.

Опции выравнивания обеспечивают следующие возможности: Задайте опцию [вПисный/Поширине/Центр/сЕредина/вПраво /ВЛ/ВЦ/ВП/СЛ/СЦ/СП/НЛ/НЦ/НП].

Вписанный – выравнивает строку текста между двумя заданными точками. При этом для соблюдения пропорций автоматически корректируется высота текста.

По ширине – выравнивает строку текста между двумя заданными точками. Высота текста не изменяется.

Середина – располагает центр базовой линии первой строки текста в заданной точке.

Центр – выравнивает по центру заданную строку текста по вертикали и по горизонтали относительно заданной точки. Последующие строки текста центрируются относительно первой.

Вправо – создает текст, выровненный по правому краю от заданной точки.

Опции ВЛ/ВЦ/ВП/СЛ/СЦ/СП/НЛ/НЦ/НП - выполняют выравнивание текстовых строк относительно указанных точек.

Стиль - выбор стилей текста. Для этого на запрос о вводе начальной точки текста нужно выбрать опцию Стиль и ввести имя стиля, созданного в настоящем чертеже. Если вместо имени стиля ввести знак вопроса, то можно просмотреть список всех существующих в чертеже стилей.

Создание многострочного текста

Большие блоки текста (многострочный текст) удобно создавать командой МТЕКСТ. При этом создается прямоугольная область, в которой и располагается весь текст. Объект может содержать как одну строку, так и один или несколько абзацев. Весь текст независимо от его объема, воспринимается как единый объект. Команда МТЕКСТ имеет больше опций, чем другие текстовые команды.

Вызвать команду создания многострочного текста можно несколькими способами:

- 1) ввести `_mtext` непосредственно в командную строку;
- 2) в меню Рисование выбрать пункт Текст и активизировать команду Многострочный текст;
- 3) использовать кнопку на панели инструментов.

Запросы:

Команда: `_mtext` текущий текстовый стиль: "standard", высота текста: 2.5

- Первый угол: *указать расположение первого угла области, в которой должен располагаться текст*

- Противоположный угол или [высота/выравнивание/межстрочный интервал/поворот/стиль/ширина]: *ввести второй угол ограничивающей текстовой рамки, расположенный по диагонали относительно первого или выбрать одну из предлагаемых опций. Эти опции дублируются в окне редактора.*

Отобразится текстовый редактор, используемый в месте редактирования (рис. 2.62). Для задания отступа первой строки абзаца на горизонтальной линейке переместить маркер "Первая строка" в положение, с которого должен начинаться текст. Для задания отступа остальных строк абзаца на горизонтальной линейке переместить маркер "Абзац".

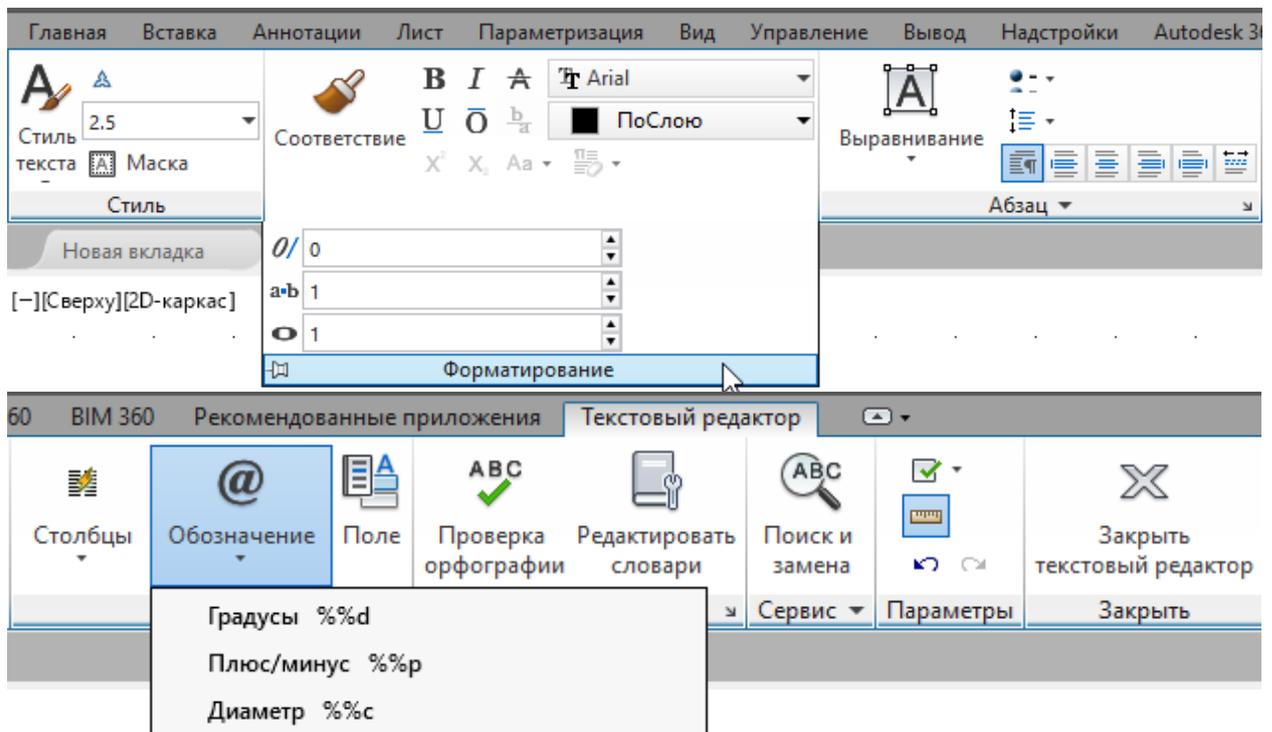


Рисунок 2.62 – Многострочный текст

Для задания позиций табуляции щелкнуть в нужных местах на горизонтальной линейке.

Для задания текстового стиля, отличного от установленного по умолчанию, раскрыть список "Стиль" на панели Редактора и выбрать требуемый стиль.

– Ввести текст, который можно редактировать и выравнивать, предварительно выделив его.

- С помощью панели инструментов изменить формат следующим образом:
 - Для изменения шрифта выбрать нужное значение из списка "Шрифт".
 - Для изменения высоты символов ввести новое значение в поле "Высота текста".
 - Для изменения начертания символов на полужирное или курсивное (только для шрифтов TrueType) или создания подчеркнутого или надчеркнутого текста щелкните соответствующую кнопку на панели инструментов.
 - Для изменения цвета символов выбрать нужное значение из списка "Цвет". Если необходимый цвет отсутствует в списке, следует воспользоваться пунктом "Выбор цвета".
 - Для вставки символов использовать кнопку Обозначение (@).
 - Поле трекинг ($a \rightarrow b$) отвечает за межзнаковый интервал.
 - Для сохранения текста и выхода из редактора воспользуйтесь одним из следующих приемов:
 - Нажать кнопку Закрывать текстовый редактор.
 - Щелкнуть мышью в области рисования за пределами Редактора.
 - Использовать сочетание клавиш CTRL + ВВОД.

Создание текстовых полей

Поле представляет собой текст, содержащий правила отображения данных, которые могут поменяться в ходе работы над чертежом.

При обновлении поля отображаются самые новые данные. Например, значением поля «Имя Файла» является имя файла. Если имя файла изменяется, то при обновлении данного поля отобразится новое имя файла.

Для вставки текстового поля необходимо зайти в текстовый редактор, поместить курсор в то место, где должен располагаться текст поля, и нажать правую кнопку мыши. В контекстном меню выбрать "Вставить поле" (рис.2.63). Или для вызова этой команды использовать сочетание клавиш Ctrl+F. В диалоговом окне "Поле" в группе "Категории полей" выбрать "Все" или категорию поля. Поля в выбранной категории будут отображены в списке "Имена полей".

Поле, для которого нет значений, будет отображать черточки (---). Например, поле "Автор", устанавливаемое в диалоговом окне "Свойства чертежа", может быть оставлено пустым. Неверное поле будет отображать знаки решеток (####). Например, поле "НазваниеТекущегоЛиста", допустимое только для пространства листа, будет отображать знаки решетки, если его расположить в пространстве модели.

Поле является частью тестового объекта и не может редактироваться в текстовом редакторе. Самый простой способ редактирования поля состоит в

следующем. Нужно дважды нажать кнопку мыши на текстовом объекте, содержащем поле, затем, для отображения диалогового окна "Поле", нужно дважды нажать кнопку мыши на поле. Эти операции можно выполнять также в контекстных меню.

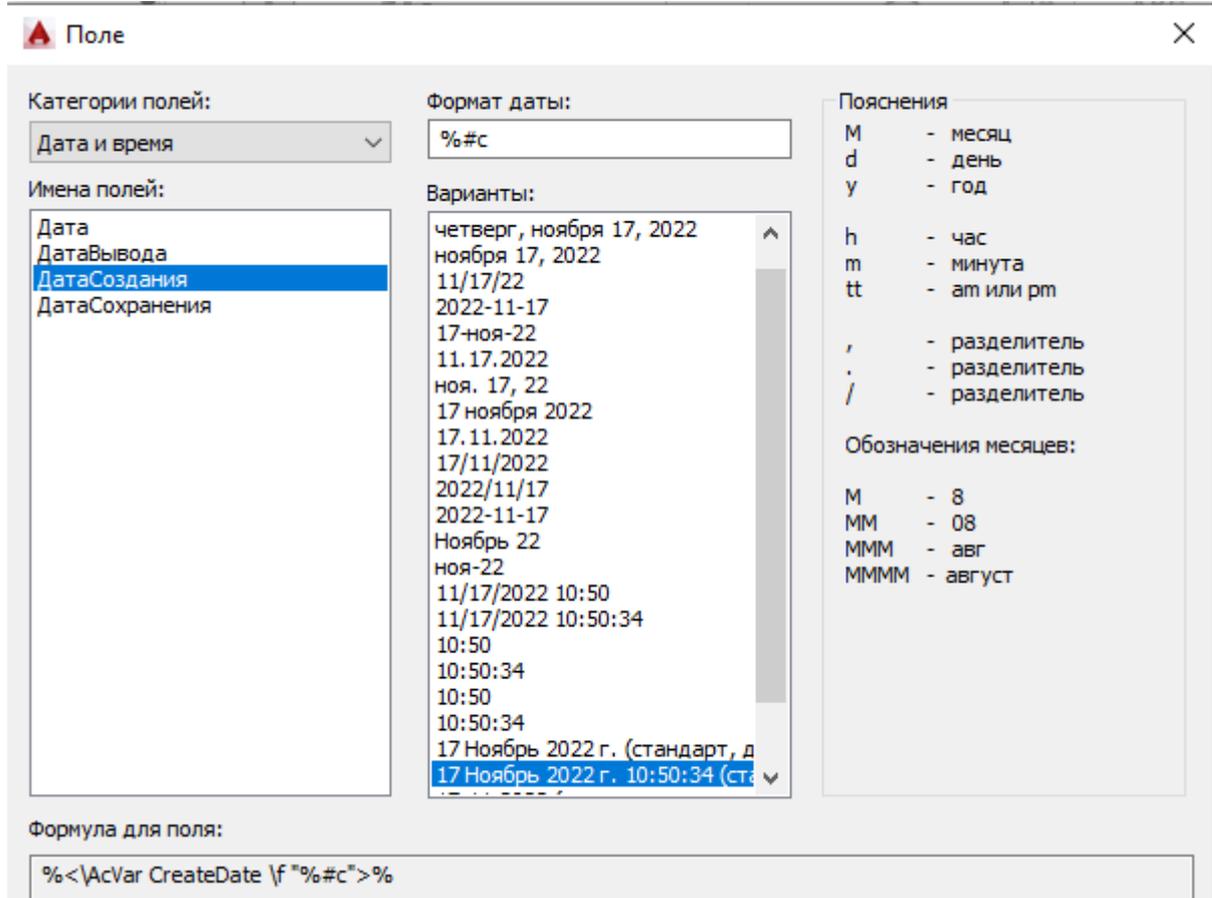


Рисунок 2.63 – Вставка текстового поля

2.2.4 Штриховка и градиент

Часто на чертежах требуется заполнить некоторую область в соответствии с определенным шаблоном. Заполнение области, называемое *штриховкой*, используется для обозначения материала, из которого должен быть изготовлен проектируемый объект, например металл, бетон и т.д.

Штриховка – это отдельный объект, заполняющий пространство, она должна пояснять назначение других объектов, а не доминировать на чертеже.

Штриховка вызывается с помощью кнопки  панели *Рисование* на вкладке *Главная*.

После вызова команды в командной строке появляется запрос:

Команда: `_hatch`

Укажите внутреннюю точку или [Выбрать объекты/Отменить/ Параметры]: Выбор всех объектов...

Выбираются все видимые объекты...

После выбора объектов открывается вкладка Создание штриховки (рис.2.64).

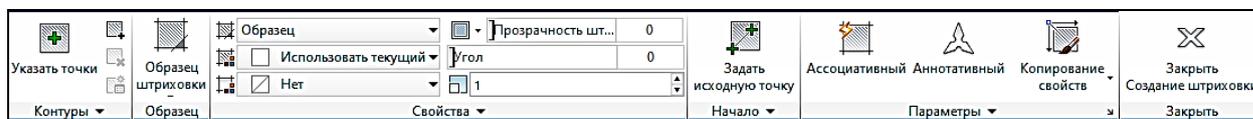


Рисунок 2.64 – Вкладка Создание штриховки

Рассмотрим назначение панелей для создания штриховки:

1. **Контур.** Команда позволяет создать штриховку области, ограниченной замкнутой линией (линиями) либо путем простого указания точки внутри замкнутого контура, или позволяет указать объекты, образующие границу области штриховки, нажав на кнопку **Выбрать объекты контура**.

2. **Образец.** Кнопка **Образец штриховки** открывает список различных типов штриховки.

3. **Свойства.** Здесь можно установить тип и цвет штриховки, а также цвет фона.

В списке **Образец** можно выбрать **Градиент**, и заливка области будет осуществляться не линиями, а несколькими выбранными цветами. Градиентная заливка может состоять из оттенков одного цвета или представлять собой плавный переход из одного цвета в другой. Градиентные заливки можно использовать для украшения рисунков, создавая эффект отражающегося от объекта света.

В раскрывающемся списке **Угол** задается угол наклона штриховых линий шаблона относительно оси X. В раскрывающемся списке **Масштаб** изменяется плотность нанесения линий для выбранного шаблона штриховки.

4. **Начало.** Смещение образца штриховки в соответствии с указанной исходной точкой. Иногда может потребоваться перенести начальную точку штриховки (исходная точка). Например, при создании кирпичного образца, возможно, потребуется начать с целого кирпича в левом нижнем углу штрихуемой области. В этом случае используется параметр **Задать исходную точку**.

5. **Параметры.** Ассоциативная – штриховка может следовать за изменением формы и размеров контуров. Неассоциативная – штриховка не зависит от конфигурации контура.

Кнопка **Копирование свойств**. Команда наносит указанную этой кнопкой штриховку на новый объект.

На этой панели можно открыть диалоговое окно **Штриховка и градиент** (рис. 2.65).

Поле *Острова* используется в том случае, когда зона штриховки состоит из вложенных друг в друга объектов, и важно определить способ об-

наружения внутренних контуров. Режим обработки островков (замкнутых объектов внутри контура штриховки) содержит три стиля:

- нормальный – штриховка наносится через контур;
- внешний – штрихуется только внешний контур;
- без островков – штрихуются все контуры подряд.

По умолчанию используется обычный режим.

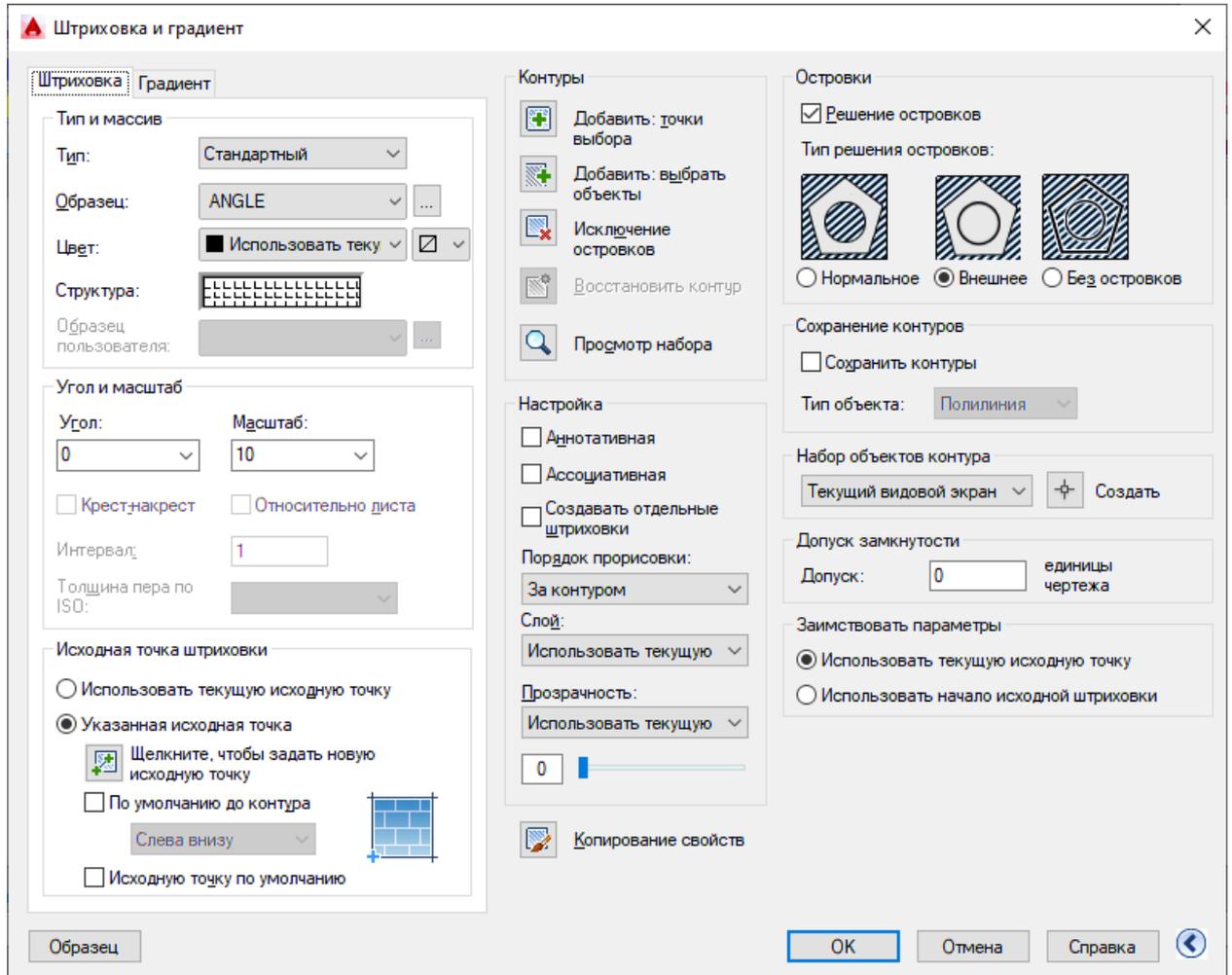


Рисунок 2.65 – Установка параметров штриховки

6. Закрыть. Закрыть создание штриховки.

AutoCAD не наносит штриховку на текстовые объекты, атрибуты, формы и закрашенные фигуры. Это позволяет, например, сохранить читаемость текста внутри заштрихованных секторов круговых диаграмм.

Штриховка создается как единый примитив, поэтому при выборе штриховки как объекта высвечивается только одна ручка.

2.2.5 Получение твердой копии чертежа

Команда Файл – Диспетчер плоттеров позволяет вызвать Мастера установки плоттеров (рис. 2.66). В этом окне выбирается тип устройства пе-

чати – локальный, сетевой или системный, модель плоттера, порт подключения, драйвер и имя плоттера.

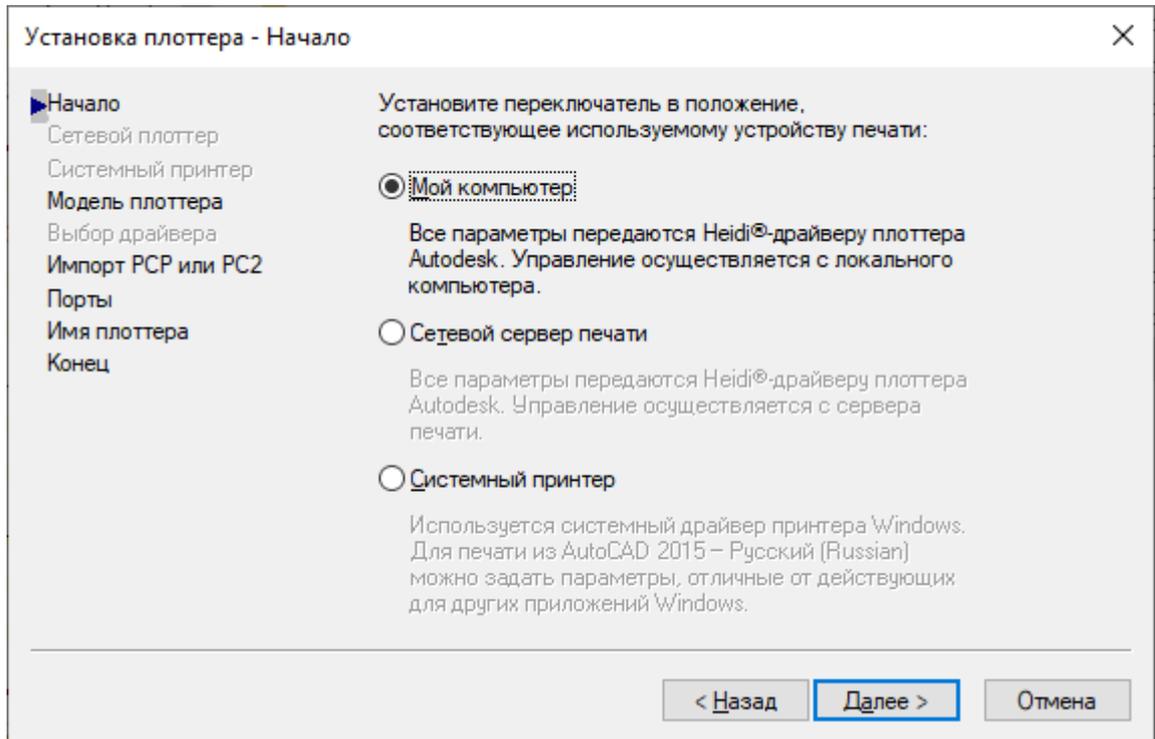


Рисунок 2.66 – Мастер установки плоттера

Таблица стилей печати – это необязательный элемент слоя, предназначенный для определения совокупности данных о цвете, толщине, контрастности и т.д. и позволяющий видоизменить объекты чертежа при выводе его на печать.

Стиль печати определяет цвет объектов, тип и толщину линий. Таким образом, изменение стиля печати позволяет получить при выводе различные виды чертежа. Параметры стиля печати определяются в таблице стиля. Для различных видов при выводе можно определить различные таблицы стиля печати.

Стили бывают цветозависимые и именованные. Цветозависимые стили основываются на цвете самих объектов, т.е. на том цвете, с которым они видны на экране. Именованные стили не зависят от цвета объектов и могут быть присвоены независимо от их цвета.

Чтобы создать новую таблицу стиля печати (рис. 2.67), необходимо выполнить следующие действия:

- Вызвать мастер создания стилей печати (Файл – диспетчер стилей печати).
- На первой странице Начало выбрать переключатель Простейший шаблон и нажать кнопку Далее.

- В следующем окне Указать таблицу стиля печати выбрать опцию Именованная таблица стилей печати для создания именованного стиля.
- В окне Имя файла указать имя нового стиля и нажать кнопку Далее. Будет создана новая таблица стилей печати с параметрами, установленными по умолчанию.
- В заключительном окне Готово нужно нажать кнопку Редактор таблиц стилей печати и перейти к окну редактирования установленных по умолчанию параметров.

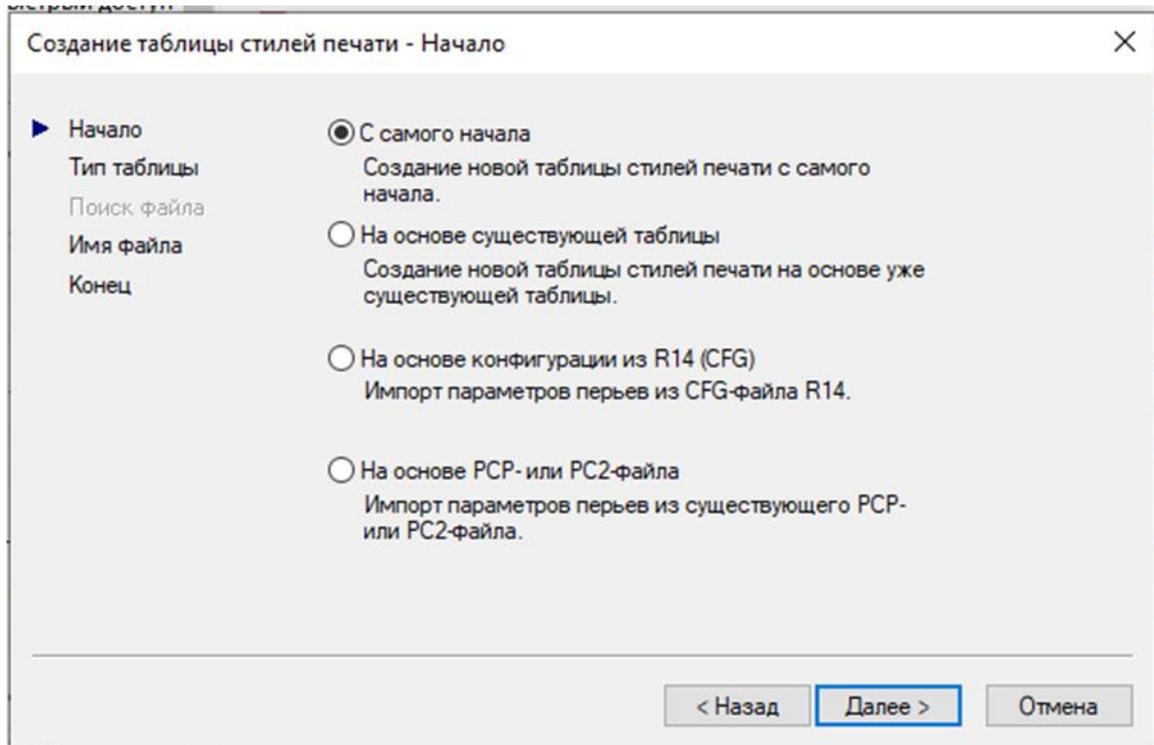


Рисунок 2.67 - Создание стиля печати

Чтобы распечатать чертеж с нужными параметрами, необходимо воспользоваться опцией "Диспетчер параметров листов" (рис. 2.68), которая находится в меню Файл. В открывшемся окне предлагается ввести имя набора параметров листа.

Затем предлагается выбрать для заполнения следующие параметры:

1. Принтер/плоттер. Содержит сведения о сконфигурированном устройстве печати для листов:

- Имя. Список доступных PC3 файлов или системных принтеров, выбираемых для вывода на печать текущего листа. Устройство идентифицируется в качестве файла PC3 или системного принтера по значку, стоящему перед его именем.

- Плоттер. Устройство печати для текущего набора параметров листа.

- Подключение. Физическое размещение выбранного устройства печати для текущего набора параметров листа.

– Описание. Текстовое описание к устройству печати для текущего набора параметров листа. Текст можно редактировать в Редакторе параметров плоттера.

– Печать в файл (а не на плоттер или принтер). При нажатии на ОК в диалоговом окне "Печать" открывается диалоговое окно "Поиск файла чертежа" (стандартное диалоговое окно поиска файлов).

– Частичный образец. Точное отображение действительной области печати, соотносимое с форматом бумаги и печатаемой областью. На образце указан формат листа и обозначена печатаемая область.

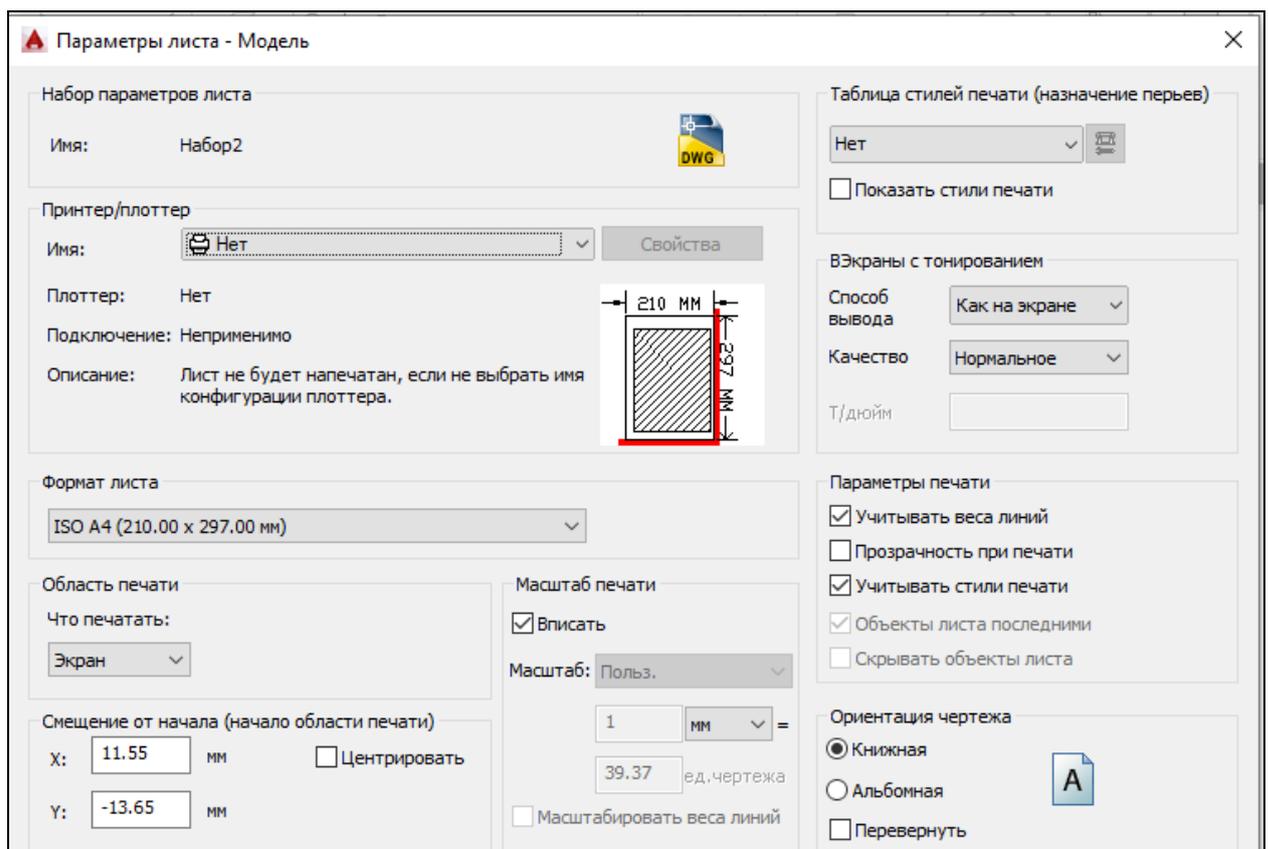


Рисунок 2.68 – Диспетчер параметров листа

2. Формат листа. Список стандартных форматов, разрешенных для применения на выбранном устройстве. Если плоттер еще не выбран, в списке перечислены все поддерживаемые форматы.

3. Количество экземпляров. Определяет количество выводимых на печать экземпляров. Параметр недоступен при печати в файл.

4. Область печати. Задание выводимой на печать части чертежа. Опция "Что печатать" определяет, какая часть чертежа будет напечатана.

– Лист/, вычисленный относительно координат 0,0 в листе. При выводе на печать из вкладки "Модель" распечатывается вся область рисования, определяемая границами сетки. Если вид на текущем видовом экране Преде-

лы. При выводе на печать листа распечатываются все объекты в пределах печатаемой области для указанного формата листа начиная с точки отличается от вида в плане, параметр работает аналогично параметру "Границы".

- Границы. Печать той части чертежа, которая в данный момент содержит объекты. Печатаются все объекты текущего пространства. Чертеж может быть регенерирован для пересчета границ перед печатью.

- Показать (Экран). Печать текущего видового экрана (для выбранной вкладки "Модель") или текущего вида пространства листа.

- Вид. Печать вида, предварительно описанного и сохраненного с помощью команды ВИД. Возможен выбор именованного вида из списка. Если в чертеже нет именованных видов, параметр недоступен. При выборе параметра "Вид" рядом появляется список всех именованных видов, сохраненных в текущем чертеже. Возможен выбор вида для печати из списка.

- Рамка. Вывод фрагмента чертежа, указанного пользователем. После выбора рамки кнопка "Рамка" становится доступной. Нажав кнопку "Рамка", пользователь либо указывает два противоположных угла на экране, либо вводит их координаты.

5. Смещение от начала. Определяет смещение области печати относительно левого нижнего угла печатаемой страницы или края бумаги, в зависимости от установки параметра "Отсчет смещения чертежа" (диалоговое окно "Параметры", вкладка "Печать/Публикация"). Выбранный вариант смещения приводится в круглых скобках после названия группы параметров "Смещение от начала" в диалоговом окне "Печать".

Печатаемая область листа чертежа определяется выбором устройства вывода и выделяется на листе штриховой линией. При выборе другого печатающего устройства область печати может измениться.

Возможно смещение геометрии на листе путем ввода положительного или отрицательного значения в окна смещения по осям X и Y. Единицами смещения для плоттера являются дюймы или миллиметры на листе.

6. Центрировать. Автоматическое определение смещений по X и Y так, чтобы чертеж оказался в центре листа. Данный параметр недоступен, если в качестве области печати указан "Лист".

X - Указывается начальная точка чертежа в направлении X по отношению к точке, определяемой текущим значением параметра "Отсчет смещения чертежа".

Y- Задание начала координат печати в направлении оси Y относительно настройки параметра "Определение смещения печати".

7. Масштаб печати.

Управление относительным размером единиц чертежа для печатаемых единиц. По умолчанию для листа устанавливается масштаб 1:1. Для вкладки "Модель" по умолчанию устанавливается значение масштаба "Вписать".

- Вписать. Печатаемая область масштабируется для вписывания в указанный формат листа; пользовательский масштабный коэффициент отображается в полях "Масштаб", "дюйм =" и "единицы".

- Масштаб. Точное значение масштаба печати. *Пользовательский* является нестандартным масштабом. Он задается путем указания соответствия между дюймами (миллиметрами) на чертеже и единицами чертежа.

Определяет соответствие установленного числа единиц чертежа в дюймах, миллиметрах или пикселях единицам чертежа.

- Дюйм/мм/пиксел. Задание дюймов или мм для отображения единиц в диалоговом окне "Печать". Значение по умолчанию зависит от формата листа и изменяется при каждом выборе нового формата. Значение в пикселях доступно только при выборе растрового вывода.

- Единицы. Определяет соответствие единиц чертежа указанному количеству дюймов, миллиметров или пикселов.

- Масштабировать веса линий. Масштабирование весов линий пропорционально масштабу печати. Веса линий, как правило, обозначают ширину линий объектов чертежа, и печать для них выполняется с шириной линий, не зависящей от масштаба печати.

8. Таблица стилей печати (назначение перьев)

Установка текущей таблицы стилей печати, редактирование имеющихся и создание новых таблиц.

9. ВЭкраны с тонированием.

Задается способ вывода на печать тонированных и визуализированных видовых экранов и определяются их уровни разрешения и количество точек на дюйм (т/дюйм). Параметры визуализированных видовых экранов недоступны в AutoCAD LT.

– Обычный. Печать объектов в том виде, в каком они отображаются на экране. (Доступно в AutoCAD LT.)

– Каркас из предыдущих версий. Печать объектов в каркасном представлении независимо от того, как они отображаются на экране, с помощью команды РЕЖИМРАСКР из предыдущих версий. (Доступно в AutoCAD LT.)

– Скрытие линий из предыдущих версий. Печать объектов с удалением скрытых линий независимо от того, как они отображаются на экране, с помощью команды РЕЖИМРАСКР из предыдущих версий. (Доступно в AutoCAD LT.)

– Концептуальный. Печать объектов с применением визуального стиля "Концептуальный", независимо от того, как они отображаются на экране.

– Скрытие линий. Печать объектов с удалением скрытых линий, независимо от того, как они отображаются на экране.

– Реалистичные. Печать объектов с применением визуального стиля "Реалистичный", независимо от того, как они отображаются на экране.

– Тонированный. Печать объектов с применением визуального стиля "Тонированный", независимо от того, как они отображаются на экране.

– Тонированный с ребрами. Печать объектов с применением визуального стиля "Тонированный с кромками", независимо от того, как они отображаются на экране.

– Оттенки серого. Печать объектов с применением визуального стиля "Оттенки серого", независимо от того, как они отображаются на экране.

– Эскизный. Печать объектов с применением визуального стиля "Эскизный", независимо от того, как они отображаются на экране.

– Каркас. Печать объектов в каркасном представлении, независимо от того, как они отображаются на экране.

– Просвечивание. Печать объектов с применением визуального стиля "Просвечивание", независимо от того, как они отображаются на экране.

– С визуализацией. Печать объектов с визуализацией, независимо от того, как они отображаются на экране.

– Качество. Задание разрешения для печати тонированных и визуализированных видовых экранов.

Можно выбрать один из следующих параметров разрешения:

– Черновое. Визуализированные и тонированные виды пространства модели печатаются в каркасном режиме.

– Просмотр. Визуализированные и тонированные виды пространства модели печатаются с разрешением 1/4 от текущего разрешения устройства печати — максимум 150 т/д.

– Нормальное. Визуализированные и тонированные виды пространства модели печатаются с разрешением 1/2 от текущего разрешения устройства печати — максимум 300 т/дюйм.

– Презентационное. Визуализированные и тонированные виды пространства модели печатаются с текущим разрешением устройства печати — максимум 600 т/д.

– Максимум. Визуализированные и тонированные виды пространства модели печатаются с текущим разрешением устройства печати без установки максимального ограничения.

– Пользовательское. Визуализированные и тонированные виды пространства модели печатаются с разрешением, заданным в поле "Т/дюйм". Указанное пользователем разрешение не может превышать текущее разрешение устройства печати.

10. Параметры печати.

Параметры, относящиеся к весам линий, прозрачности, стилям печати, способу и порядку вывода объектов на печать.

- Фоновая печать. Признак выполнения печати в фоновом режиме. (системная переменная BACKGROUND PLOT)

- Учитывать веса линий. Печать с учетом весов линий, назначенных объектам и слоям.

- Прозрачность при печати. Определяет возможность вывода объектов на печать с учетом прозрачности. Данный параметр используется только при печати чертежей с прозрачными объектами.

- Учитывать стили печати. Определяет, учитываются ли во время печати стили печати, применяемые к объектам и слоям.

- Объекты листа последними. Печать объектов пространства модели в первую очередь. При положительном ответе первыми выводятся объекты пространства листа.

- Скрывать объекты листа. Определяет, применяется ли операция скрывания к объектам на видовых экранах листа. Доступно только для вкладок-листов. Эффект от данного параметра проявляется в образце для предварительного просмотра, но не в листе.

- Штемпель вкл. Включение режима нанесения штампов. Наносит штамп в определенном углу каждого чертежа и/или записывает соответствующую информацию в файл журнала.

О параметрах штампов можно узнать из описания диалогового окна "Штамп чертежа", в котором можно задать информацию, проставляемую на штампе, например имя чертежа, дату и время, масштаб печати и т. д. Для открытия диалогового окна "Штамп чертежа" следует поднять флажок "Штамп вкл" и нажать кнопку "Параметры штампа".

Диалоговое окно "Штамп чертежа" можно также получить, нажав кнопку "Штамп чертежей" на вкладке "Печать/Публикация" диалогового окна "Параметры".

- Кнопка "Параметры штампа". Отображает диалоговое окно "Штамп чертежа", если в диалоговом окне "Печать" выбрана опция "Штамп вкл".

11. Ориентация чертежа. Задание ориентации чертежа для плоттеров, поддерживающих ее книжный и альбомный варианты. Текущая ориентация выбранного листа иллюстрируется значком. Буква на значке указывает на расположение чертежа на листе.

– Книжная. Чертеж ориентируется таким образом, что при печати верх страницы соответствует короткой стороне листа.

– Альбомная. Чертеж ориентируется таким образом, что при печати верх страницы соответствует длинной стороне листа.

– Перевернуть. Поворот чертежа на 180 градусов.

– Значок. Указывает ориентацию выбранной бумаги, и задает ориентацию чертежа на странице в виде, совпадающем с расположением букв на бумаге.

2.3 Построение технологических схем и спецификаций

2.3.1 Построение сложных объектов

Окно Палитры инструментов (рис. 2.69) предназначено для быстрого вызова и вставки таких элементов оформления, как штриховки, блоки и заливки, шаблоны которых собраны на различных палитрах.

Вызвать Инструментальную палитру можно несколькими способами:

- Сервис – Палитры – Инструментальные палитры.
- Вкладка Вид – Палитры.
- Сочетание клавиш Ctrl+3.

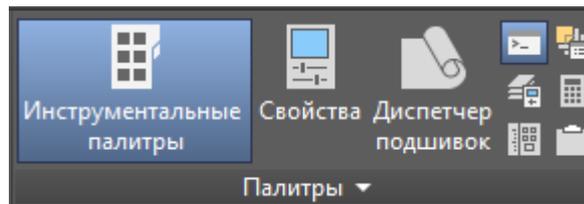


Рисунок 2.69 – Окно Палитры

На инструментальных палитрах (рис. 2.70) содержатся вкладки с различными палитрами, на которых находятся уже готовые блоки.

Выделенный блок можно просто перетащить на чертеж, либо, нажав на пиктограмму, ввести в командной строке требуемые параметры.

Некоторые блоки отмечены значком желтой молнии. Это динамические блоки. При вставке можно изменять их размеры и даже конфигурацию.

Любой блок можно открыть в редакторе блоков и внести изменения.

Например, вставим элемент балки двутавровой. При нажатии на пиктограмму, в командной строке появляются запросы на указание масштаба, угла поворота и точки вставки.

Запросы:

Команда: Блок Балка двутавровая широкополочная:

Точка вставки или [Базовая точка/Масштаб/Поворот]: Масштаб
Масштаб по осям X, Y и Z <1>: 0.5

Точка вставки или [Базовая точка/Масштаб/Поворот]: Поворот
Угол поворота <0>: 90

Точка вставки или [Базовая точка/Масштаб/Поворот]:

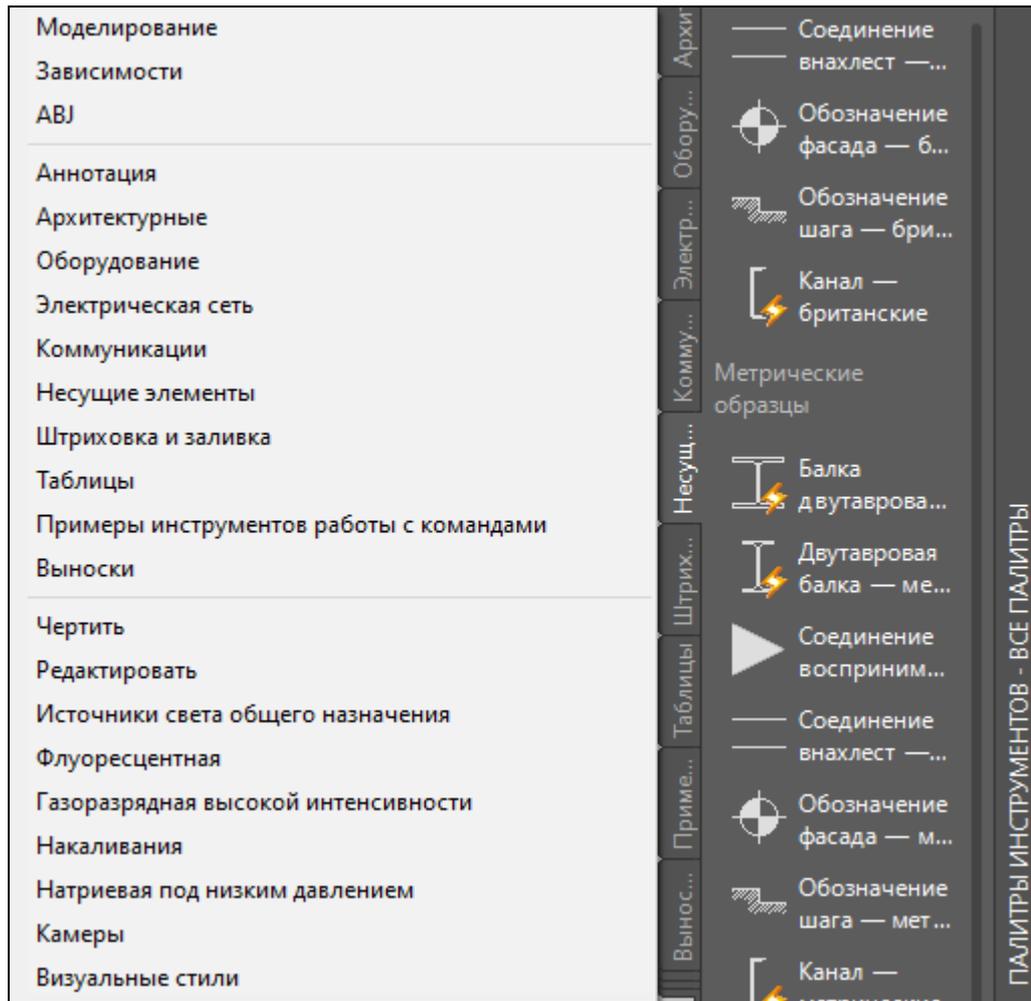


Рисунок 2.70 – Инструментальные палитры

Для создания новой палитры (рис.2.71) необходимо щелкнуть правой кнопкой мыши на свободном месте инструментальной палитры, выбрать Создать палитру и ввести новое название. Затем созданные блоки можно просто перетянуть на эту палитру.

В процессе работы палитры можно переименовывать и удалять.

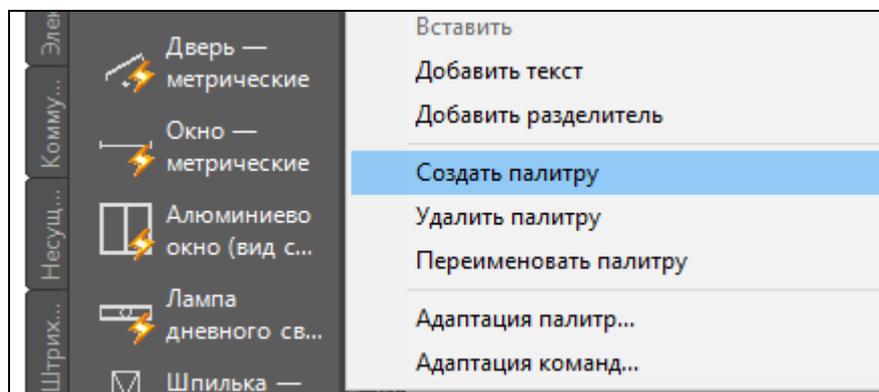


Рисунок 2.71 - Создание новой палитры

Мультилиния

Мультилиния представляет собой набор параллельных линий, создающихся одновременно с помощью одной команды. Количество линий, составляющих мультилинию, может варьироваться от 2 до 16.

Использование мультилиний особенно полезно и удобно при вычерчивании линий автодорог, трубопроводов, линий стен и т. д.

Построение мультилиний в AutoCAD осуществляется с помощью команды Mline (Млиния), вызвать которую можно одним из следующих способов (рис. 2.72):

1. Ввести в командной строке название команды на русском или английском языке.
2. Панель Рисование – Мультилиния.

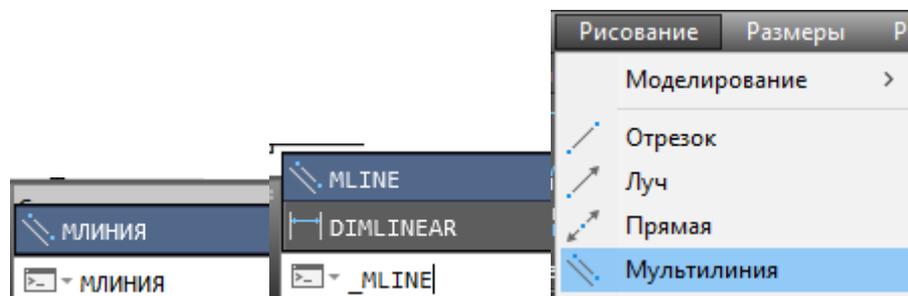


Рисунок 2.72 – Способы вызова команды Мультилиния

Перед началом рисования можно задать режим расположения, масштаб мультилинии (рис. 2.73). При построении мультилинии можно использовать стиль Standard, в котором есть два элемента, или присвоить стиль, созданный ранее.

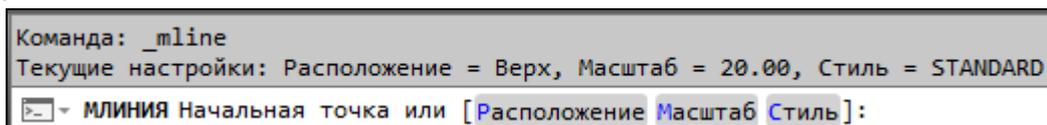


Рисунок 2.73 – Параметры команды Мультилиния

Далее можно либо сразу приступить к построению мультилинии с параметрами, установленными по умолчанию (верхняя строка запроса), либо можно изменить эти параметры, выбрав одну из опций:

- Расположение (Justification) - выбрав данную опцию, вы сможете указать, как должна строиться мультилиния;

Центр (Zero) - мультилиния строится путем указания начальной и конечной точек оси мультилинии (условной невидимой линии, проходящей через ее центр);

Верх (Top) и Низ (Bottom) - мультилиния строится путем указания начальных и конечных точек крайней верхней или крайней нижней линии мультилинии.

- Масштаб (Scale) - опция, позволяющая изменить общую ширину линии относительно стандартного размера. Стандартный размер устанавливается для каждого стиля индивидуально в настройках стиля.

- Стиль (Style) - позволяет задать другой стиль для мультилинии. Стиль - это имеющий имя набор параметров. Просмотреть список всех доступных стилей можно, введя в командную строку вопросительный знак.

Можно добавить команду Мультилиния на Ленту, на вкладку Главная, панель Рисование. Для этого необходимо выбрать команду Пользовательский интерфейс на вкладке Управление. Откроется окно "Адаптация пользовательского интерфейса" (рис. 2.74). Установить курсор на вкладке 2D-рисование. Справа в поле значков щелкнуть по значку Кольцо, который сразу отразится на вкладке 2D-рисование. В поле Список команд выбрать команду Мультилиния и, не отпуская мышку, перетянуть на вкладку 2D-рисование под значок кольца. Когда курсор преобразуется в горизонтальную линию, отпустить кнопку мышки. Команда Мультилиния появится на панели Рисование.

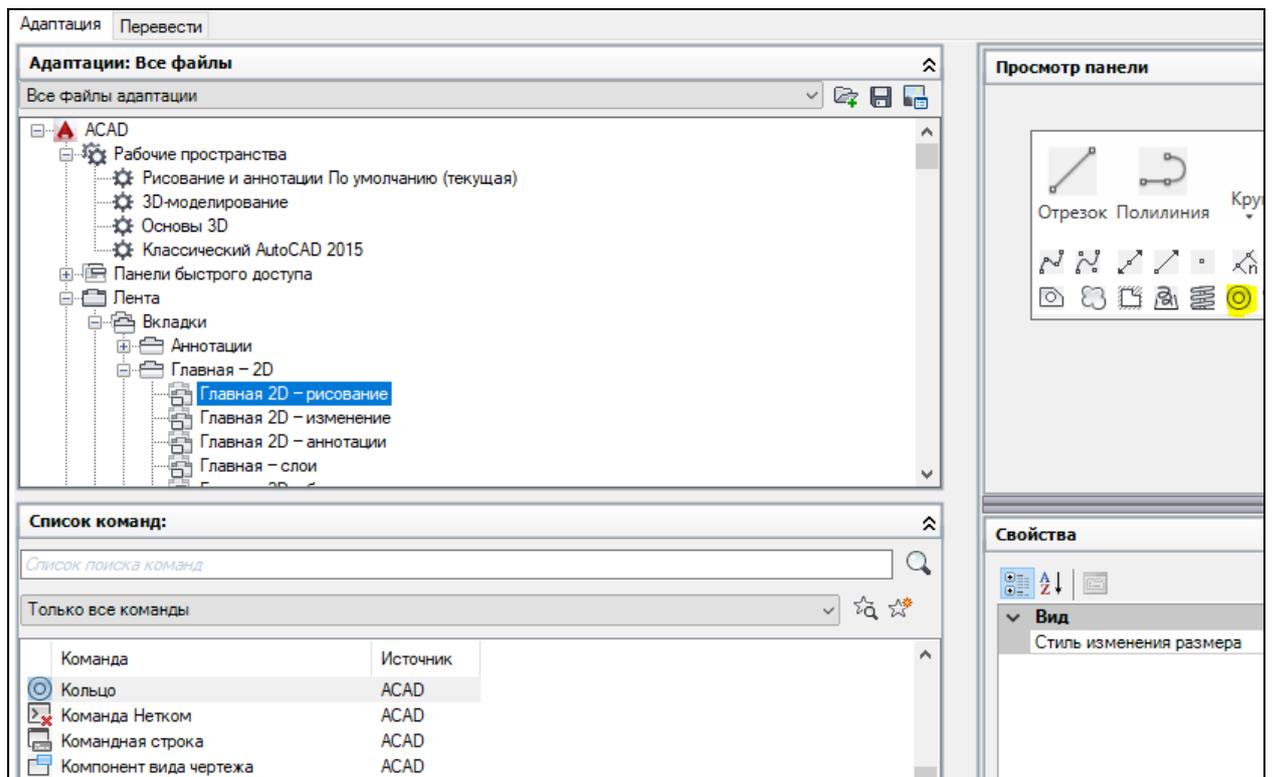


Рисунок 2.74 - Адаптация пользовательского интерфейса

Стили мультилиний

Создание нового стиля мультилиний осуществляется с помощью команды MLSTYLE (МЛСТИЛЬ), которую можно вызвать:

- из строки меню Формат (Format) - Стиль Мультилиний (Multiline Style);

- вводом в командную строку: `_mlstyle`.

После вызова этой команды AutoCAD выводит на экран диалоговое окно Стиль Мультилиний (рис. 2.75), в котором и производятся все необходимые действия по созданию стиля. При нажатии на кнопку "Создать" можно создать новый стиль мультилинии.

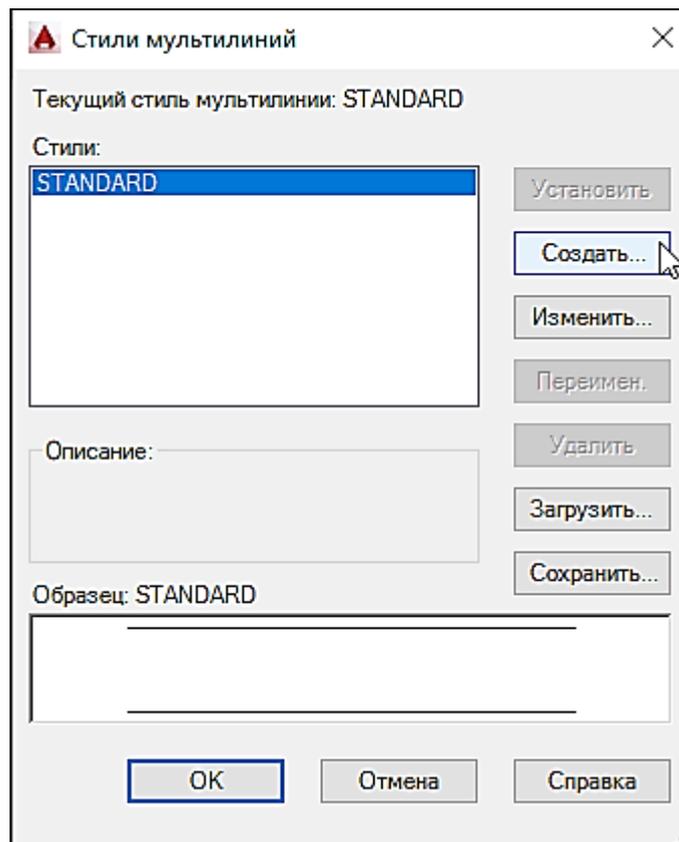


Рисунок 2.75 – Создание нового стиля мультилиний

В диалоговом окне "Создать новый стиль мультилинии" вводим имя стиля мультилинии.

В диалоговом окне "Новый стиль мультилинии" (рис. 2.78) выбираем параметры стиля мультилинии:

Описание не является обязательной характеристикой стиля мультилиний. Его длина не должна превышать 255 символов, включая пробелы.

В поле Элементы, приведен перечень всех элементов (линий-элементов), входящих в мультилинию. Чтобы изменить свойства какой-либо линии-элемента, ее необходимо сначала выбрать в этом списке.

Для добавления еще одной линии-элемента используется кнопка **Добавить**, а для удаления - кнопка **Удалить**.

В поле **Смещение** указывается расстояние линии-элемента от оси мультитилинии. Нулевое значение означает, что линия-элемент будет строиться прямо по оси. В используемом по умолчанию стиле **Standard** составляющие элементы-линии строятся с отступом 0,5 по разные стороны от оси (значения 0,5 и -0,5).

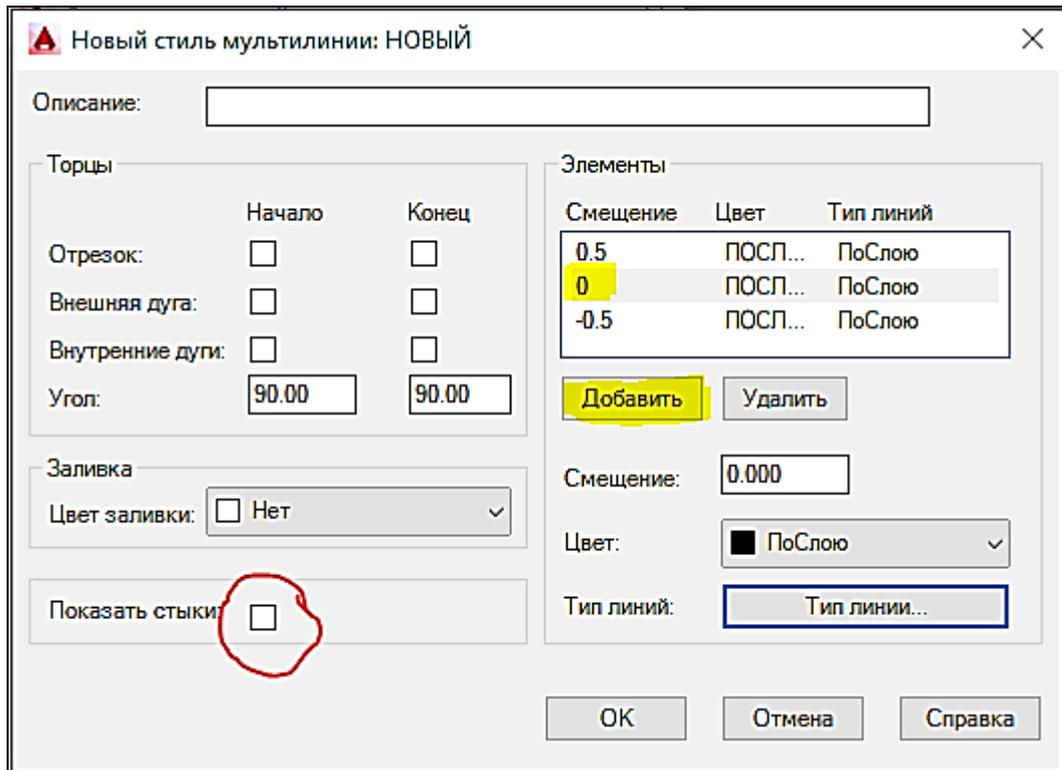


Рисунок 2.78 – Установка параметров нового стиля мультитилинии

В раскрывающемся списке **Цвет** можно выбрать один из стандартных цветов для выделенного элемента. Если же ни один из стандартных вариантов цвета вам не подходит - в конце списка указать команду **Выбор цвета**, и на экране появится одноименное диалоговое окно, в котором можно указать нужный цветовой оттенок.

Для задания типа линии нажимаем на кнопку **Тип линии**. При этом откроется диалоговое окно **Выбор типа линий**. Изначально на выбор предлагается только три типа линии. Однако можно добавить ("подгрузить") другие типы, нажав на кнопку **Загрузить**.

В левой половине окна задания параметров мультитилинии **Стиль Мультитилиний** находятся настройки, относящиеся к мультитилинии целиком.

В частности, в группе **Закрашивание** можно включить или выключить заливку мультитилинии, выбрав для нее какой-либо цвет. По умолчанию ника-

кого цвета не выбрано, соответственно закрашивание не производится, а нажав на кнопку Цвет, можно выбрать цвет заливки

Команда Показать стыки указывает дополнительной линией стык изгиба мультилинии.

После создания нового стиля, необходимо сохранить его в файле. Иначе он будет доступен только в данном чертеже. Чтобы сохранить созданный стиль, следует в окне Стиль Мультилинии нажать кнопку Сохранить. По умолчанию стили мультилинии хранятся в файле acad.mln.

В связи с тем, что элементы мультилинии рассматриваются как единый объект, многие команды общего редактирования для них недоступны. Поэтому в AutoCAD имеется набор команд, ориентированный исключительно на работу с мультилиниями. Основной из них является команда Mledit. Для редактирования мультилинии необходимо ее выделить и выбрать в контекстном меню Редактировать мультилинию или ввести в командной строке Mledit. После вызова команды Mledit открывается диалоговое окно Инструменты редактирования мультилиний, изображенное на рис. 2.79.

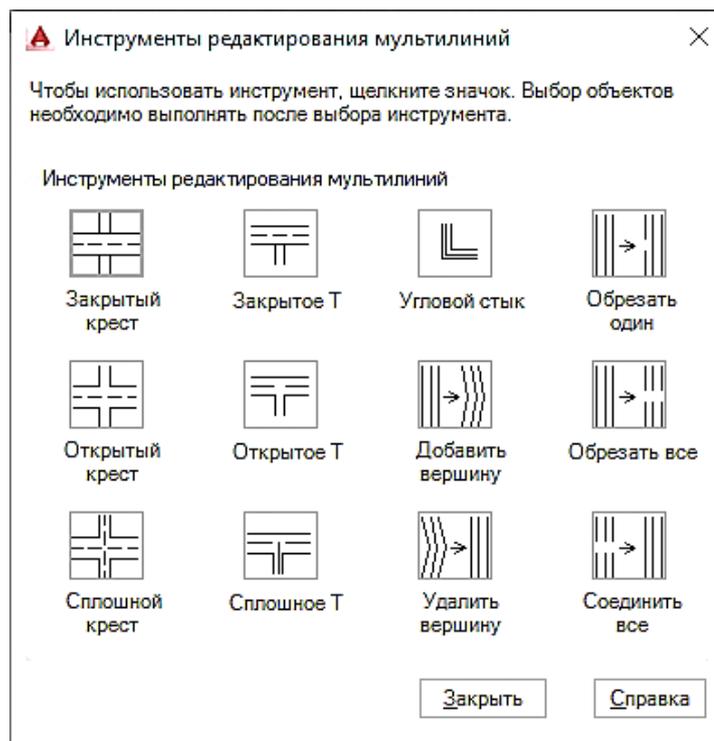


Рисунок 2.79 – Инструменты редактирования мультилиний

В этом окне представлены пиктограммы 12 операций, расположенные в четырех столбцах по три в каждом. При выборе той или иной пиктограммы в нижней части окна появляется подсказка с наименованием соответствующей операции. Для подтверждения своего выбора достаточно нажать кнопку ОК. Далее на запрос выбрать первую мультилинию нужно щелкнуть левой кнопкой мыши по первой линии, а на запрос выбрать вторую мультилинию – по

второй. При оформлении пересечений существенное значение имеет то, какая именно из линий была выбрана первой, а какая – второй. Это влияет на последовательность наложений (перекрытий) мультилиний друг на друга – верхней оказывается мультилиния, выбранная в последнюю очередь.

В первом столбце окна расположены кнопки изменения типа взаимного пересечения двух мультилиний.

"Закрытый крест" – в пересечении вырезается содержимое первой выделенной мультилинии, а содержимое второй остается без изменений.

"Открытый крест" – в пересечении удаляются все линии первой выбранной мультилинии и крайние линии второй.

"Сплошной крест" – в пересечении попарно соединяются все внутренние и удаляются крайние линии второй выбранной мультилинии.

Во втором – кнопки оформления окончания одной мультилинии на другой (форматирование стыков).

"Закрытый тавр" – отсекается продолжение первой выбранной мультилинии со стороны, противоположной точке выбора; в пересечении вырезается содержимое первой мультилинии, а содержимое второй остается без изменений.

"Открытый тавр" – отсекается продолжение первой выбранной мультилинии со стороны, противоположной точке выбора, а в пересечении удаляется крайняя линия второй выбранной мультилинии.

"Сплошной тавр" – отсекается продолжение первой выбранной мультилинии со стороны, противоположной точке выбора, а в пересечении попарно соединяются все внутренние линии, и удаляется крайняя линия второй мультилинии.

В третьем – кнопки обработки углового пересечения двух мультилиний или одной мультилинии (а также операций с вершинами).

"Угловой стык" применяется для оформления углового пересечения.

Изменение количества вершин мультилинии выполняется в основном для добавления или удаления на ней "переломов".

В четвертом – кнопки операций разрывов (соединения) отдельных либо всех линий в мультилинии.

Калькулятор

Калькулятор "БыстрКальк" (рис. 2.80) выполняет весь спектр математических, научных и геометрических вычислений, создает и использует переменные, а также преобразует единицы измерений.

Вызвать калькулятор можно на вкладке Главная, панель Утилиты.

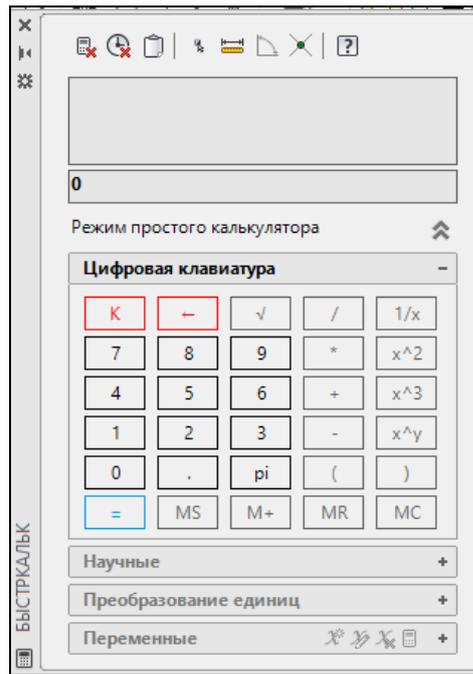


Рисунок 2.80 - Калькулятор
Средства измерения

Команда *ИЗМЕРИТЬГЕОМ* позволяет измерить расстояния, радиусы, углы, площади и объем для выбранных объектов или заданной на экране последовательности точек. Эта команда также доступна из панели *Утилиты* закладки ленты *Главная*. Опция команды по умолчанию - измерение расстояния. При выборе другого инструмента измерения он устанавливается по умолчанию до конца текущей сессии AutoCAD или пока не будет выбран другой инструмент (рис. 2.81). Результаты измерений отображаются в командной строке.

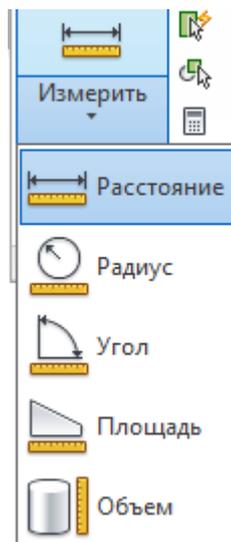


Рисунок 2.81 – Панель Утилиты

Опция **Расстояние** позволяет измерять расстояние между двумя точками, при этом отображаются расстояние, дельта X , дельта Y и угол на плоскости XU .

Опцию **Радиус** можно использовать для отображения радиуса выбранной дуги или окружности.

Опция **Угол** измеряет угол выбранных дуг, окружностей, линий или между указанными вершинами.

Опция **Площадь** позволяет по выбранным точкам или объектам вычислить площадь внутренней области.

Используя опции **Добавить** или **Вычесть**, можно определять суммарную площадь. По мере указания точек или выбора объектов занимаемая ими площадь динамически подсвечивается, так что можно видеть то, что выбрано.

Дополнительные параметры в опции **Площадь** позволяют переключаться между функциями измерения линий и дуг, что дает возможность измерять площадь объектов с дуговыми сегментами.

2.3.2 Создание таблиц

Таблица представляет собой сложный объект, в котором данные организованы по строкам и столбцам. Инструмент Таблица можно вызвать на панели Аннотация вкладки Главная либо на вкладке Аннотации (рис. 2.82).

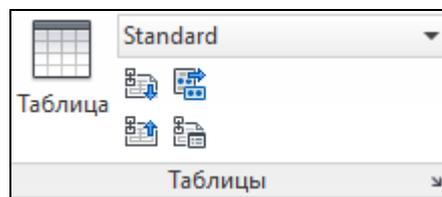


Рисунок 2.82 – Вызов таблицы

После вызова команды появляется диалоговое окно "Вставка таблицы" (рис. 2.83), в котором требуется указать параметры таблицы.

- Стиль таблиц.

Данное поле содержит список, имеющихся в стилях таблиц Автокад. Справа от списка находится кнопка создания нового стиля таблицы.

- Параметры вставки.

Вставить таблицу можно тремя способами:

1. С пустой таблицы - вставка таблиц Автокад с пустыми ячейками, которые необходимо будет потом заполнить.

2. На основе связи с данными - вставка таблицы, ячейки которой связаны с внешними данными электронной таблицы Excel. В этом способе производится вставка таблицы Автокада, ячейки которой подгружают данные из таблицы Excel.

3. На основе данных объекта на чертеже (извлечение данных) - запуск Мастера извлечения данных.

- Просмотр. Предварительный вид вставляемой таблицы в Автокад.
- Способ вставки.

Запрос точки вставки - таблица вставляется в чертеж по точке вставки. Данная точка будет располагаться в верхнем левом углу таблицы.

Если в таблице определено направление построения таблицы снизу вверх, точка вставки соответствует левому нижнему углу таблицы.

Запрос занимаемой области - при вставке таблицы необходимо указать две ее диагональные точки. Ширина столбцов и высота строк ячеек таблицы при таком методе вставки будет зависеть от указанной области.

- Параметры столбцов и строк.

Можно указать количество столбцов и строк таблицы, а также соответственно их ширину и высоту. Минимальная ширина столбцов - один печатный символ. Высота текста в ячейке и его отступы от границ ячейки и другие параметры таблицы задаются в стиле таблицы.

- Задание стилей ячеек.

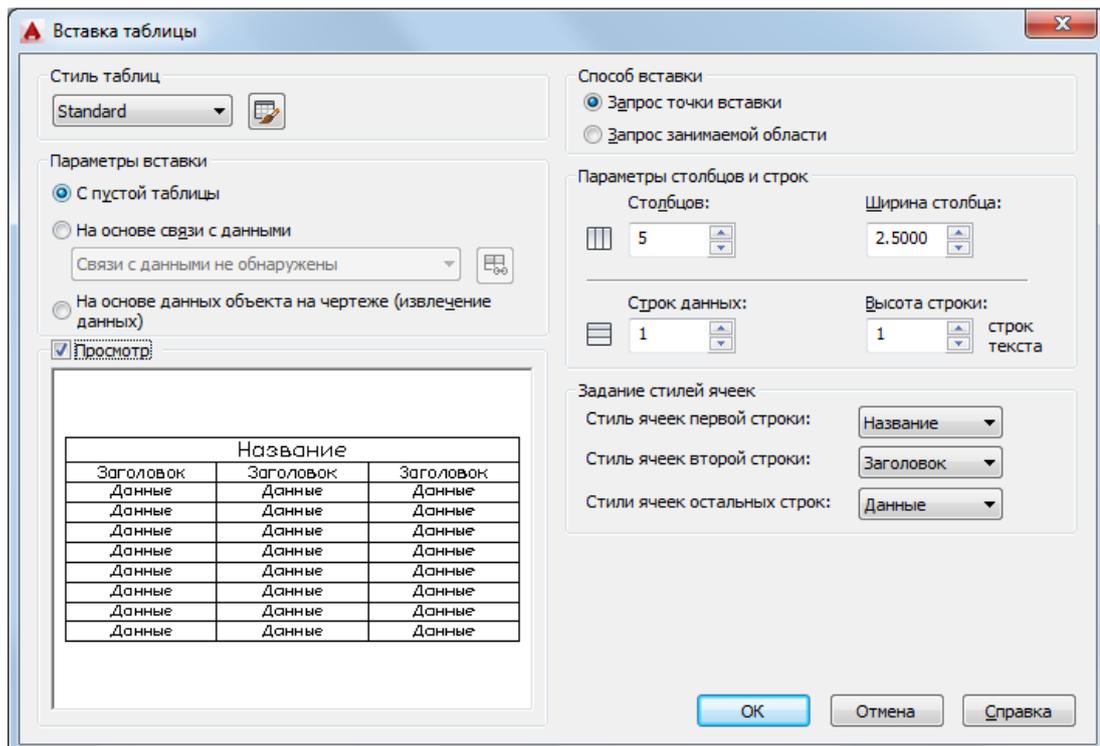


Рисунок 2.83 – Диалоговое окно Вставка таблицы

В таблице первая строка – название, вторая – заголовки столбцов, третья и последующие строки – данные.

Стили ячеек таблицы заданы ее стилем, но можно изменить стиль ячеек первой, второй и последующих строк при ее вставке. При изменении рас-

положения стилей ячеек в таблице может измениться сам вид таблицы, но не ее стиль.

После заполнения всех полей нажимаем кнопку "ОК". Теперь необходимо указать верхний левый угол таблицы.

Как только таблица будет вставлена в чертеж, появится приглашение на ввод данных в первую ячейку Название, а на ленте инструментов появится временная вкладка "Текстовый редактор таблицы" (рис. 2.84).

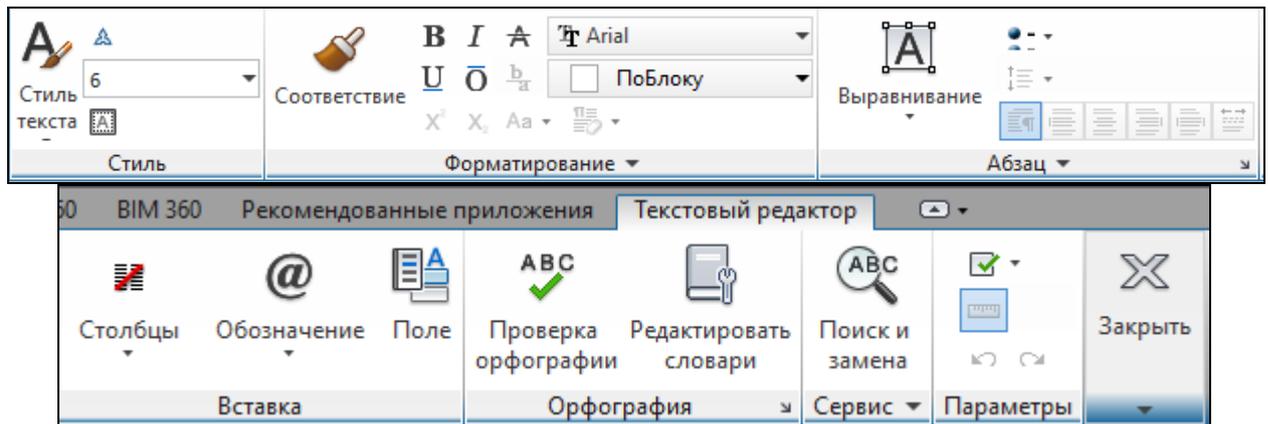


Рисунок 2.84 – Текстовый редактор таблицы

На вкладке находятся следующие панели, с помощью которых можно редактировать текст в ячейках:

- **Стиль.** Изменение стиля и высоты текста. Размещение непрозрачного фона позади текста.
- **Форматирование.** Изменение начертание и имени шрифта. Формат по образцу.
- **Абзац.** Выравнивание текста относительно границ ячеек. Форматирование абзаца и междустрочного интервала.
 - Вставка символов и поля.
 - Проверка орфографии.
 - Поиск и замена текста.
 - Параметры. Отражается тип шрифта.

При выделении любой ячейки таблицы на ленте появляется временная вкладка Ячейка таблицы (рис. 2.85).

На этой вкладке можно выполнять следующие операции:

- вставка и удаление строк;
- вставка и удаление столбцов;
- объединение ячеек;
- выравнивание и обрамление ячеек;
- форматирование ячеек.

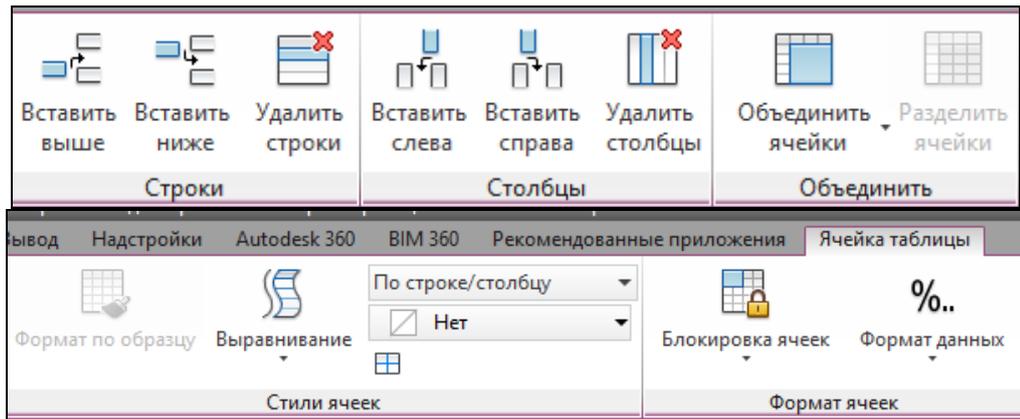


Рисунок 2.85 – Временная вкладка Ячейка таблицы

Создание стиля таблицы

Создание нового стиля таблицы или редактирование существующего осуществляется с помощью команды Управление стилями таблицы, которая находится на вкладке Аннотации или Главная.

При открытии окна Стили таблицы можно нажать на кнопку Создать для создания нового стиля (далее ввести имя нового стиля) или на кнопку Редактировать для изменения существующего стиля. Открывается окно нового стиля таблиц (рис. 2.86).

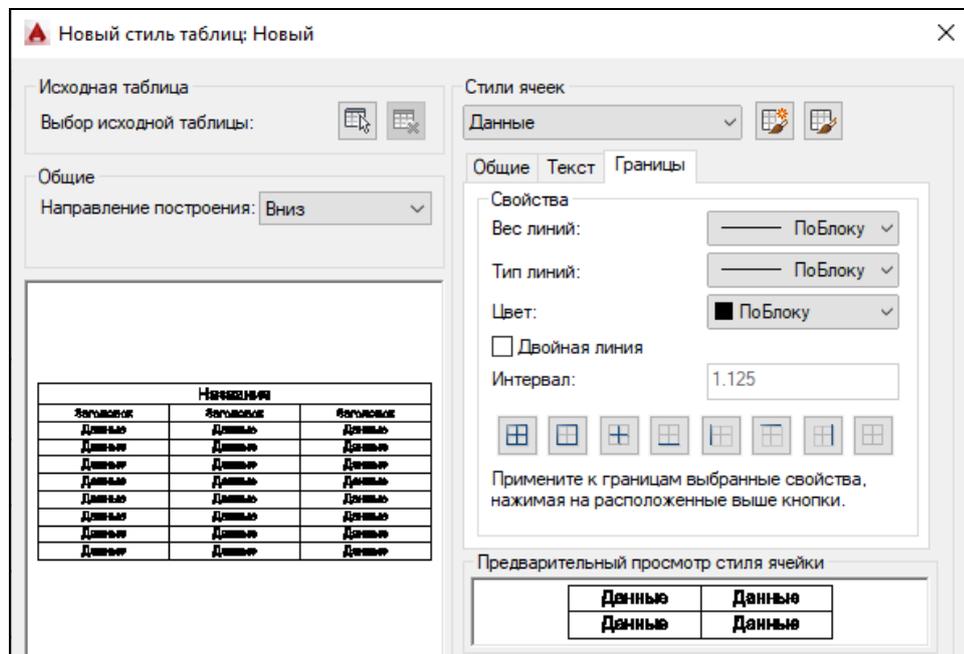


Рисунок 2.86 – Создание нового стиля таблицы

Если при создании нового стиля не выделена таблица, то при помощи кнопки Исходная таблица можно указать редактируемую таблицу.

Поле Общие отвечает за направление таблицы, т.е. заглавие таблицы может находиться вверху или внизу таблицы. Ниже на образце будут отражены результаты изменений.

Изменить параметры ячеек можно в поле Стили ячеек (рис.2.87).

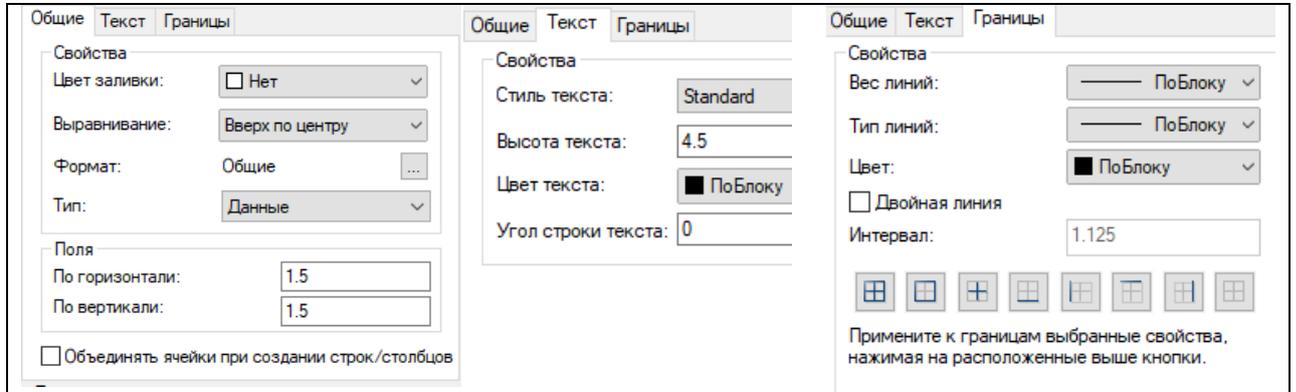


Рисунок 2.87 – Изменение параметров ячеек

Вкладка Общие содержит команды, изменяющие цвет заливки ячейки, выравнивание и формат. Изменение поля по горизонтали и вертикали позволяет располагать текст ближе или дальше от границ ячейки. Перед изменением необходимо выбрать тип ячейки: название, заголовок или данные.

Вкладка Текст устанавливает стиль, высоту и цвет текста.

Вкладка Границы позволяет выбрать тип, вес и цвет линии границы и применить эти свойства, выбрав расположение границы.

Внизу в поле предварительного просмотра можно увидеть результаты изменений параметров стиля ячеек.

При выделении таблицы появляются ручки, с помощью которых можно редактировать размер таблицы (рис. 2.88). "Желтые ручки" позволяют изменить ширину столбцов или высоту строк всей таблицы. При помощи синего квадрата можно изменить размер только одного столбца или одной строки. "Красный треугольник" разрывает таблицу.

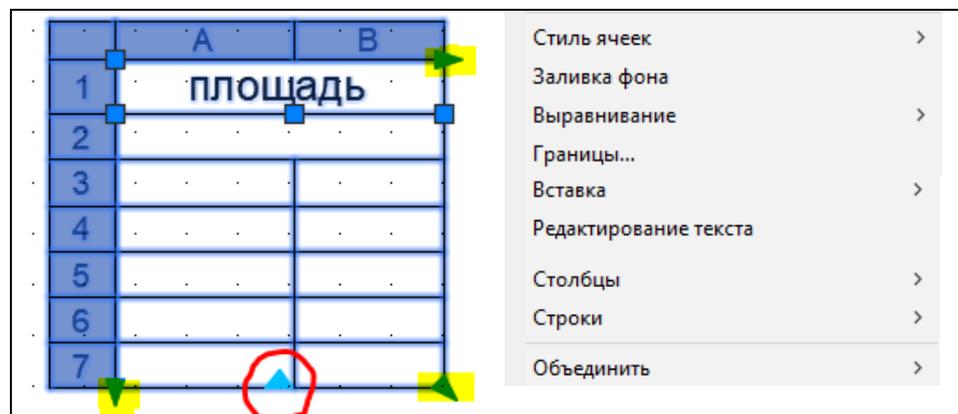


Рисунок 2.88 – Редактирование таблицы

Редактировать таблицу удобно с помощью контекстного меню. Можно изменять стили ячеек, фон, границы, вставлять, удалять и объединять строки и столбцы.

Для вставки формулы или поля в таблицу необходимо выделить ячейку и в контекстном меню выбрать команду Вставка – Формула – Уравнение (рис. 2.89).

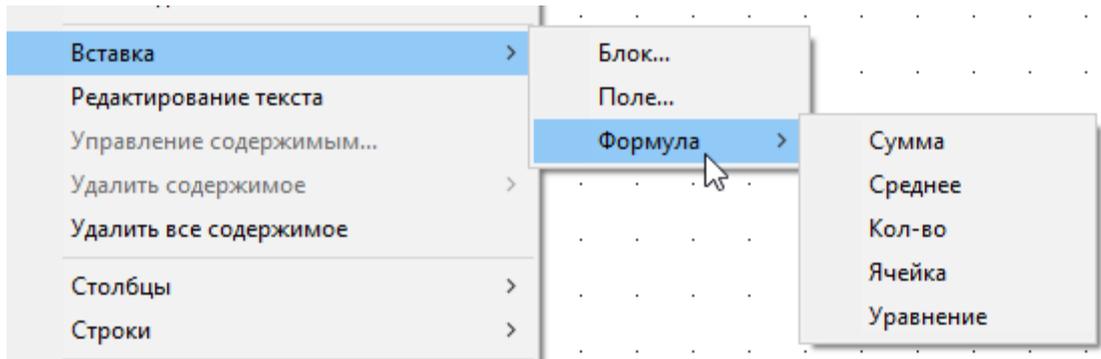


Рисунок 2.89 – Вставка формулы в таблицу

Пример 4.1. Необходимо вставить значение площади нарисованной замкнутой фигуры (рис. 2.90).

В контекстном меню выделенной ячейки выбираем команду Вставка – Поле. В полях Категории и Имена выбираем Объект. В поле Тип объекта щелкаем по значку курсора, временно возвращаемся в графическое окно и выделяем фигуру. В поле Свойства указываем Площадь. Затем устанавливаем формат и точность единиц измерения. На вкладке Дополнительный формат можно добавить коэффициент преобразования мм в м и обозначение единиц измерения м².

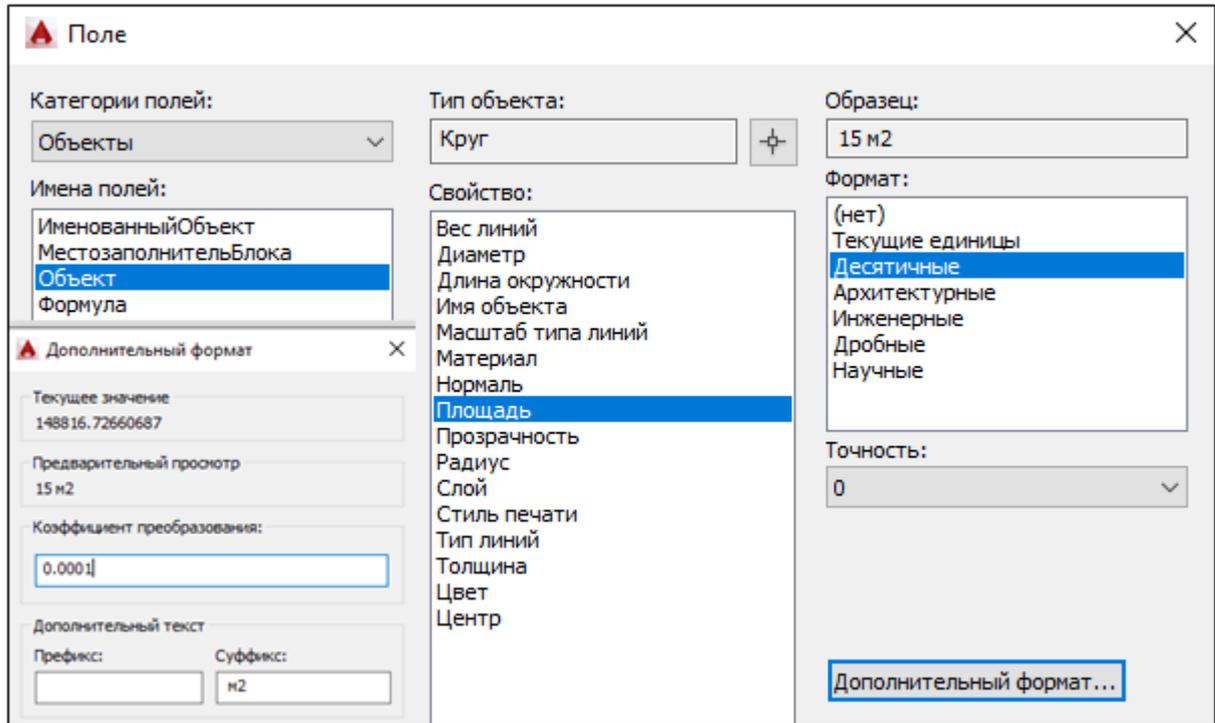


Рисунок 2.90 – Вычисление площади

2.3.3 Блоки, внешние ссылки и динамические блоки

Блоком называется один или группа объектов AutoCAD, которые создаются специальной командой, имеют имя для идентификации и обрабатываются как единый объект.

Блоки создаются из стандартных объектов AutoCAD путем их объединения специальной командой. В процессе создания блоков им присваиваются имена, по которым они затем распознаются и вставляются в чертеж. Блоки можно использовать в следующих целях:

- для создания библиотеки часто используемых элементов чертежей, которые затем вставляются в другие чертежи;
- для изменения отдельных элементов путем переопределения блоков;
- для экономии дискового пространства и уменьшения размера файла (в базе чертежа блок сохраняется как один объект и занимает значительно меньше места);
- для связи графических элементов чертежа с текстовым описанием.

Большим преимуществом блоков является то, что достаточно отредактировать только лишь одно определение блока (т.е. основное описание блока), и все остальные вхождения (элементы, которые присутствуют на чертеже) автоматически изменяться. В противном случае, пришлось бы редактировать каждый объект отдельно или вносить изменения для одного объекта и далее копировать его и заново вставлять.

С блоком можно работать так же, как с любым объектом чертежа. Можно использовать привязку к характерным точкам отдельных объектов блока, подрезать и продолжать объекты чертежа по объектам блока. Однако изменять отдельные объекты блока нельзя.

Создать блок можно при помощи команды Создать блок, находящейся на панели Определение блока на вкладке Вставка или на вкладке Главная (рис. 2.91).

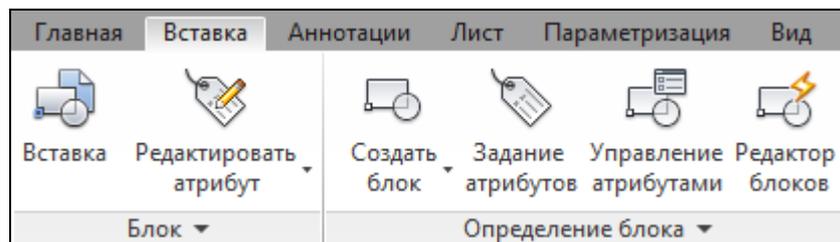


Рисунок 2.91 – Панели для создания и вставки блоков

Создание определения блока из выбранных объектов

Прежде всего, необходимо вычертить все элементы, которые будут включены в состав создаваемого блока. Определение блока – это информация об имени, базовой точке и объектах, входящих в состав блока. Вхождением же блока является вставленный в чертеж блок, представляющий из себя цельный объект и отображающий данные определения. За вхождение блока отвечает команда "Вставка". При вызове команды "Создать блок" открывается диалоговое окно Определение блока (рис. 2.92), в котором вводим следующие параметры:

- Поле имя создаваемого блока может содержать до 255 русских или английских символов. В составе имени могут содержаться буквы, цифры, пробелы и специальные символы.

- В поле Базовая точка задаются координаты. Базовой называется точка, определенная в блоке и применяемая для его вставки. Она есть в каждом блоке. По умолчанию базовая точка привязывается к точке с координатами 0,0,0. Чаще всего для каждого блока базовая точка определяется индивидуально. Координаты базовой точки вводятся в соответствующие поля X, Y, Z либо нажать кнопку Указать для выбора точки на объекте.

- В поле Объекты расположена кнопка Выбрать объекты, при нажатии на которую временно закрывается окно Определение блока, чтобы выбрать объекты в области построения. Ниже находятся три переключателя, которые определяют, что произойдет с объектами, выбранными в блок после его создания:

- a) Сохранить. Исходные объекты остаются на чертеже и в базе данных чертежа.

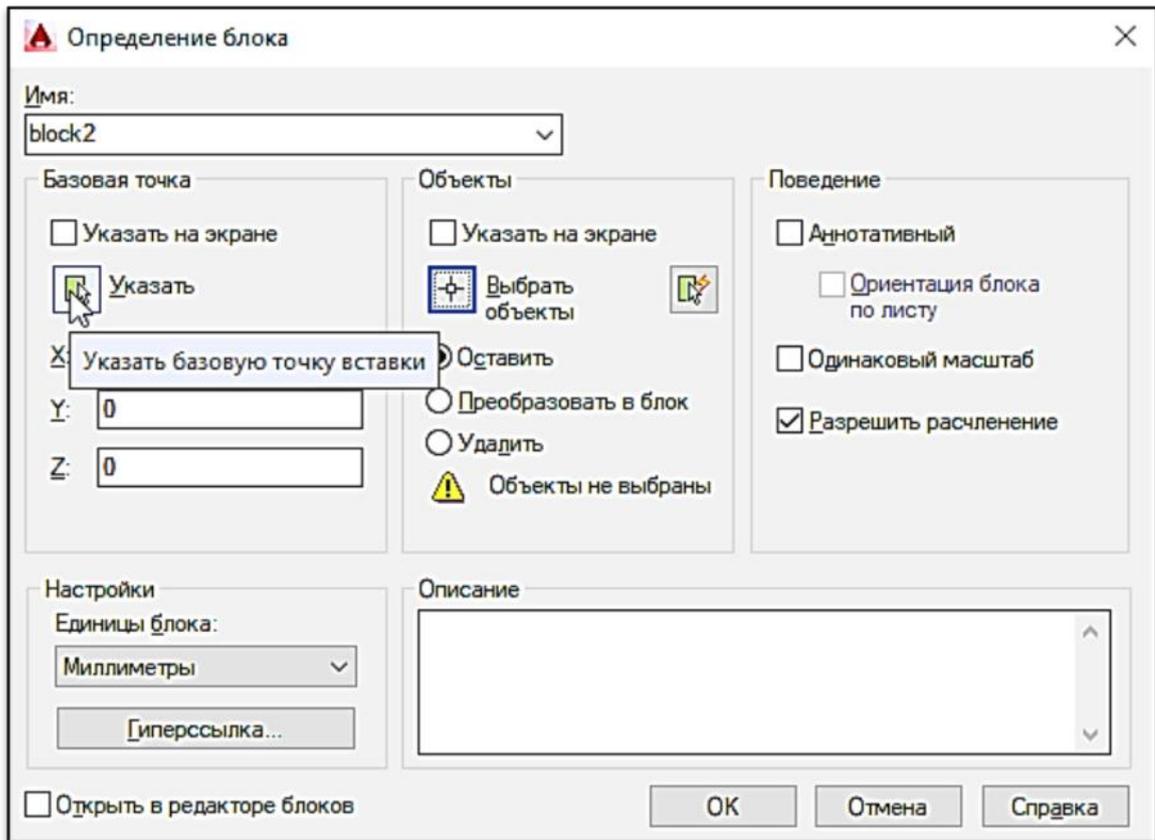


Рисунок 2.92 – Определение блока

б) Преобразовать в блок. Исходные объекты преобразуются в блок и остаются на чертеже и в базе данных чертежа. Этот переключатель установлен по умолчанию.

с) Удалить. Исходные объекты удаляются, если объекты первоначально создавались для вставки в другом месте и после объединения в блок больше не нужны.

– В правом верхнем углу окна, рядом с полем ввода имени блока, расположена графическая область, в которой выводится графический образец создаваемого блока. Это поле появляется после того, когда будут выбраны объекты для блока.

– Поле Поведение содержит флажки Одинаковый масштаб – при вставке блока нельзя задавать разные масштабы по осям, Разрешить расчленение – при вставке блока можно расчленить блок на составляющие.

– В поле Настройки можно определить единицы блока.

– Флажок Открыть в редакторе блоков позволяет редактировать блок сразу после создания.

Удобно организовывать отдельные файлы с типовыми блоками, так как с такими файлами работать легче, чем разыскивать компоненты в других, подчас довольно громоздких чертежах. Таким способом организуются библиотеки типовых деталей или символов, которые могут использовать по мере

необходимости все сотрудники некоторого подразделения или конструкторского бюро. Библиотеки блоков - мощное средство повышения эффективности черчения.

Для сохранения блока в виде файла, необходимо вызвать команду Запись блока на диск, вызвать которую можно либо введя в командной строке WBLOCK либо на панели Определение блока вкладки Вставка. Диалоговое окно Запись блока (рис. 2.93) обеспечивает удобный способ для сохранения указанного определения блока в отдельном файле чертежа.

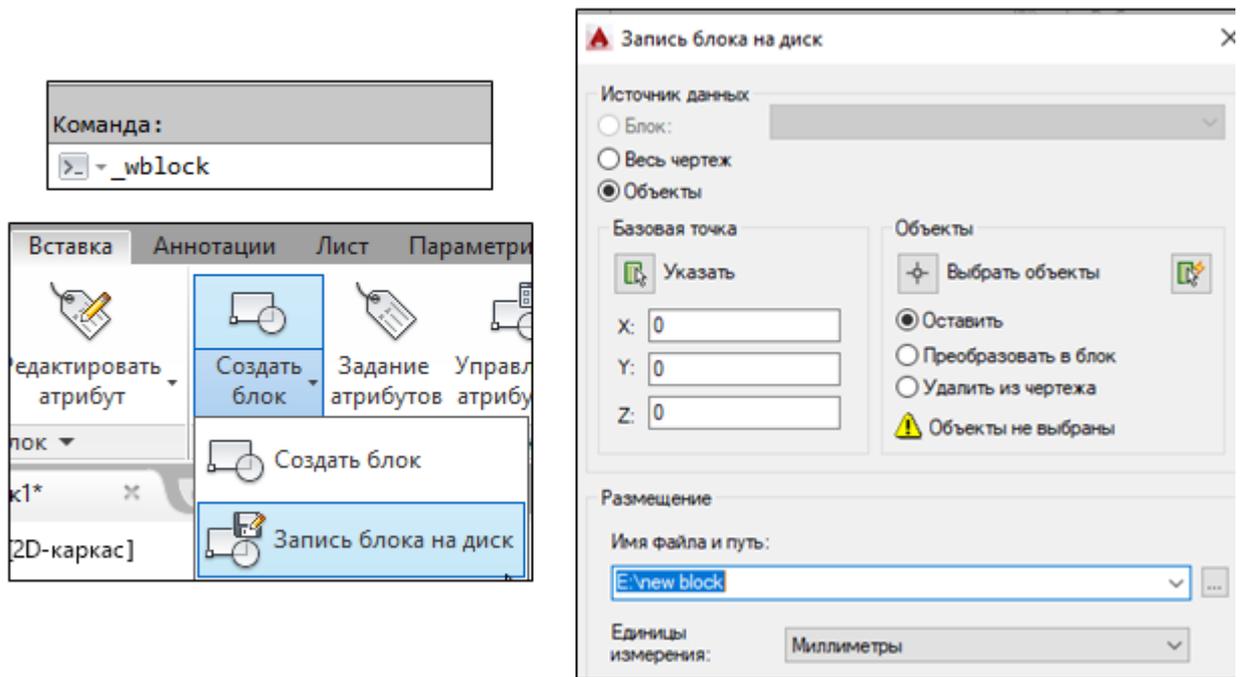


Рисунок 2.93 – Запись блока на диск

Диалоговое окно Запись блока на диск содержит следующие поля:

1. Источник данных;
 - Блок. Если блок на чертеже уже создан и теперь остается только сохранить его в виде отдельного файла, установить этот переключатель и выбрать имя блока в раскрывающемся списке.
 - Весь чертеж. Будет скопирован целый чертеж. В дальнейшем блок можно будет извлечь из него.
2. Объекты. При установке этого переключателя становятся доступными элементы управления, необходимые для выделения объектов и задания базовой точки (базовая точка и объекты).
3. Размещение. Указываются имя и расположение файла.
4. Единицы измерения. Выбрать нужные единицы измерения или установить значение.
5. Завершив перечисленные операции, щелкнуть на кнопке ОК.

При вставке блока базовая точка совмещается с точкой вставки, и все объекты блока занимают места относительно точки вставки.

Вставка блоков и файлов на чертежи

Блоки, как определенные в чертеже, так и записанные в отдельные файлы, вставляются одним и тем же способом (рис. 2.94). После того как выбрано место размещения, можно изменить размер и ориентацию блока. Эта возможность идеальна для библиотеки деталей. Детали могут быть созданы по размеру одного узла, а затем при необходимости масштабированы и переориентированы.

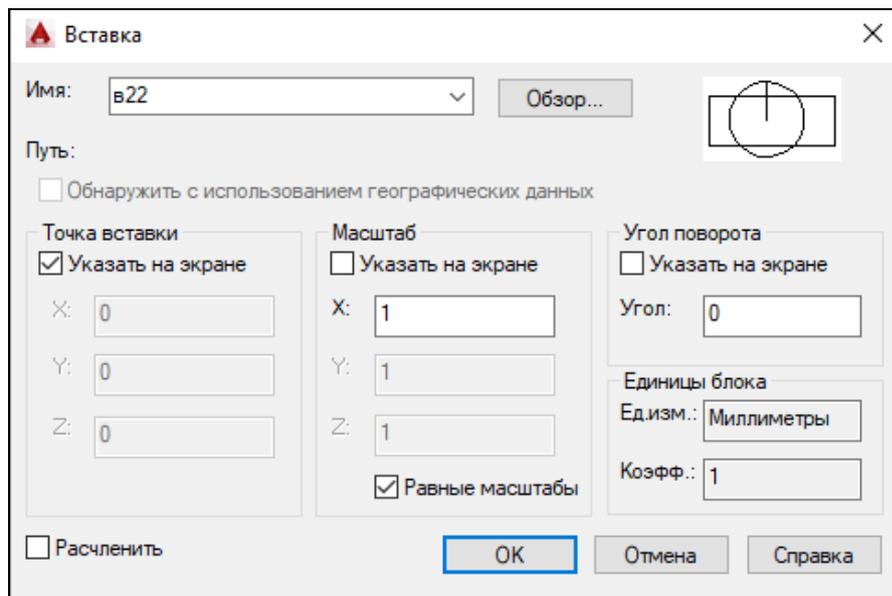


Рисунок 2.94 – Вставка блока

Вставить блок можно несколькими способами:

1. Произвести вставку блока в Автокад из строки меню - вкладка "Вставка" - панель "Блок".
2. Вставить блок Автокад можно из ленты инструментов - вкладка "Главная" - панель "Блок".
3. Прописать наименование команды "Вставить" в командной строке Автокад.
4. Можно вставлять блоки из инструментальных палитр перетаскиванием инструмента работы с блоком в чертеж либо нажатием на инструмент работы с блоком и последующим обозначением точки вставки.

При перетаскивании блока из инструментальной палитры в чертеж он автоматически масштабируется в соответствии с соотношением между единицами изменения, определенными в чертеже, и единицами измерения, определенными в блоке. Например, если в качестве единиц измерения в це-

левом чертеже используются метры, а в блоке - сантиметры, блок вставляется с масштабом 1/100.

5. Блок вставляется из центра управления. Центр управления обеспечивает быстрый наглядный способ перетаскивания блоков в пределах текущего чертежа или из другого чертежа. Дважды щелкните имя нужного блока, чтобы задать положение, угол поворота и масштабные коэффициенты блока.

Команда вставки блока содержит следующие поля:

- Имя. Выбор блока осуществляется из раскрывающегося списка "Имя". В этом списке имеются блоки Автокад определенные на данном чертеже. Если нам требуется вставить внешний блок Автокад, сохраненный в отдельном файле (библиотеке блоков) или целый чертеж в виде блока, то необходимо воспользоваться кнопкой "Обзор" диалогового окна "Вставка блока". Если мы выберем определение блока текущего чертежа, то в правом верхнем углу окна будет отображаться эскиз выбранного блока.

- Точка вставки. Если установить флажок "Указать на экране", то можно указать точку вставки блока Автокад щелчком мыши. Либо указать координаты вставки блока Автокад в соответствующих полях X, Y, Z.

- Масштаб. Масштаб можно определять одинаковым по всем осям или разным. При вводе разных значений масштабов блок будет растягиваться по оси, для которой введен коэффициент больше 2 и сжиматься по оси с коэффициентом меньше 1. Отрицательный коэффициент масштабирования для какой-либо из осей задает зеркальное отражение блока.

- Угол поворота. Задать конкретное значение угла поворота или указать на экране.

- Единицы блока.

- Расчленил - при установке разрушает блок на отдельные элементы, по умолчанию отключен.

Блоки могут состоять из объектов, изначально находившихся на различных слоях и имевших различные цвета, типы линий и веса линий. Хотя вставляемый блок всегда размещается на текущем слое, для каждого объекта, входящего в блок сохраняется информация об исходных слоях, цветах и типах линий. Слой, на котором создаются объекты чертежа, а также определенные параметры свойств влияют на то, сохраняют ли объекты во вставленном блоке исходные свойства или наследуют свойства из текущих параметров слоя, цвета, типа линии или веса линии. При вставке блока создается вхождение блока. Пользователь указывает точку вставки, масштабные коэффициенты и угол поворота.

Блок, созданный на 0 слое, при вставке наследует свойства текущего слоя.

Атрибуты блока

Атрибут – информационный блок, хранящий в себе указанные данные о конкретном объекте или чертеже в целом, таковыми данными могут яв-

ляться: вес, дата изготовления, фирма производитель, стоимость, материал и т.д. Наиболее важным фактором необходимости использования атрибутов, является возможность извлечения данных из чертежа и переноса их в программы, работающие с базами данных, а так же перенос данных непосредственно в таблицы системы AutoCAD.

Для возможности использовать Атрибуты блока, необходимо вначале специальной командой создать их описание. После создания атрибута его надо включить в определение блока в качестве одного из объектов, входящих в блок. После этого при каждой вставке блока в рисунок AutoCAD будет запрашивать значения атрибута, т.е. для каждого вхождения блока можно вводить новый блок.

Большинство команд редактирования объектов воспринимает атрибуты блока и блок как один объект. Однако имеются специальные команды, которые позволяют редактировать атрибуты отдельно от блока как самостоятельный текст.

С блоком можно связывать несколько атрибутов. Только необходимо, чтобы они имели разные имена. Атрибуты могут быть невидимыми на экране. Однако их можно извлекать из рисунка и выводить в отдельный файл.

При создании определения атрибутов задаются параметры формирования текста, который будет запрашиваться при вставке блока с атрибутами в чертеж, и месторасположения этого текста относительно элементов блока.

Для создания описания атрибутов используется команда Атопр, которая вызывается одним из следующих способов:

- Вкладка Вставка – панель Определение блока.
- Командная строка – ввести команду Атопр.

После вызова команды откроется диалоговое окно Определение атрибута, показанное на рис. 2.95, в котором определяются следующие параметры:

1. Режим. В этой зоне расположены переключатели выбора одного из четырех необязательных режимов отображения атрибутов:

- Скрытый. Устанавливает режим, при котором значение атрибута блока, вставленного в рисунок, не должно отображаться на экране, но информация может выводиться в файл.

- Постоянный. Задаёт режим ввода фиксированных значений атрибута для всех вхождений блока. Это значит, что значение постоянного атрибута не будет запрашиваться при вставке блок, а будет выводиться значение, которое определено при описании атрибута.

- Контролируемый. Устанавливает режим, когда AutoCAD проверяет значение вводимого атрибута и сравнивает его со значением, принятым по умолчанию. Данный режим обрабатывается только при задании атрибутов в командной строке.

- Установленный. Автоматически присваивает атрибуту значение по умолчанию.

2. Параметр Атрибут содержит три поля:
 - В поле Тег вводится имя атрибута, которое используется для идентификации каждого вхождения атрибута в блок.
 - В поле Подсказка задается текст, который выводится при вставке блока в командной строке либо в диалоговом окне.
 - Поле Значение по умолчанию используется для ввода значения атрибута, который будет использоваться при вставке блока как значение по умолчанию.
3. Точку вставки можно указать на экране либо указать числовые значения координат X, Y, Z.
4. В зоне Параметры текста задаются:
 - Раскрывающийся список Выравнивание выбирает порядок выравнивания текста относительно точки вставки.
 - Текстовый стиль, в котором будут представлены символы атрибута.
 - Высота символов текста атрибута.
 - Угол поворота текста атрибута относительно оси X.
5. Выровнять по предыдущему атрибуту. Этот параметр становится активным после выбора создания первого атрибута. Следующий атрибут помещается ниже первого атрибута.

Рисунок 2.95 - Определение атрибута

Для создания блока с атрибутами необходимо выполнить следующие действия:

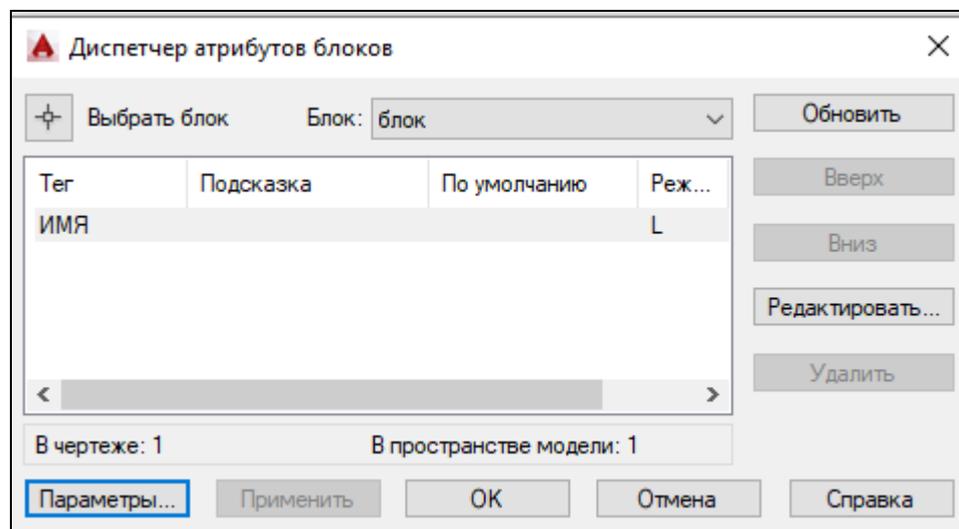
- создать чертеж, который впоследствии будет входить в блок;
- создать определения атрибутов;
- затем из чертежа и атрибутов создать блок.

Можно, изменив значение системной переменной `_ATTDIA` в командной строке, управлять видом ввода атрибутов. Если значение `ATTDIA=1`, то подсказки для ввода атрибутов будут отражаться в диалоговом окне. Если значение `ATTDIA=0`, подсказки будут отражаться в командной строке.

Если в блоке заданы атрибуты и их положение не было заблокировано в окне определения атрибута, то высвечиваются ручки не только у точки вставки блока, но и у точек вставки атрибутов. За собственные ручки атрибуты можно переносить на новое место. Кроме того можно отредактировать значения и свойства атрибутов во вхождениях блоков. Имеется несколько способов редактирования атрибутов блоков:

1. Вкладка Вставка – панель Определение блока – команда Управление атрибутами. Открывается диалоговое окно Диспетчер атрибутов блока (рис.2.96), в котором при нажатии на кнопку Редактировать можно изменить:

- режим и данные на вкладке Атрибут;
- выравнивание, высоту и текстовый стиль на вкладке Параметры текста;
- слой и тип линии на вкладке Свойства.



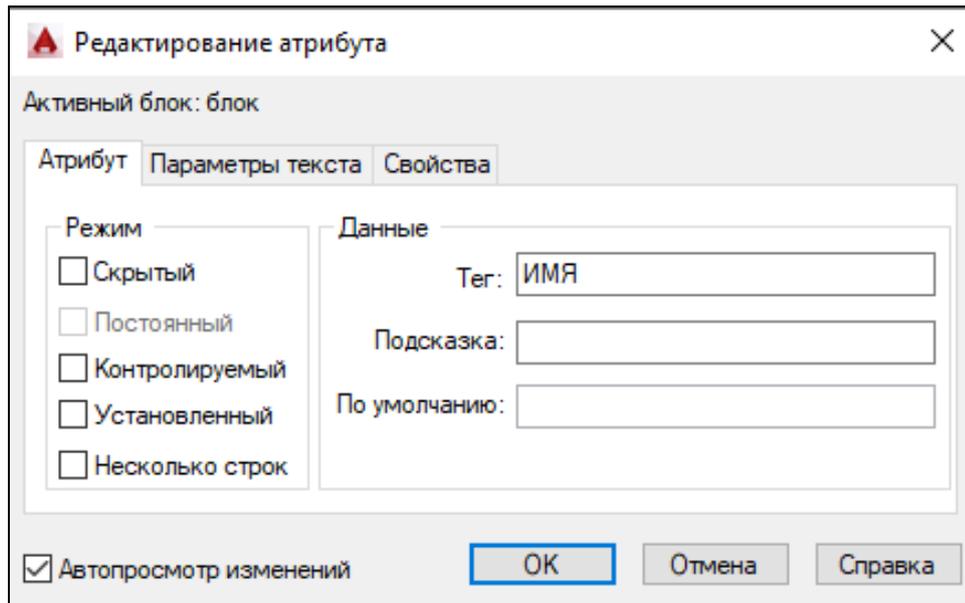


Рисунок 2.96 – Диспетчер атрибутов блоков

– Редактирование значений атрибутов в окне свойств, которое можно вызвать, выделив блок с атрибутами правой кнопкой мыши, и в контекстном меню выбрать команду Свойства.

– Редактирование значений атрибутов в окне редактирования, которое вызывается командой АТРЕД (_ATTEDIT). При этом запрашивается вхождение блока и вызывается диалоговое окно, изображенное на рис. 2.97.

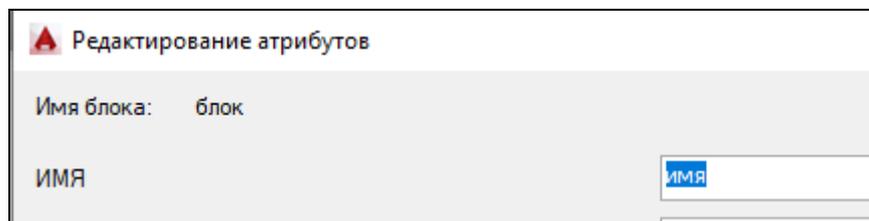


Рисунок 2.97 – Редактирование атрибута из командной строки

Динамические блоки

Блоки бывают двух видов: статические и динамические. Статические блоки в AutoCAD – это группа объектов, объединённых в цельный объект, представляющий из себя блок.

Блоки бывают простые, те которые нельзя изменять (можно лишь перетаскивать с места на место и разбивать на составляющие) и *динамические* (такие блоки после вставки поддаются редактированию: можно растягивать, перемещать отдельные элементы, использовать заранее заданные параметры размеров из таблицы, при этом будут срабатывать массивы элементов).

Для оптимизации работы с блоками, уменьшения их общего количества в библиотеках и файлах, существует понятие динамического блока. Динами-

ческий блок гибкий инструмент позволяющий вставлять блок в различных вариациях (масштабах, углах поворота).

Форматировать обычный блок в динамический возможно в Редакторе блоков. Параметры определяют особые свойства динамического блока, в том числе позиции, расстояния и углы. Динамические блоки можно создать на Палитре вариаций блоков в редакторе блока (рис. 2.98).

Изначально динамический блок является стандартным статическим, а уже потом через редактор к нему применяются добавочные опции и операции. Первая часть создания динамического блока представляет собой, непосредственное определение блока.

Ниже представлен процесс создания динамического блока.

1. Создать блок.
2. Открыть блок в **Редакторе блоков**:
 - Двойной щелчок по блоку.
 - Нажать пиктограмму на панели.
 - Ввести в командной строке **БЛОКРЕД**

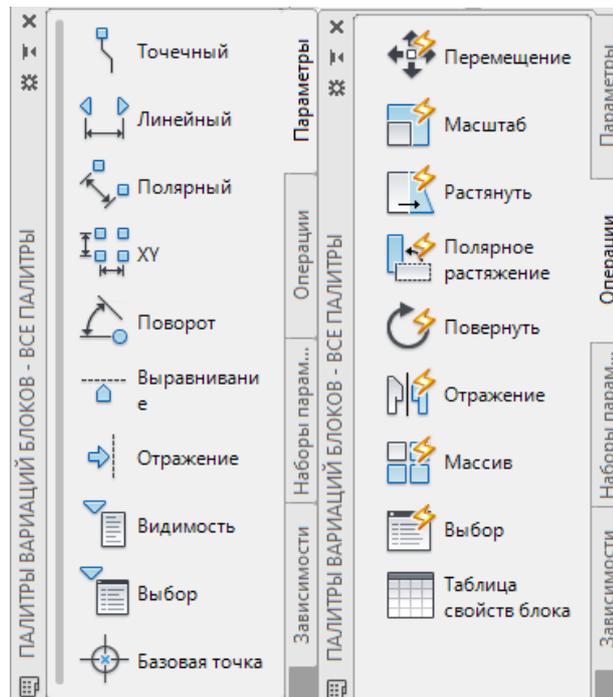


Рисунок 2.98 – Палитры вариаций блоков

3. Добавить параметр и связанное с ним значение.
4. Сохранить определение блока в **Редакторе блоков**.
5. Закрывать **Редактор блоков**.

Панель Палитры вариаций блоков содержит в себе четыре вкладки, имеющие особые назначения.

– **Параметры** – назначает параметры для редактирования блока, создает местоположение ручек. Также можно определить имя параметра.

- Операции – выбор операций, которые будут применяться к параметрам.
- Набор параметров - задается набор параметров и операций, которые с ними связаны.
- Зависимости – геометрические зависимости.

Ниже в таблице 2.1 приведены основные параметры, определяющие динамический блок.

Таблица 2.1 - Основные параметры, определяющие динамический блок

№ п/п	Тип параметра	Количество ручек	Операции для связывания с параметрами.
1	Точка – задается базовая точка	0 или 1	Переместить и растянуть
2	Линейный – задается расстояние между двумя точками.	0,1 или 2	Переместить, растянуть, установить масштаб, копирование.

Окончание табл. 2.1

3	Полярный – задается расстояние между двумя точками и угол наклона между точками.	0,1 или 2	Переместить, Растянуть, установить масштаб, копирование.
4	X Y – задается горизонтальное и вертикальное расстояние между четырьмя точками	От 0 до 4	Переместить, Растянуть, установить масштаб, копирование.
5	Поворот – задается центр поворота	0 или 1	Поворот
6	Отражение - задаются две точки оси отражения	0 или 1	Отразить
7	Выравнивание – задаются точки для выравнивания блока относительно других объектов чертежа.	1	Содержится внутри параметра
8	Видимость – задается одна точка условная точка	0 или 1	Нет
9	Параметр поиска – задается перечень свойств	1	Выбор объекта

	для поиска блока.		
10	Базовая точка.		Нет

В самом редакторе можно добавить или изменить геометрию блока. Для этого необходимо выполнить одно из следующих действий.

– Добавить один или несколько параметров с вкладками **Параметры** в Палитре Разработки Блоков.

– Добавить одну или несколько операций с вкладки **Операции**. Далее следуем подсказкам в командной строке.

Потом необходимо сохранить само описание блока и выйти из редактора.

После того как был создан и сохранен динамический блок, для его вставки в новый чертеж необходимо выполнить следующие действия:

1. Использовать команду **ВСТАВИТЬ**, для вставки динамического блока в новый чертеж.

2. При выборе динамического блока, будут отображены специальные маркеры (захваты) показывающие параметры доступные для изменения блока.

3. Как правило, захваты перемещаются, тем самым изменяя параметры блока. Некоторые динамические параметры блока включают в себя выбор видимости или опции поиска из выпадающего списка.

Пример 4.3. Рассмотрим создание динамического блока Окно с изменяющейся высотой. Для этого выполним следующие действия:

1. Создадим простейший блок Окно в виде прямоугольника 150x180. Базовая точка – левый верхний угол.

2. Щелкнув по нему два раза, в открывшемся окне Редактора блока выбираем блок Окно.

3. На палитре вариаций открываем вкладку Параметры. При помощи Линейного параметра проставляем на блоке размер (рис. 2.99). При установке появляется желтый квадрат с восклицательным знаком.

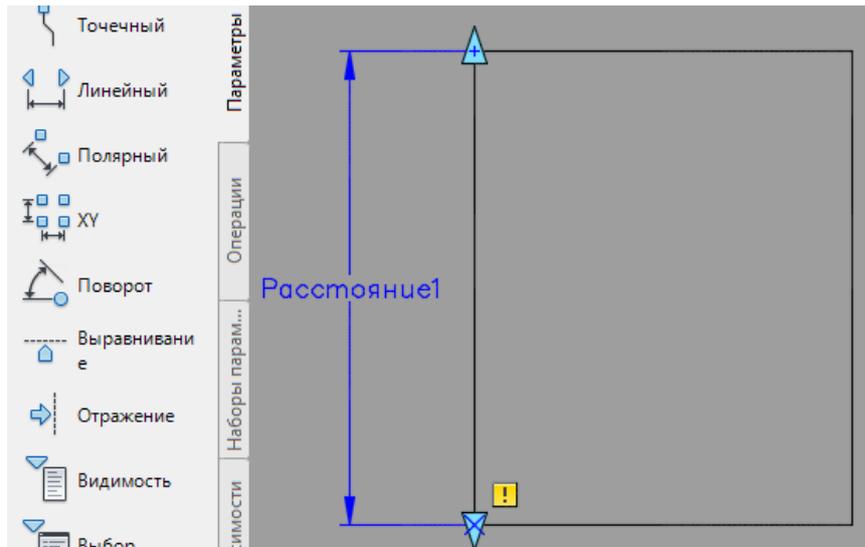


Рисунок 2.99 – Создание линейного параметра

4. К проставленному параметру применяем операцию Растянуть (рис. 2.100). В командной строке появляются следующие запросы:
- Выберите параметр: *указываем мышкой синий параметр*
 - Укажите точку параметра, которую необходимо связать с операцией: *указываем левую нижнюю точку*
 - Укажите первый угол рамки растягивания или [СМН-угол]: *указываем Секрамка Лассо синей пунктирной линией рамку. Указать Противоположный угол.*
 - Укажите объекты для растяжения: *выбираем объекты Секрамкой Лассо зеленого цвета внутри рамки растягивания. Те объекты, которые полностью попали в рамку, просто перемещаются, остальные – растягиваются.*
 - Выберите объекты: найдено: 3. *Нажать Ввод.*

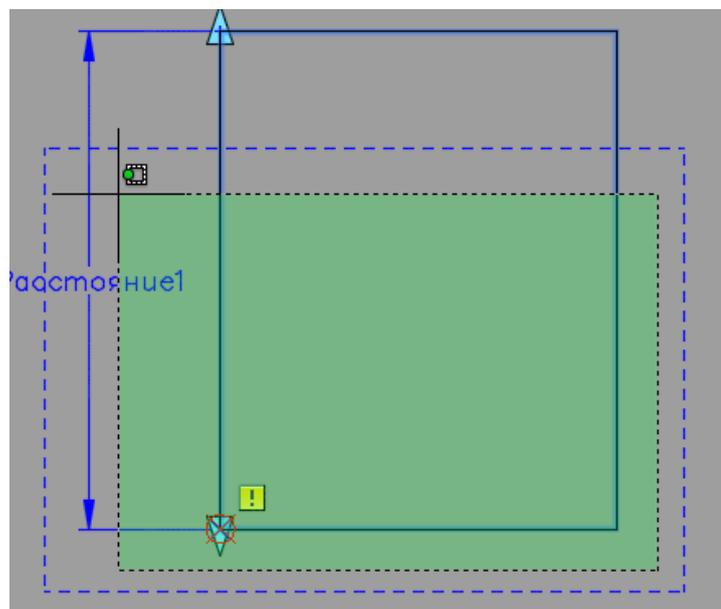


Рисунок 2.100 – Применение операции Растянуть



5. Возле параметра появится значок примененной операции

6. Для завершения нажимаем на кнопку Закреть Редактор блока и выбираем Сохранить изменения.

При вставке блока на рис. 2.101а отражается точка вставки (синий квадрат) и ручка растяжения (голубой треугольник).

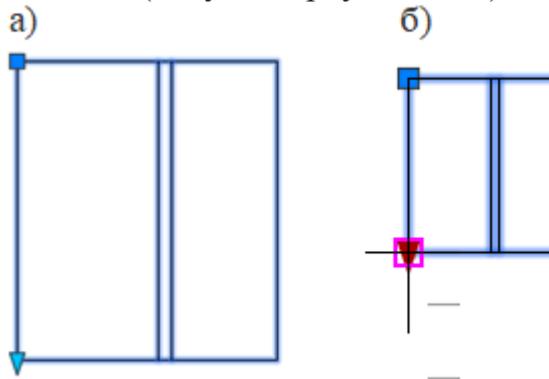


Рисунок 2.101 – Вставленный блок Окно

а – блок растягивается с помощью ручки; б – блок растягивается по указанным значениям в списке

Теперь можно задать конкретные параметры изменение по длине. Для этого нужно выделить наш блок, кликнуть правой кнопкой и нажать Редактор блока.

Далее, в Редакторе динамических блоков, нужно выделить линейный параметр “расстояние 1” и кликнуть на экране правой кнопкой мышки и выбрать пункт “свойства” внизу списка. Откроется окно Свойства. В пункте “Набор значений”, в поле "Тип расст" выбрать тип “список” (рис. 2.102).

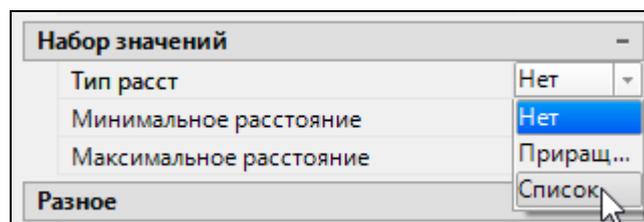


Рисунок 2.102 – Вызов списка значений

В поле Список значений расстояния необходимо щелкнуть на желтый квадратик справа, при этом откроется окно Добавление значения расстояния (рис. 2.103), где можно задать конкретные значения размеров окна.

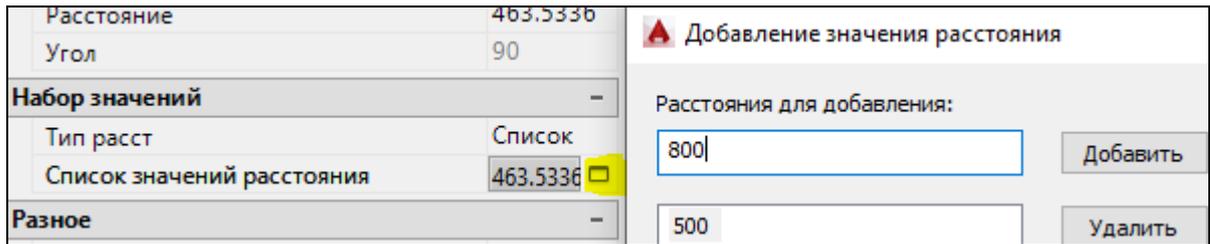


Рисунок 2.103 – Добавление списка значений расстояния

Теперь при вставке блока Окно при нажатии на ручку растяжения появляются "черточки", идентичные заданным расстояниям в списке, представленные на рис. 2.101б.

Внешние ссылки

Внешние ссылки AutoCAD – это возможность в свой чертеж вставить чертежи из других подразделений, причем, без копирования всех примитивов. Содержимое дополнительного файла-ссылки подгружается в открытый файл, но не хранится в нем. Это дает ряд преимуществ:

1. Размер файла существенно разгружается, т.к. чертеж хранит только те объекты и примитивы, которые начертили именно вы.

2. Любые изменения, которые будут сделаны в чертежах, вставленных как ссылки Автокада, будут отображаться в открытом чертеже. Сразу же после внесения изменений и сохранения файла-ссылки будет получено уведомление.

3. Внешние ссылки Автокада – это возможность организовать грамотную разработку сложного проекта, в котором каждое подразделение (отдельный человек) отвечает за свою работу: отдел архитектуры за разработку планировки этажа, проектный отдел - за инженерные системы и сети и т.д.

В качестве внешней ссылки могут также выступать следующие файлы: DWF, PDF, DGN, DWG И ИЗОБРАЖЕНИЯ.

Внедрение внешних ссылок в Автокад осуществляется через команду Присоединить на панели Ссылка вкладки Вставка (рис. 2.104).

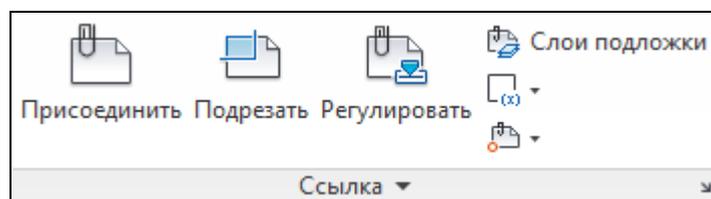


Рисунок 2.104 – Панель Ссылка

После выбора команды нужно найти файл, который в дальнейшем и будет выступать как внешняя ссылка, и отображаться в окне вставки ссылки. После этого откроется диалоговое окно вставки внешней ссылки (рис. 2.105).

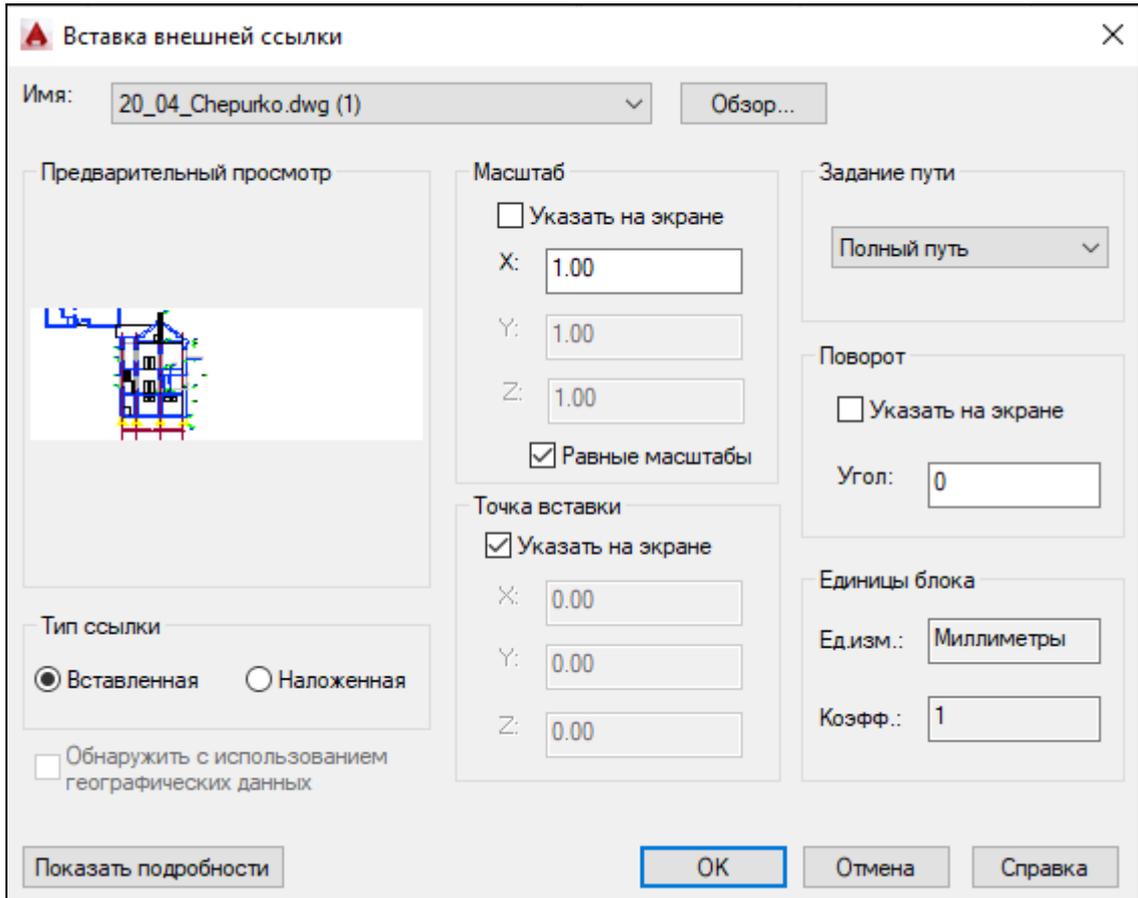


Рисунок 2.105 – Параметры вставки внешней ссылки AutoCAD

Здесь можно настроить масштаб, угол поворота и указать точку вставки, а также единицы измерения внедряемого чертежа. Существует два типа ссылок:

1. Вставленная – внешняя ссылка сама может иметь внешние ссылки, причем неограниченное число раз и с разным масштабным коэффициентом. Графически это все отображается на всех чертежах.

2. Наложённая – при наложении внешней ссылки в открытом файле будут отображать графически все объекты внедряемого файла, а при дальнейшем использовании уже вашего чертежа, как внешней ссылки – нет.

Внизу в правом углу появляется красный значок уведомлений внешних ссылок. При нажатии на него открывается Диспетчер внешних ссылок (рис.2.106).

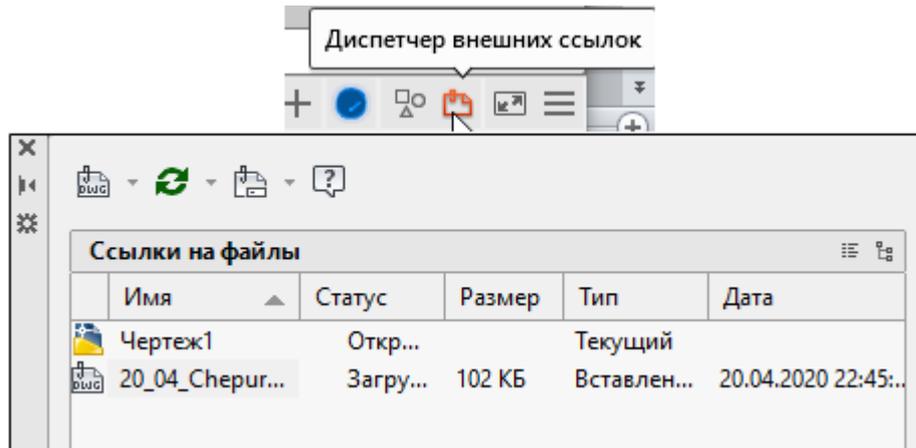


Рисунок 2.106 – Диспетчер внешних ссылок

При нажатии правой кнопкой мышки по названию внешней ссылки в контекстном меню появляются варианты возможных действий с ссылкой:

- Открыть
- Вставить
- Выгрузить
- Обновить
- Удалить
- Внедрить.

Для подрезки внешней ссылки можно выбрать команду Подрезать на панели Ссылка (рис. 4.34). Подрезать внешнюю ссылку можно, используя существующий замкнутый объект на чертеже, а также многоугольным или прямоугольным контуром, который указывается в команде подрезки.

В командной строке появляются запросы:

Команда: `_clip` Выберите объект для подрезки: найдено: указываем внешнюю ссылку.

Задайте параметр подрезки

[Вкл/Откл/Плоскости/Удалить/полиЛиния/Новый контур] <Новый>:

Внешний режим - Будут скрыты объекты вне контура.

Укажите контур подрезки или выберите параметр инверсии:

[Выбрать полилинию/Многоугольный/Прямоугольный/Обратная подрезка] <Прямоугольный>:

Первый угол: Противоположный угол:

Результат подрезки внешней ссылки прямоугольником представлен на рис. 2.107.

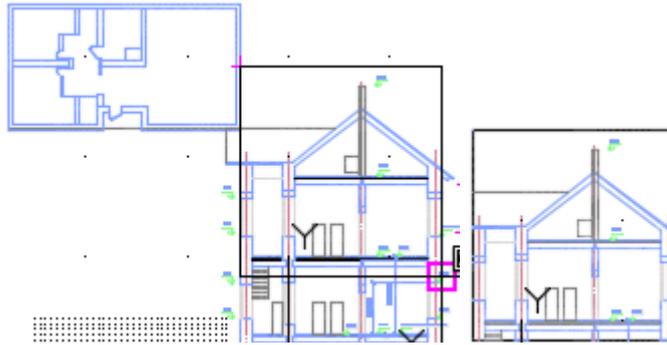


Рисунок 2.107 – Подрезка внешней ссылки прямоугольником

2.3.4 Видовые экраны в пространстве модели и в пространстве листа

Создавать объекты на чертеже можно в двух различных рабочих средах. Эти среды получили название Пространство Модели и пространство Листа. Переключение между данными пространствами осуществляется с помощью вкладок, расположенных в нижней части области рисования: вкладка "Модель" и одна или несколько вкладок "Лист".

Для разработки и отладки чертежей обычно используется пространство Модели, которое становится активным при запуске программы и является безграничной областью. Пространство Листа применяется для вывода чертежа на печать.

В пространстве модели можно рисовать модель в масштабе 1:1, и задавать значение одной единицы рисунка – один миллиметр, сантиметр, метр.

В пространстве листа одна единица соответствует расстоянию на распечатанном листе. Единицы задаются в миллиметрах или дюймах, в зависимости от настроек принтера.

Графическую область на вкладке "Модель" можно разбить на несколько неперекрывающихся прямоугольных областей, называемых видовыми экранами пространства модели. В больших или сложных чертежах использование различных видов позволяет избежать частого выполнения операций зумирования и панорамирования. Ошибки, незаметные на одном виде, могут быть обнаружены на другом.

Видовые экраны на вкладке "Модель" полностью занимают графическую область, не перекрывая друг друга. Во время редактирования рисунка на одном видовом экране тут же происходит обновление изображений на других видовых экранах.

Видовые экраны пространства модели можно изменять, разбивая их на части, и объединяя друг с другом. Объединяемые видовые экраны должны иметь общую границу одной длины.

Различные настройки графической программы AutoCAD, например, шаг сетки, для каждого из видовых окон можно устанавливать отдельно.

Видовые экраны пространства Модели устанавливаются следующим образом: меню Вид – Видовые экраны модели (рис. 2.108). Есть возможность задать горизонтальное или вертикальное расположение видовых экранов.

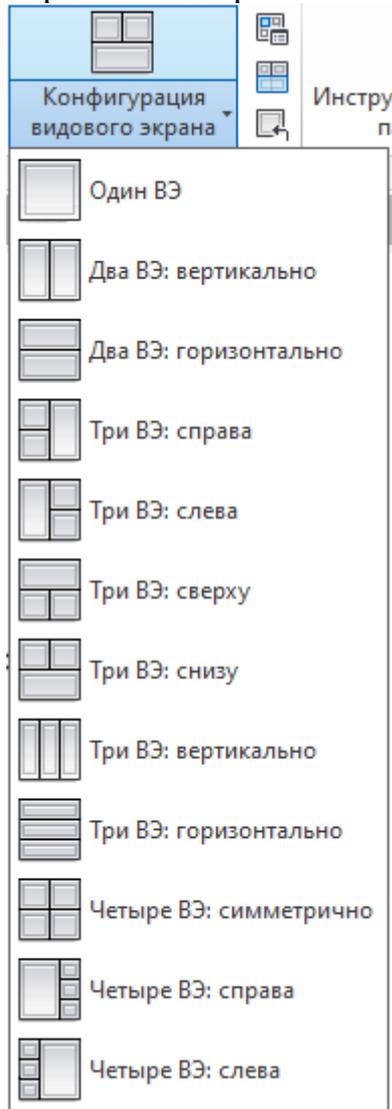


Рисунок 2.108 – Видовые экраны модели

Элементы управления видовым экраном пространства модели

Средства управления видовым экраном отображаются в левом верхнем углу каждого видового экрана и позволяют легко изменять виды, визуальные стили и другие параметры **[+] [Сверху] [2D каркас]**.

Метки отображают текущие параметры видового экрана.

Можно нажать левую кнопку мыши внутри каждой из трех заключенных в скобки областей для изменения настроек:

- Нажатие на кнопку "+" или "-" приводит к отображению параметров, позволяющих развернуть видовой экран, изменить его конфигурацию или управлять отображением инструментов навигации.

- Нажатие на кнопку "Сверху" позволяет выбрать один из нескольких стандартных и пользовательских видов.
- Нажатие на кнопку "2D каркас" позволяет выбрать один из нескольких визуальных стилей. Большинство остальных визуальных стилей используются для 3D-визуализации.

Чтобы сделать активным тот или иной видовой экран, достаточно дважды щелкнуть внутри него ЛКМ. Он сразу выделится синим прямоугольником.

На неперекрывающихся видовых экранах в пространстве модели допускается:

- Производить панорамирование и зумирование, настраивать режимы сетки, шаговой привязки и изображения пиктограммы ПСК;
- Задавать систему координат и восстанавливать виды для каждого отдельного видового экрана;
- Переключаться с одного видового экрана на другой в ходе выполнения команд рисования;
- Сохранять именованную конфигурацию видовых экранов в пространстве модели или применять ее в пространстве листа.

При работе с трехмерными моделями обычно требуется назначение различных систем координат для отдельных видовых экранов.

На плоттер неперекрывающиеся видовые экраны выводятся только поодиночке.

Ранее сохраненные расположения видовых экранов можно восстановить по имени.

Именованный вид – это фрагмент чертежа, который сохраняется под собственным именем и может быть установлен на экране в любой момент времени над моделью или при открытии файла. Но сначала нужно создать в рисунке именованные виды, присвоив им уникальные имена.

Новый видовой экран можно создать на панели Видовые экраны модели меню Вид. На этой панели находится кнопка Именованные виды, нажав на которую, открывается диалоговое окно Видовые экраны (рис. 2.109), содержащее две вкладки.

– Вкладка Именованные ВЭ содержит список всех сохраненных конфигураций видовых экранов в модели в чертеже. Задать именованный ВЭ можно на вкладке Новые ВЭ экраны, введя в поле Новое имя название для видового экрана.

– Вкладка "Новые видовые экраны" - Пространство модели (Диалоговое окно "Видовые экраны") содержит следующие поля:

Новое имя

Имя новой конфигурации видовых экранов в пространстве модели. Если имя не вводится, конфигурация видовых экранов применяется, но не со-

храняется. Если конфигурация видовых экранов не сохранена, ее нельзя использовать в пространстве листа.

Стандартные конфигурации

Список стандартных конфигураций видовых экранов, включая конфигурацию "ТЕКУЩАЯ".

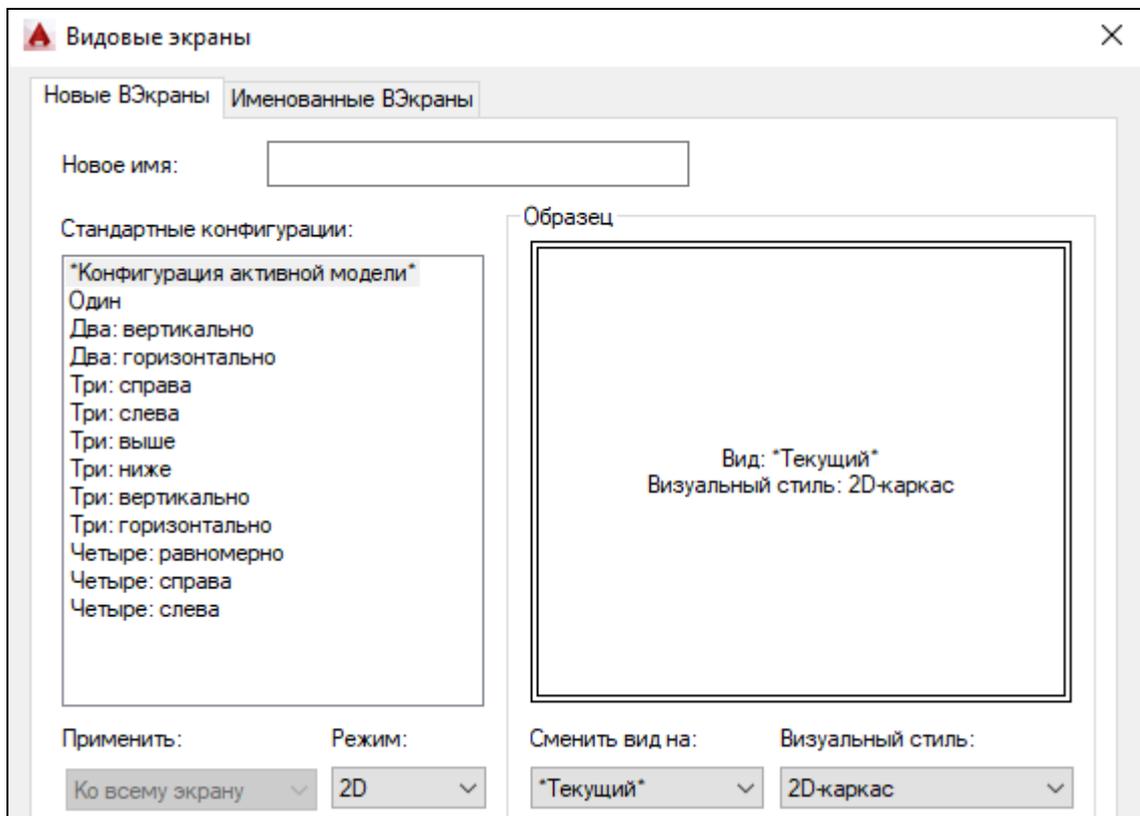


Рисунок 2.109 – Установка видовых экранов модели

Образец

Предварительный просмотр выбранной конфигурации видовых экранов, включая информацию о виде, назначенному каждому из них.

Применить

Применение конфигурации видовых экранов ко всей области рисования или только к текущему видовому экрану.

- *Экран:* Применяет конфигурацию видовых экранов ко всей области рисования вкладки "Модель".
- *Текущий ВЭкран:* Применяет конфигурацию видовых экранов только к текущему видовому экрану.

Режим

Выбор 2D- или 3D-режима. Если выбран 2D-режим, на всех видовых экранах создаваемой конфигурации по умолчанию устанавливается такой же вид, как и у текущего активного видового экрана. Если выбран 3D-режим, на

создаваемых видовых экранов сразу же будут установлены стандартные виды модели.

Сменить вид на

Список возможных видов для создаваемых видовых экранов. В списке выводятся имеющиеся в чертеже именованные виды, а если установлен 3D-режим, то еще и стандартные виды модели. Для смены вида на каком-либо видовом экране нужно выделить его в группе "Образец" и выбрать нужный вид из списка.

Визуальный стиль

Визуальный стиль, применяемый к видовому экрану. Отображаются все доступные визуальные стили.

Пространство листа облегчает получение твердых копий рисунков и чертежей, разработанных автоматизированным путем. Если бы пространство листа не использовалось, пришлось бы загромождать пространство модели графической информацией, необходимой лишь для формирования чертежных листов. Ведь такие элементы, как рамка чертежного листа, основная надпись и другая графическая и текстовая информация, не имеют отношения к реальной модели и требуются только в распечатке.

Листом называется компонент среды AutoCAD, имитирующий лист бумаги и хранящий в себе набор установок, используемых при выводе на плоттер. На листе можно размещать видовые экраны, а также строить геометрические объекты (например, элементы основной надписи). Рисунок может содержать несколько листов с разными видами модели; для каждого листа автономно задаются значения масштаба печати и размеров сторон.

Изображение листа выглядит на экране точно так же, как вычерченный на плоттере лист (рис. 2.110). Белый прямоугольник с тенью соответствует на экране формату бумаги, на который настроено устройство печати. Границы области печати обозначены штриховыми линиями. Синий прямоугольник внутри – видовой экран. Пиктограмма ПСК на листе принимает вид треугольника.

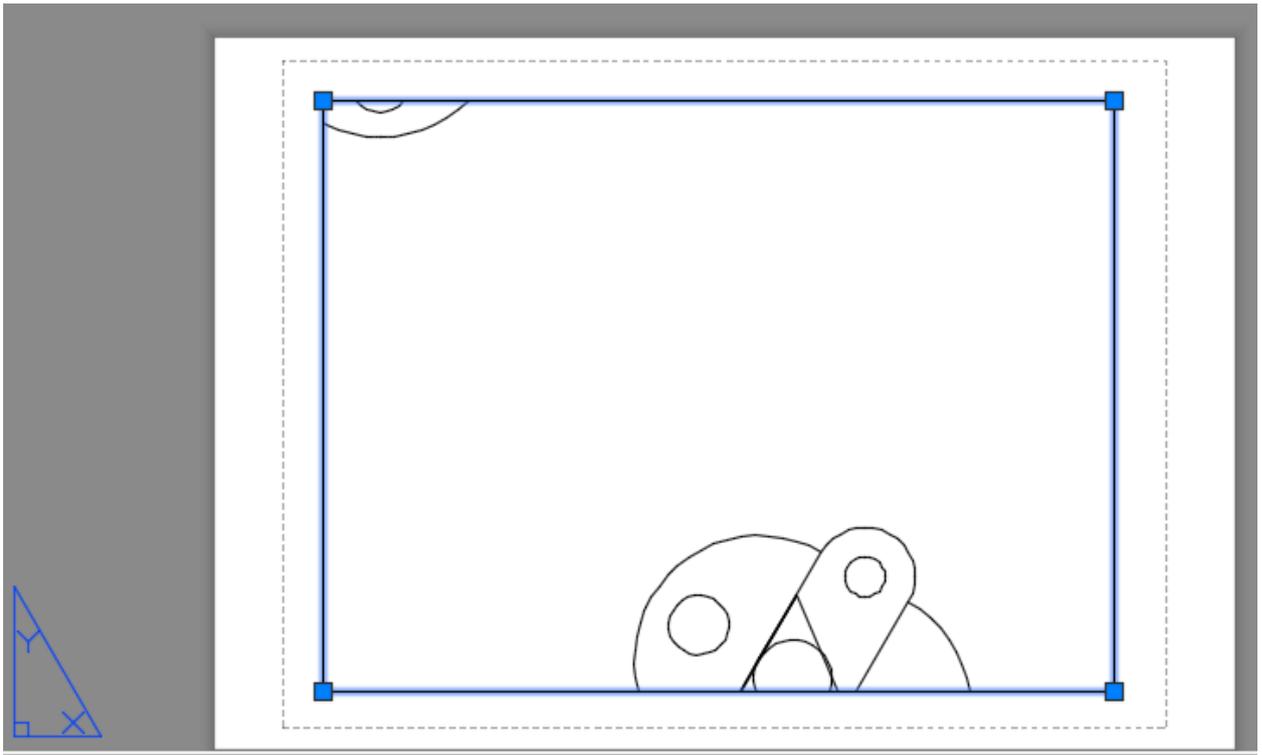


Рисунок 2.110 – Видовой экран в пространстве листа

Для подготовки чертежа к печати нужно переключиться в пространство Листа. Для этого нужно щелкнуть ЛКМ по вкладке Лист1, находящейся внизу слева графического окна. Иными словами, лист - это прототип будущего чертежа.

Особенностью чертежной области, предназначенной для вывода на печать, является то, что 1 мм пространства листа теоретически должен быть равен 1 мм на бумаге печатающего устройства. На практике получается, что почти все малоформатные принтеры имеют поля печати. Поэтому при выводе на печать имеется некоторая погрешность (уменьшение). Обычно она не превышает 5-8%.

На Листе осуществляется компоновка видов чертежа с помощью видовых окон. Видовое окно – это участок экрана графической программы AutoCAD, на котором можно представить часть чертежа, его отдельный вид. Видовые окна следует создавать на отдельном слое, который перед печатью нужно отключить, чтобы при выводе на печать не печатались границы видовых окон, или же использовать для них слой Defpoints, который не печатается. Для видовых окон можно задавать различные масштабы, чтобы виды чертежа на листе располагались наилучшим образом.

Видовой экран (ВЭ) пространства листа - это не что иное, как примитив. Каждый видовой экран листа представляет собой своего рода кадр изображения в пространстве модели, содержащий вид, в котором отображается модель в заданном масштабе и ориентации. Таким образом, через видовой

экран мы попадаем в пространство модели. Если дважды щелкнуть ЛКМ внутри видового экрана, то он станет активным и у нас появится доступ к пространству Модели, где находятся чертежи. Вращая колесико мыши, мы можем изменять масштаб, тем самым отдаляться или приближаться к объектам чертежа.

Чтобы выйти из видового экрана, достаточно дважды щелкнуть ЛКМ по листу за границей ВЭ.

В пространстве листа может создаваться один видовой экран, занимающий весь лист, или несколько видовых экранов. Если его выделить, то появятся четыре ручки для редактирования.

Создание нового ВЭ в пространстве листа

Чтобы создать новый видовой экран, можно перейти на вкладку Лист - панель «Видовые экраны листа»

Существует несколько вариантов создания ВЭ: прямоугольной формы, многоугольной или вообще произвольной формы (рис.2.111).

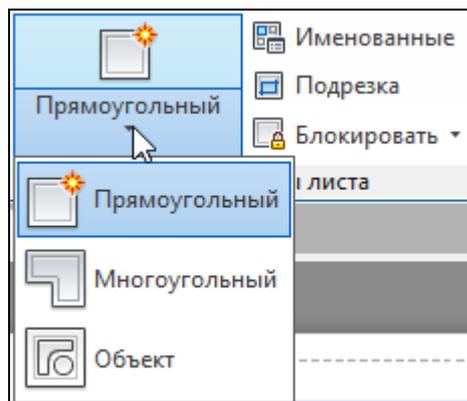


Рисунок 2.111 - ВЭ в пространстве Листа

Для создания нескольких видовых экранов используем кнопку Именованные на панели видовые экраны листа вкладки Лист. Открывается диалоговое окно "Видовые экраны" (рис. 2.112), содержащие следующие поля.

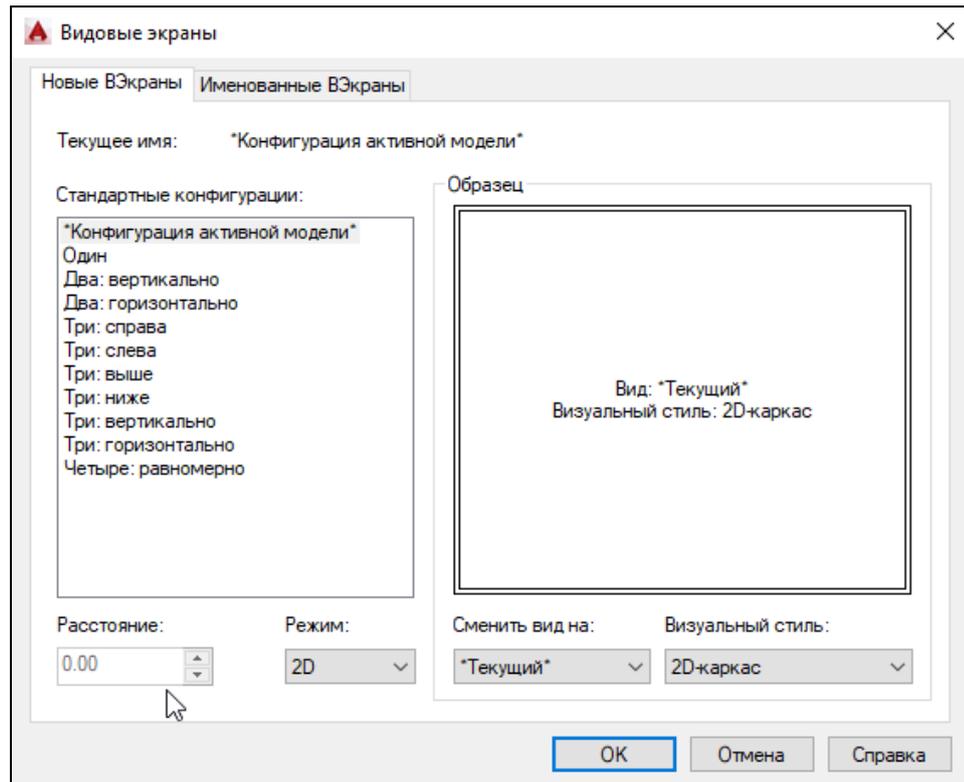


Рисунок 2.112 - Диалоговое окно "Видовые экраны"

Стандартные конфигурации

Создание неперекрывающихся видовых экранов на основе стандартных конфигураций видовых экранов.

Образец

Предварительный просмотр выбранной конфигурации видовых экранов, включая информацию о виде, назначенному каждому из них.

Расстояние

Расстояние между создаваемыми плавающими видовыми экранами.

Режим

Выбор 2D- или 3D-режима. Если выбран 2D-режим, на всех видовых экранах создаваемой конфигурации по умолчанию устанавливается такой же вид, как и у текущего активного видового экрана. Если выбран 3D-режим, на создаваемых видовых экранах сразу же будут установлены стандартные виды модели.

Сменить вид на

Список возможных видов для создаваемых видовых экранов. В списке выводятся имеющиеся в чертеже именованные виды, а если установлен 3D-режим, то еще и стандартные виды модели.

Создавать и сохранять именованные конфигурации плавающих видовых экранов нельзя.

Автоматическое замораживание и размораживание слоев на новых видовых экранах

При создании новых видовых экранов листа можно автоматически замораживать на них определенные слои.

1. Перейти на вкладку разметки листа.
2. Выбрать вкладку "Главная" > панель "Слои" > "Свойства слоя".
3. В окне "Диспетчер свойств слоев" выделить слои, которые необходимо заморозить или разморозить на новых видовых экранах.

Например, можно скрыть размеры путем замораживания слоя РАЗМЕРЫ для всех новых видовых экранов. Если же на одном из видовых экранов размеры должны присутствовать, то для этого видового экрана можно переопределить установленные ранее настройки. Включение или отключение замораживания слоев для новых видовых экранов не изменяет видимость слоев на уже имеющихся видовых экранах.

4. В столбце "Замороженный на новых ВЭ" нажать значок для изменения состояния слоя.

На листе может создаваться один видовой экран, занимающий весь лист, или несколько видовых экранов. Размеры, свойства, масштаб и расположение видовых экранов могут меняться.

Подрезка видового экрана

Для подрезки ВЭ можно использовать кнопку Подрезка (рис.4.41) на панели видовые экраны Листа.

В командной строке появляется запрос:

Команда: `_vrslip`

Выберите видовой экран для подрезки:

Выберите подрезающий объект или [Многоугольник] <Многоугольник>:

- Параметр "Объект" позволяет преобразовать в видовой экран замкнутый объект из пространства листа, например, окружность или полилинию. Объект описывает границу вновь создаваемого видового экрана; в результате получается видовой экран неправильной формы.
- Используя параметр "Многоугольный" можно создать непрямоугольный видовой экран листа по заданным точкам. Последовательность действий такая же, как и при создании полилинии.

Для подавления отображения границ видового экрана листа можно просто отключить слой видового экрана неправильной формы, вместо того, чтобы замораживать его. При замораживании слоя непрямоугольного видового экрана не обеспечивается корректная обрезка последнего.

**3 РАЗДЕЛ 2. ПРАКТИЧЕСКИЙ РАЗДЕЛ. МЕТОДИЧЕСКИЕ
УКАЗАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ ПО ДИСЦИПЛИНЕ
"РЕШЕНИЕ ТЕХНОЛОГИЧЕСКИХ ЗАДАЧ НА ЭВМ"**

3.1 Лабораторная работа №1 "ЗНАКОМСТВО С ИНТЕРФЕЙСОМ СИСТЕМЫ ПРОГРАММИРОВАНИЯ VISUAL BASIC"

Цель работы: познакомиться с интерфейсом системы программирования VISUAL BASIC, рассмотреть основные компоненты среды программирования. Получить навыки работы с объектами метка, текстовое окно и рамка.

Методика выполнения работы

Перед выполнением первой работы создайте на диске свой каталог и в дальнейшем сохраняйте свои проекты в нем.

- При сохранении проекта обязательно давайте уникальные имена файлам (не соглашайтесь с именами, предложенными по умолчанию), расширение не менять.

- Имена файла формы и файла проекта могут совпадать.

- Практические работы выполняйте в указанной последовательности.

- Имя объекта (**Name**) должно быть изменено (если это требуется) до создания программного кода для этого объекта.

- В окне программного кода не удалять и не менять строки **Private Sub** и **End Sub**.

- Самая распространенная ошибка, возникающая при выполнении приложения «**Run-time error '424'**» означает, что неверно указано имя объекта в программном коде. В этом случае программу следует остановить, исправить ошибку и загрузить заново.

При загрузке Visual Basic открывается следующий экран приложения, содержащий компоненты:

7. Строка меню

Меню содержит команды, используемые при работе Visual Basic . кроме стандартных меню **File, Edit, View, Window, Help**, здесь расположены меню, обеспечивающие доступ к функциям программирования, например, **Project, Format, Debug**

8. Панель инструментов (Toolbars Standard)

Предоставляет быстрый доступ к наиболее часто используемым командам среды программирования

9. Форма (**Form**)– окно будущего приложения

10.Панель элементов управления (**Toolbox**)

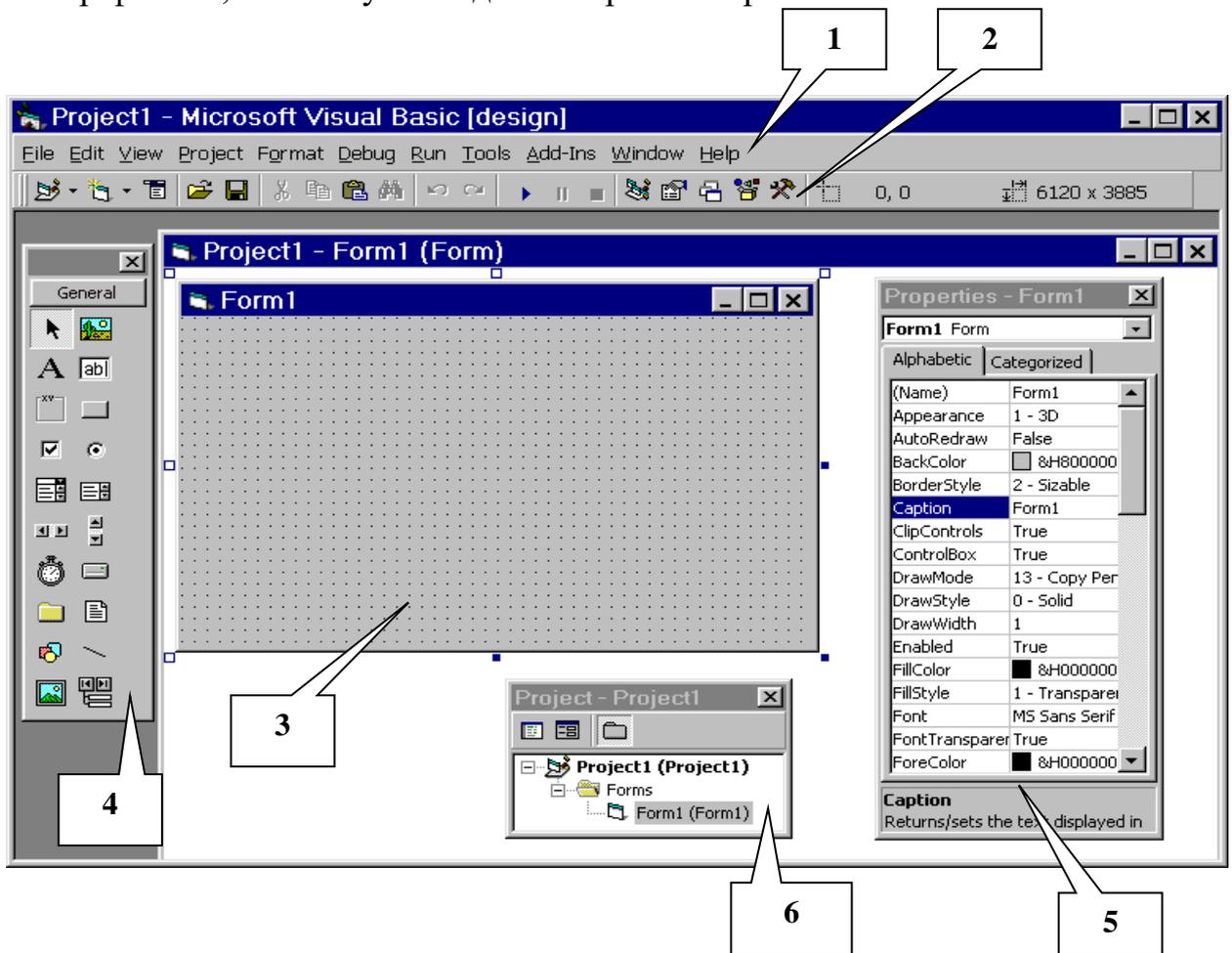
Панель обеспечивает проектировщика набором инструментов, необходимых во время разработки для размещения элементов управления на форме

11.Окно свойств (**Properties**)

Перечисляет установленные свойства для выбранной формы или элемента управления.

12.Окно проводника проекта (**Project Explorer**).

Представляет список форм и модулей текущего проекта. Проект – это набор файлов, используемых для построения приложения.



Любой объект можно связать с набором процедур, исполняемых в строго определенных моменты. Процедура (*procedure*) – это группа операторов языка. Исполняется процедура – исполняются ее операторы. Так или иначе, весь составленный вами исполняемый код обязательно помещается в какую-нибудь процедуру.

Процедура, присвоенная объекту, связана с определенным событием и поэтому называется *процедурой обработки события*. С одним объектом могут быть связаны несколько событий.

Имя процедуры обработки события для элемента управления составляется из имени элемента управления (Name), знака подчеркивания (_) и имени события.

Для открытия окна кода процедуры существует три способа:

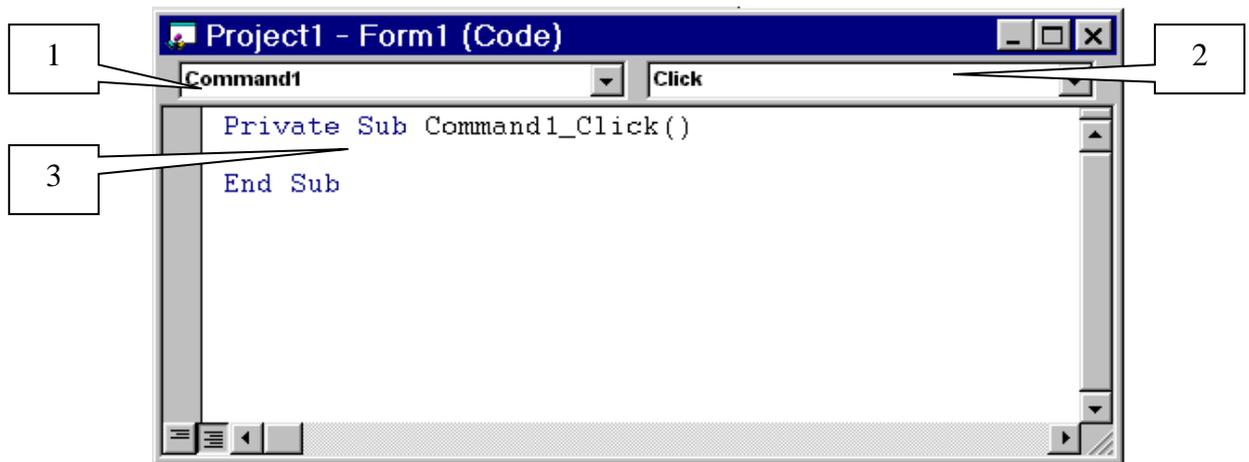
4. двойной щелчок по объекту
5. клавиша F7
6. в меню выбрать View / Code

Процедуры по умолчанию не делают ничего; они состоят лишь из объявления процедуры (*Sub*) и оператора, помечающего конец процедуры (*End Sub*).

Окно кода процедуры включает:

1. список объектов формы
2. список событий объекта
3. процедура

Программный код вводится между строками **Private Sub** и **End Sub**



Задание 1

1. Установите на форме следующие объекты и измените им свойства

Объект	Свойство	Значение
Метки	Caption	Ваше имя
	Font	Размер 10, полужирный
	Caption	Ваша фамилия
	Font	Размер 10, полужирный
	Name	LblHello
	Caption	(отсутствует)
	BorderStyle	1
	Alignment	2

Командные кнопки	Name	CmdHello
	Caption	Приветствие
	Name	CmdCls
	Caption	Очистить
Текстовые окна	Name	TxtFam
	Text	(отсутствует)
	Name	TxtName
	Text	(отсутствует)
Рамка	Name	Frame1

Caption	(отсутствует)
---------	---------------

2. Приведите свою форму в соответствие с образцом
3. Самостоятельно измените значения свойств **BackColor**, **ForeColor**, **Font** для меток и текстовых окон



4. В окне кода процедуры обработки события Click для объекта CmdHello набрать

```
Frame1.Caption = "Приветствую тебя"
LblHello.Caption = txtFam.Text + " " + txtName.Text
```

5. В окне кода процедуры обработки события Click для объекта CmdCls набрать

```
Frame1.Caption = " "
LblHello.Caption = " "
txtName.Text = " "
txtFam.Text = " "
```

6. Проверьте работу программы на различных входных данных. Остановите работу программы

7. Установите для метки **LblHello** автоподстройку размера, для этого:

- выделите метку
- установите для свойства **AutoSize** значение **True**

8. Проверьте работу.

9. Остановите работу программы. Измените для свойства **AutoSize** значение **False**.

10. Установите блокировку редактирования для текстового окна **TxtFam**, для этого:

- выделите текстовое окно
- в окне свойств для свойства **Locked** установите значение **True**

11. Проверьте работу программы.
12. Остановите работу программы. Отмените блокировку (самостоятельно)
13. Установите ограничение на количество символов для окна **txtName**, для этого:
 - выделите окно
 - в окне свойств для свойства **MaxLength** установите значение **10**
14. Установите разрешение на ввод нескольких строк для поля **txtFam**, для этого:
 - выделите поле
 - для свойства **MultiLine** установите значение **True**
 - для свойства **ScrollBars** установите значение **1 (2, 3)**
15. Проверьте работу программы.
16. Добавьте на форму еще одну метку (**LblInfo**) и командную кнопку (**cmdInfo**)
17. В окне кода процедуры обработки события **Click** для объекта **CmdInfo** набрать:


```
LblInfo.Caption = txtFam.SelText
```
18. Проверьте работу программы:
 - введите любой набор символов в поле **txtFam**,
 - с помощью мышки выделите произвольный фрагмент;
 - щелкните по кнопке **cmdInfo**.

Задание 2

Разработать приложение, которое в зависимости от выбранного переключателя в группе «Метод платежа» отображает в окне приложения только одну из следующих групп переключателей: «Наличными» или «Кредитная карта».

Задание 3

Разработать приложение, позволяющее добавлять и удалять фамилии студентов в список, а также выводить в метку выбранные значения из трех списков (при щелчке по командной кнопке). Значения для списков «Студенты», «Список экзаменов», «Оценка» сформировать на стадии проектирования.

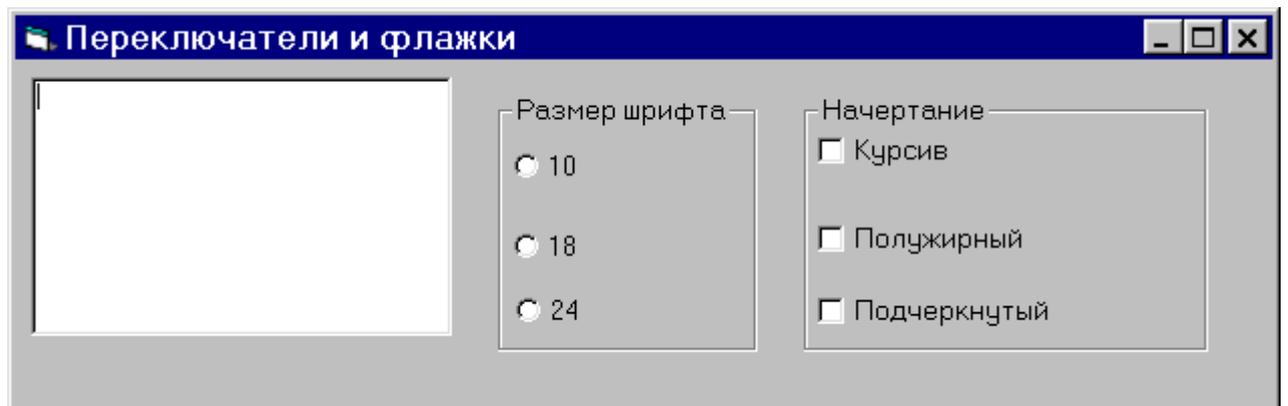
Задание 4

1. Установите на форме следующие объекты и измените им свойства

Объект	Свойство	Значение
Рамки	Caption	Размер шрифта

	Font	Размер 10
	Caption Font	Начертание Размер 10
Переключатели	Name Caption	Opt10 10
	Name Caption	Opt18 18
	Name Caption	Opt24 24
	Name Caption	ChkItalic Курсив
Флажки	Name Caption	ChkBold Полужирный
	Name Caption	ChkUnder Подчеркнутый
	Name Text	Text1 (отсутствует)

2. Приведите свою форму в соответствие с образцом



3. Самостоятельно измените значения свойств **BackColor**, **ForeColor**, **Font** для объектов формы

4. В окне кода процедуры обработки события **Click** для объекта **Opt10** набрать

Text1.FontSize = 10

5. Самостоятельно введите код для других переключателей

6. В окне кода процедуры обработки события **Click** для объекта **ChkItalic** набрать

Text1.FontItalic = ChkItalic.Value

7. Самостоятельно введите код для других флажков

8. Проверьте работу программы: введите текст в текстовое окно и отформатируйте его, используя переключатели и флажки

9. Остановите работу программы
10. Установите на форму еще одну рамку «Цвет шрифта», разметите в ней три переключателя:

OptRed	Красный
OptBlue	Синий
OptGreen	Зеленый
11. В окне кода процедуры обработки события **Click** для объекта **OptRed** набрать
Text1.ForeColor=vbRed
12. Введите код для других переключателей, используя встроенный константы **vbBlue, vbGreen**
13. Проверьте работу программы. Остановите работу программы
14. Измените стиль и цвет для переключателей в рамке «Цвет шрифта», для этого:
 - выделите переключатель **OptRed**
 - установите для свойства **Style** значение **1-Graphical**
 - удалите значение в свойстве **Caption** у переключателя и уменьшите ширину
 - установите для свойства **BackColor** красный цвет
15. По аналогии измените два других переключателя
16. Измените стиль флажков в рамке «Начертание», для этого:
 - выделите флажок **ChkItalic**
 - установите для свойства **Style** значение **1-Graphical**
 - удалите значение в свойстве **Caption**
 - для свойства **Picture** выберите любую иконку (VB\Graphics\Icons\...)
 - при необходимости измените размеры флажка
17. По аналогии измените два других флажка (картинки все разные)
18. Проверьте работу программы.

3.2 Лабораторная работа №2 "ТИПЫ ДАННЫХ. ОПЕРАТОРЫ ПРИСВАИВАНИЯ. ПРОЦЕДУРЫ"

Цель работы: изучить типы данных Visual Basic. Научиться пользоваться операторами описания и присваивания. Рассмотреть различные типы процедур. Получить навыки работы с процедурами и функциями.

Методика выполнения работы

Для работы в VB необходимы два окна:

- a) Окно программы (содержит текст программы, запускается командой View\Code).
- b) Окно отладки (содержит полученные в результате выполнения программы результаты, запускается командой View\Immediate window).

Основное наполнение модуля. Самое начало модуля называется общей областью, в которой располагаются общие описания, например, типа данных, используемого по умолчанию.

Любая программа выполняет обработку данных. Данные могут быть постоянными величинами (константы) и переменными величинами. Переменные и постоянные величины должны быть отнесены к какому-либо типу данных. В Visual Basic существует набор стандартных типов данных.

Типы данных для VB:

1) Byte – Массивы данного типа служит для хранения двоичных данных, например, изображений. Использование данного типа предохраняет двоичные данные во время преобразования формата.

2) Boolean – для хранения логических (булевых) значений. По умолчанию значением булевской переменной является False - ложь.

3) Currency - для хранения чисел с дробной частью до четырех цифр и целой частью до 15 цифр, то есть данных с фиксированной десятичной точкой, удобных для денежных вычислений.

4) Double - для хранения чисел с плавающей десятичной точкой.

5) Integer- для хранения малых целых числовых значений (от -32768 до +32767).

6) Long – для хранения больших целых числовых значений (от -2 147483648 до +2 147483647).

7) Object – поскольку VBA является объектно-ориентированным языком, в нем можно манипулировать различными объектами, адреса, расположения которых в памяти (указатели) имеют этот тип.

8) String – по умолчанию данные строкового типа имеют переменную длину и могут удлиняться или укорачиваться. Однако такие строки занимают на 10байт памяти больше, поэтому можно объявить строки фиксированной длины, указав количество символов. Если количество символов будет меньше объявленного, то свободные места заполняются пробелами, при попытке занесения большего количества символов лишние отбрасываются.

9) Variant – может быть использован для хранения данных всех базовых типов без выполнения преобразования (приведения) типов. Применение данного типа позволяет выполнять операции, не обращая внимание на тип данных, которые они содержат. Удобен для объявления переменных, тип которых заранее неизвестен.

10) Date – используется для хранения даты и времени.

Для объявления типов данных используются операторы описания.

Операторы описания

Объявление переменной производится операторами Dim, Private за которым следует имя переменной и необязательная часть с ключевым словом As, после которого задается тип переменной, например Dim name [As type].

Оператор Dim используется только вне модуля, в его общей части и делает описываемую переменную доступной из всех процедур всех модулей проекта.

Оператор Private служит для объявления переменной уровня модуля, доступной только процедурам данного модуля.

Если необходимо объявить константу, то используется оператор Const (например Const a=5).

Для присвоения же конкретного числового значения переменной используются оператор присваивания.

Оператор присваивания

Инструкция Let Присваивает значение выражения переменной или свойству:

[Let] имяПеременной = выражение

Явное использование ключевого слова Let зависит от вкуса пользователя, обычно это слово опускают, записывая, например, a=5.

Значение выражения может быть присвоено переменной, только если оно имеет совместимый с этой переменной тип данных. Невозможно присвоить строковое выражение числовой переменной или числовое выражение строковой переменной. Такая попытка приведет к ошибке во время компиляции.

Переменным типа Variant могут присваиваться как строковые, так и числовые выражения. Однако обратное не всегда верно.

Присвоение выражения с одним из числовых типов переменной с другим числовым типом данных преобразует значение выражения в тип данных результирующей переменной (т.е. если a - integer, то если let a=2,33, то будем иметь целое значение 2).

Область видимости переменных

Переменные могут быть объявлены в модулях, формах или процедурах.

– Переменная, объявленная в процедуре, доступна для использования только в пределах этой процедуры (*локальная*), даже если она была объявлена как **Static**;

– Переменная, объявленная в **General Declarations** (раздел общих объявлений) формы доступна в пределах всей формы.

– Область видимости переменных, объявленных на уровне модуля, зависит от ключевых слов, использованных при объявлении.

– Если переменная объявлена посредством **Dim**, область видимости – модуль.

– Если переменная объявлена посредством ключевого слова **Public**, область ее видимости – вся программа (переменная - *глобальная*).

После объявления типов данных вводятся процедуры - наборы описаний и инструкций, сгруппированных для выполнения. Существует три типа процедур:

1) процедура Sub - набор команд, с помощью которого можно решить определенную задачу. При ее запуске выполняются команды процедуры, а затем управление передается в приложение пакета MS Office или процедуру, которая вызвала данную процедуру. Закрывается процедура командой End Sub.

2) процедура Function (функция) также представляет собой набор команд, который решает определенную задачу. Различие заключается в том, что такие процедуры обязательно возвращают значение, тип которого можно описать при создании функции.

3) процедура Property используется для ссылки на свойство объекта. Данный тип процедур применяется для установки или получения значения пользовательских свойств форм и модулей.

Задание 1

Набрать и откомпилировать следующую программу для расчёта периметра треугольника со сторонами 4, 5, 4.

```
Dim b As Integer
Dim c As Integer
Dim p As Integer
Const a = 4
Private Sub Command1_click()
b = 5
c = 4
p = a + b + c
Debug.Print a, b, c, p
End Sub
```

Команда Private Sub Command1_click() - служит для определения переменной уровня модуля и процедуры Sub и используется практически в любой программе VBA.

Команда Debug.Print a, b, c, p - отображает значения всех указанных величин в окне "Окно отладки (Immediate)".

Команда End Sub - закрывает процедуру.

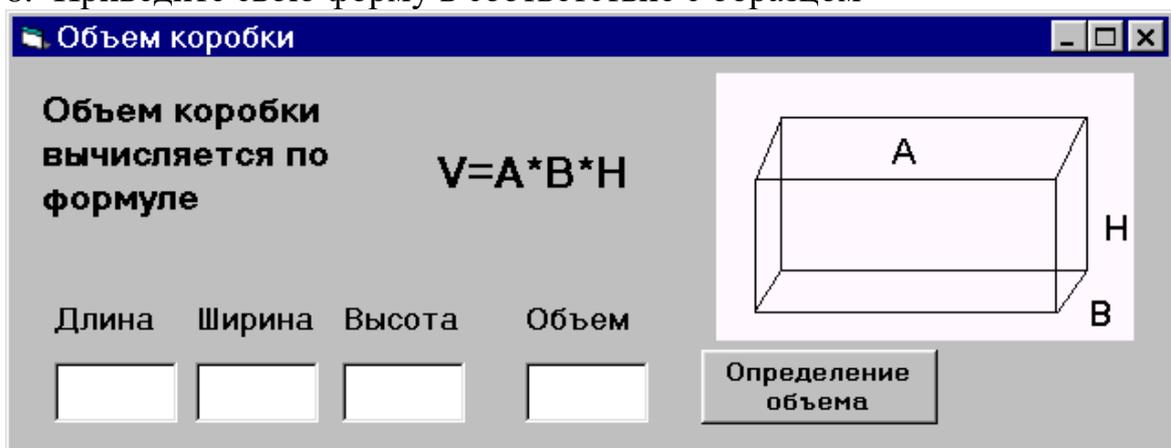
Задание 2

1. Загрузите графический редактор **Paint**.
2. Выберите в меню программы **Рисунок / Атрибуты**
3. В появившемся окне установите *Единицы см, Ширина 6, Высота 3*, нажмите **Ок**.

4. Используя инструментарий программы нарисуйте параллелепипед с обозначением сторон и сохраните свой рисунок.
5. Закройте программу **Paint**
6. Загрузите **Visual Basic**.
7. Установите на форме следующие объекты и измените им свойства

Объект	Свойство	Значение
Метки	Caption	Объем коробки вычисляется по формуле
	Caption	$V=A*B*H$
	Caption	Длина
	Caption	Ширина
	Caption	Высота
Командная кнопка	Name	CmdResult
	Caption	Определение объема
Текстовые окна	Name	Text1
	Name	Text2
	Name	Text3
	Name	Text4
Изображение	<i>Name</i>	Image1
	Picture	(созданный Вами рисунок)

8. Приведите свою форму в соответствие с образцом



9. В окне кода процедуры обработки события **Click** для объекта **CmdResult** набрать

```
Dim a,b,h,v As Integer
a = Val(Text1.Text)
b = Val(Text2.Text)
```

$h = \text{Val}(\text{Text3.Text})$

$v = a * b * h$

$\text{Text4.Text} = v$

10. Проверьте работу программы на различных входных данных.

Задание 3

1. Даны два числа a и b . Получить их сумму, разность и произведение.
2. Даны действительные числа x и y . Получить $(|x| - |y|) / (1 + |x \cdot y|)$.
3. Дана длина ребра куба. Найти площадь грани, площадь полной поверхности и объем этого куба.
4. Даны два действительных положительных числа. Найти среднее арифметическое и среднее геометрическое этих чисел.
5. Даны катеты прямоугольного треугольника. Найти его гипотенузу и площадь.
6. Определить периметр правильного n -угольника, описанного около окружности радиуса r .

7. Даны x, y, z . Вычислить a, b , если

$$\text{а) } a = \frac{\sqrt{|x-1|} - \sqrt{|y|}}{1 + x^2/2 + x^2/4}, b = x(\text{arctg}(z) + e^{-(x+3)});$$

$$\text{б) } a = \frac{3 + e^{y-1}}{1 + x^2|y - \text{tg}(z)|}, b = 1 + |y - x| + \frac{(y-x)^2}{2} + \frac{|y-x|^3}{3};$$

$$\text{в) } a = (1+y) \frac{x+y/(x^2+4)}{e^{-x-2} + 1/(x^2+4)}, b = \frac{1 + \cos(y-2)}{x^4/2 + \sin^2 z};$$

$$\text{г) } a = y + \frac{x}{y^2 + \left| \frac{x^2}{y + x^3/3} \right|}, b = \left(1 + \text{tg}^2 \frac{z}{2} \right);$$

$$\text{д) } a = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y}, b = 1 + \frac{z^2}{3 + z^2/5};$$

$$\text{е) } a = \frac{1 + \sin^2(x+y)}{2 + |x - 2x/(1 + x^2y^2)|} + x, b = \cos^2 \left(\text{arctg} \frac{1}{z} \right);$$

$$\text{ж) } a = \ln \left| \left(y - \sqrt{|x|} \right) \left(x - \frac{y}{z + x^2/4} \right) \right|, \quad b = x - \frac{x^2}{3!} + \frac{x^3}{4!}.$$

8. Дана сторона равностороннего треугольника. Найти площадь этого треугольника.

9. Известна длина окружности. Найти площадь круга, ограниченного этой окружностью.

10. Найти площадь кольца, внутренний радиус которого равен 20, а внешний – заданному числу r ($r > 20$).

3.3 Лабораторная работа №3 "ФУНКЦИИ ВВОДА-ВЫВОДА И ПРЕОБРАЗОВАНИЯ ТИПОВ ДАННЫХ"

Цель работы: изучить функции ввода-вывода данных. Получить навыки использования математических функций и разработки приложений для вычислительных задач. Систематизировать знания по теме «Арифметические операции», формировать умения работать в среде программирования с использованием функций преобразования типов данных.

Методика выполнения работы

MsgBox (“сообщение”, [кнопки, заголовок]) - эта функция отображает диалоговое окно, содержащее сообщение длиной до 1024 символов, в которое с помощью операции конкатенации можно включить значение переменных, а также (необязательно) кнопки для реакции на отображения окна (по умолчанию только кнопка ОК).

При задании сложного диалогового окна, при помощи функции MsgBox используются следующие константы:

1) Для задания внешнего вида окна сообщения:

vbCritical, vbQuestion, vbExclamation, vbInformation.

2) Для задания кнопок в окне сообщения:

vbOkCancel, vbAbortRetryIgnore, vbYesNOCancel, vbYesNO, vbRetryCancel.

3) Для задания дальнейших действий после нажатия на соответствующую кнопку:

vbOk, vbCancel, vbAbort, vbRetry, vbIgnore, vbYes, vbNO.

Пример использования констант при помощи функции MsgBox представлен в программе 1.

Программа 1:

```
Private Sub Решение_click()
```

```
    y = MsgBox("Закреть окно", vbQuestion + vbYesNoCancel, "Сообщение Windows")
```

```
End Sub
```

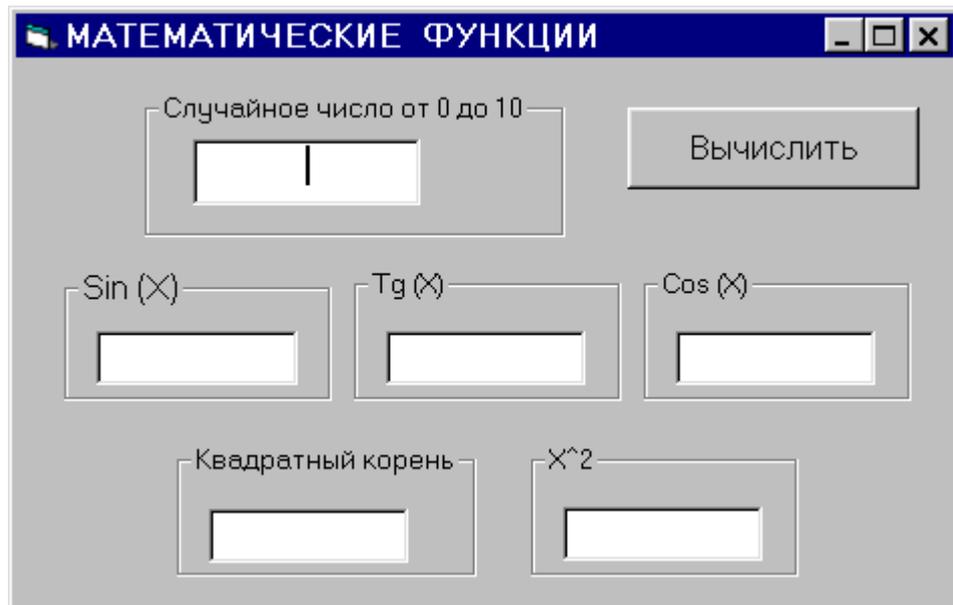
InputBox (“сообщение”[, заголовок] [, значение по умолчанию] [, координата x] [, координата y]) - функция, применяемая для ввода значений переменных в программу. Эта функция отображает диалоговое окно, содержащее окно ввода, кнопки ОК и Отмена, сообщение (подсказку для ввода) и (необязательно) заголовок окна, значение, вводимое по умолчанию, координаты окна по горизонтали и вертикали.

Задание 1

1. Расположите на форме следующие объекты и измените для них свойства:

Объект	Свойство	Значение
Рамка	Caption	Случайное число от 0 до10
Текстовое окно	Name Text	TxtX (отсутствует)
Рамка	Caption	Sin(X)
Метка	Name Caption BorderStyle	LblSin (отсутствует) 1
Командная кнопка	Name Caption	CmdResult Вычислить
Рамка	Caption	Cos(X)
Метка	Name Caption BorderStyle	LblCos (отсутствует) 1
Рамка	Caption	Tn(X)
Метка	Name Caption BorderStyle	LblTan (отсутствует) 1
Рамка	Caption	Квадратный корень
Метка	Name Caption BorderStyle	LblSqr (отсутствует) 1
Рамка	Caption	X^2
Метка	Name Caption BorderStyle	Lbl_2 (отсутствует) 1

2. Приведите свою форму в соответствие с образцом



3. В окне кода процедуры обработки события **Click** для объекта **CmdResult** набрать:


```

x = Rnd * 10
txtX.Text = x
LblSin.Caption = Sin(x)
LblCos.Caption = Cos(x)
LblTan.Caption = Tan(x)
LblSqr.Caption = Sqr(x)
Lbl_2.Caption = x ^ 2

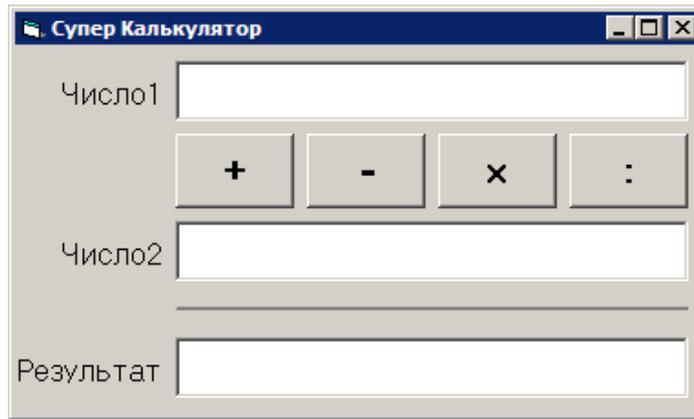
```
4. Загрузите программу и проверьте ее работу
5. Самостоятельно измените форму и программный код для:
 - изменения интервала случайного числа от 0 до 50
 - получения целого случайного числа от 10 до 100
 - определения остатка от деления числа X на 2

Задание 2

Создайте проект калькулятор, производящий простейшие арифметические действия.

Для выполнения расчёта вы набираете в двух верхних текстовых полях первое и второе число, затем щёлкаете по одной из кнопок и в нижнем текстовом поле получаете результат.

Внешний вид калькулятора следующий:



Последовательность действий:

1. Запустите систему программирования Microsoft Visual Basic, выбрав тип проекта Standard EXE.
2. Разместите на форме три текстовых поля и четыре кнопки, выбрав их на Панели объектов.
3. Щёлкнув по кнопке Command1 задайте в Окне свойств объекта (Properties) значение свойства Caption «+», подберите шрифт надписи с помощью свойства Font, смените значение свойства Name со стандартного на «Кл_сложение».
4. Аналогичные действия произведите с остальными кнопками (значения свойства Name будут «Кл_вычитание», «Кл_умножение» и «Кл_деление»).
5. Измените значения свойства Name для текстовых полей на «Число1», «Число2» и «Результат», а значения свойства Text сотрите.
6. Расположите на форме слева от каждого поля надпись с помощью объекта Label (Метка). Поменяйте значения свойства Caption на «Число 1», «Число 2» и «Результат». При необходимости смените шрифт надписи свойством Font.
7. Добавьте горизонтальную линию над полем «Результат» с помощью объекта Line (Линия). Толщину и цвет линии можно сменить, поменяв значения свойств BorderWidth и BorderColor.
8. Откройте Окно программного кода, дважды щёлкнув по клавише сложения, и добавьте такой оператор:

$$\text{Результат.Текст} = \text{Val}(\text{Число1.Текст}) + \text{Val}(\text{Число2.Текст})$$
9. Аналогично запрограммируйте и остальные клавиши (используется тот же оператор, но с другим арифметическим действием). Учтите, что умножение обозначается звёздочкой *, а деление – косой чертой /.
10. Запустите проект. Проверьте правильность выполнения команд.

3.4 Лабораторная работа №4 "ПРОГРАММИРОВАНИЕ РАЗВЕТВЛЯЮЩИХСЯ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ"

Цель работы: Систематизировать знания по теме «Оператор условно-

го перехода», формировать умения разрабатывать разветвляющие программы в среде программирования с использованием переключателей. Освоить технологию программирования разветвляющихся вычислительных процессов в среде VBA с применением операторов условного перехода.

Методика выполнения работы

Для организации разветвлений в программе используется оператор условного перехода, который имеет структуру

```
If <...> Then
    <...>
Else
    <...>
End If
```

После ключевого слова If записывается проверяемое условие. Если оно выполняется, осуществляется переход к операторам, расположенным после ключевого слова Then. Если условие не выполняется, осуществляется переход к операторам после ключевого слова Else. Это ключевое слово должно быть записано на новой строке. Такая запись оператора условного перехода называется блочной.

Пример1. Блочная запись условного оператора.

```
Public Sub ДвеВетви2 ()
    x = InputBox ("введите x")
    If x < 0 Then
        y = 0.7 * x ^ 2
    Else
        y = 5 + x
    End If
    MsgBox ("При x=" + x + vbNewLine + " y=" + Format (y, "##.00"))
End Sub
```

Если число условий больше двух, можно использовать конструкцию

```
If <...> Then <...>
Elseif <...> Then <...>
Else <...>
End if
```

После ключевого слова **If** записывается первое проверяемое условие. Если оно не выполняется, второе условие запишется после ключевого слова **Elseif**.

Если проверяемое условие является составным логическим оператором, то есть состоит из нескольких операторов - условий, то эти условия объединяются логическими операторами **and**, **or**, **not**.

Пример2. Конструкция с двумя условиями.

Если значение переменной *V* меньше 10, то увеличить его в 2 раза, если значение от 10 до 20, то увеличить в 3 раза, в остальных случаях - уменьшить в 10 раз.

```
If V<10 Then
    V=V*2
ElseIf V>=10 And V<=20 Then
    V=V*3
Else
    V=V/10
End If
```

Оператор множественного выбора **Select Case** используется при необходимости осуществить проверку более сложных условий.

Формат команды

```
Select Case арифм_выражение или симв_выражение
Case условие 1
    блок команд 1
Case условие 2
    блок команд 2
.....
Case Else
    блок команд n
End Select
```

В поле операндов **Select Case** записывается произвольное арифметическое выражение или символьное выражение, которое в процессе выполнения программы принимает то или иное числовое, логическое или символьное значение.

В поле операндов каждого оператора **Case** надо указать условие в одном из трех форматов:

1. **Case** константа_1, константа_2,...
2. **Case Is** знак_отношения константа
3. **Case** константа_1 **To** константа_2

Алгоритм множественного выбора заключается в следующем. Сначала вычисляется значение выражения, записанного в **Select Case**. Далее проверяется, удовлетворяет ли это значение одному из указанных в **Case** условий. Если значение удовлетворяет какому-то условию, выполняется блок команд, следующий за данным **Case**. Если ни одно условие не удовлетворяется, выполняется блок команд, следующий за **Case Else**. При выполнении того или иного блока команд управление передается команде, следующей за **End Select**.

Пример3:

```
Input A
Select Case A
```

Case 1,5

Print «А равно 1 или 5»

Case Is >5

Print «А больше 5»

Case -8 TO 2.5

Print «А не меньше -8, но не больше 2.5»

Case Else

Print «Ни одно условие не выполняется»

End Select

Задание 1

1. Откройте новый проект, установите следующие объекты и измените им свойства

Объект	Свойство	Значение
Рамки	Caption	Длины сторон треугольника
	Caption	Результаты проверки
Текстовые окна	Name	TxtA
	Name	TxtB
	Name	TxtC
Метки	Name	LblInfo1
	Caption	(отсутствует)
	Name	LblInfo2
	Caption	(отсутствует)
	Name	LblInfo3
	Caption	(отсутствует)
	Name	LblInfo4
	Caption	(отсутствует)
Командная кнопка	Name	CmdProverka
	Caption	Проверить

2. Приведите свою форму в соответствие с образцом

The image shows a Windows application window with the title bar 'условие'. Inside the window, there are two main sections. The first section, titled 'Длины сторон треугольника', contains three empty text input boxes for entering the side lengths of a triangle. The second section, titled 'Результаты проверки', contains four empty text boxes for displaying the results of the triangle check. At the bottom of the window, there is a button labeled 'Проверить'.

3. В окне кода процедуры обработки события **Click** набрать для объекта **cmdProverka**

```

Dim a, b, c As Integer
Dim info As String
a = Val(txtA.Text)
b = Val(TxtB.Text)
c = Val(TxtC.Text)
'проверка существует ли треугольник
If (a >= b + c) Or (b >= a + c) Or (c >= a + b) Then
    LblInfo1.Caption = "треугольник не существует"
Else
    LblInfo1.Caption = "треугольник существует"
End If
'проверка на равенство сторон
If a = b And a = c Then
    LblInfo2.Caption = "треугольник равносторонний"
Else
    LblInfo2.Caption = "треугольник не равносторонний"
End If
'проверка на равнобедренность
If a = b Or a = c Or b = c Then

```

```

    LblInfo3.Caption = "треугольник равнобедренный"
Else
    LblInfo3.Caption = "треугольник не равнобедренный"
End If
'проверка на наличие прямого угла
If (a ^ 2 = b ^ 2 + c ^ 2) Or (b ^ 2 = a ^ 2 + c ^ 2) Or (c ^ 2 = a ^ 2 + b ^ 2)
Then
    LblInfo4.Caption = "треугольник прямоугольный"
Else
    LblInfo4.Caption = "треугольник не прямоугольный"
End If
4. Проверьте работу программы.
Измените приложение так, чтобы оно определяло, уместится ли треугольник с заданными сторонами в окружность заданного радиуса.

```

Задание 2

Использование условного оператора с двумя разветвлениями. Необходимо вычислить значения величины y для разных значений величины x .

$y = \begin{cases} 0, & \text{если } x \leq 0 \\ x^2 - 2x, & \text{если } 0 < x < 2 \\ 3 \cdot (x - 2), & \text{если } x \geq 2 \end{cases}$	$y = \begin{cases} 0, & \text{если } x < -3 \\ x + 3 + 4, & \text{если } -3 \leq x < 0 \\ \frac{6}{x + 2}, & \text{если } x \geq 0 \end{cases}$
$y = \begin{cases} -(x + 1)^2, & \text{если } x < -1 \\ 1 - x , & \text{если } -1 \leq x \leq 1 \\ (x - 1)^2, & \text{если } x > 1 \end{cases}$	$y = \begin{cases} 0, & \text{если } x \leq -2 \\ -\sqrt{4 - x^2}, & \text{если } -2 < x < 2 \\ \frac{x - 2}{2}, & \text{если } x \geq 2 \end{cases}$
$y = \begin{cases} 0, & \text{если } x < -3 \\ x + 3 + 4, & \text{если } -3 \leq x < 0 \\ \frac{6}{x + 2}, & \text{если } x \geq 0 \end{cases}$	$y = \begin{cases} 0, & \text{если } x \leq -4 \text{ или } x \geq 4 \\ -\sqrt{4 - (x + 2)^2}, & \text{если } -4 < x < 0 \\ \sqrt{4 - (x - 2)^2}, & \text{если } 0 \leq x < 4 \end{cases}$

Задание 3

С использованием оператора выбора Select Case составить программу, запрашивающую через диалоговое окно, какую операцию выполнить:

- 1 – извлечение квадратного корня
- 2 – вычисление угла в градусах по известному синусу
- 3 – вычисление десятичного логарифма
- 4 – вычисление тангенса угла, заданного в градусах. В зависимости от выбранной операции, запрашиваются исходные данные, проверяется их кор-

ректность, и производятся вычисления. Результат и исходные данные выводятся в диалоговое окно.

3.5 Лабораторная работа №5 "ПРОГРАММИРОВАНИЕ ЦИКЛИЧЕСКИХ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ"

Цель работы: научиться в среде Visual Basic разрабатывать проекты, реализующие циклические алгоритмы, овладеть навыками использования оператора циклов по счетчику, предусловия и постусловия, а также построения блок-схем.

Методика выполнения работы

Цикл - это многократно повторяемая часть программы.

В языке Visual Basic предусмотрено два основных способа организации циклов:

- повторение блока команд заданное количество раз (*цикл со счетчиком*);
- циклическое повторение блока команд, пока выполняется (или не выполняется) некоторое условие.

Цикл со счетчиком For ...Next

Цикл For ... Next (Для ... Следующий) - это цикл с заранее заданным количеством повторений.

```
For переменная_цикла=нач_значение To конеч_значение [Step шаг]
    блок_команд
Next переменная_цикла
```

В поле операндов оператора For указываются:

- числовая переменная, которая называется *счетчиком* или *переменной цикла*.
- начальное и конечное значение счетчика.

Если необходимо, укажите шаг изменения счетчика по окончании каждого цикла (по умолчанию этот шаг равен 1).

Конструкция завершается оператором Next.

Цикл Do While...Loop или Do...Loop While

Здесь **While** (пока) и **Loop** (цикл) зарезервированные слова. Циклы типа **While** предназначены для ситуаций, когда количество повторений тела цикла (итераций) заранее не известно.

Синтаксис двух разновидностей цикла While:

1. **Do While** Условие Повторения Группа инструкций **Loop**
2. **Do** Группа инструкций **Loop While** Условие Повторения

Различие между ними заключается в том, что Условие Повторения (условие повторения выполнения тела цикла) проверяется в первом случае до выполнения тела цикла (цикл – пока), а во втором случае – после выполнения тела цикла (цикл – до).

Пример1. Условие работы цикла проверяется до входа в цикл.

```
Счетчик = 0
Номер = 20
Do While Номер > 10
    Номер = Номер - 1
    Счетчик = Счетчик + 1
Loop
```

Пример2. Условие работы цикла проверяется в конце.

```
Счетчик = 0
Номер = 9
Do
    Номер = Номер - 1
    Счетчик = Счетчик + 1
Loop While Номер > 10
```

Цикл Do Until...Loop или Do...Loop Until

Выполнение условия означает необходимость выхода из цикла.

Пример3. Выполнения цикла Do Until...Loop.

В этой программе *условие выхода из цикла проверяется в начале цикла.*

```
Счетчик = 0
Номер = 20
Do Until Номер = 10
    Номер = Номер - 1
    Счетчик = Счетчик + 1
Loop
```

Пример4. Выполнение цикла Do...Loop Until.

В этой программе *условие выхода из цикла проверяется в конце цикла.*

```
Счетчик = 0
Номер = 1
Do
    Номер = Номер + 1
    Счетчик = Счетчик + 1
Loop Until Номер = 10
```

Задание 1

№	Числовая последовательность	Вычислить
---	-----------------------------	-----------

1	$\{a_1 = 0, a_2 = 4, \dots, a_n = a_{n-1} - 3a_{n-2}\}$	Найти сумму отрицательных элементов при $n=25$.
2	$\left\{a_1 = \frac{\cos 2}{12}, a_2 = \frac{\sin 3}{15}, \dots, a_{n-1} - \frac{a_{n-2}}{21}\right\}$	Найти сумму отрицательных элементов при $n=100$.
3	$\left\{a_1 = \frac{\sin 5}{12}, a_2 = \frac{\cos 3}{15}, \dots, a_n = \frac{a_{n-1}}{0,85} - \frac{a_{n-2}}{21}\right\}$	Найти сумму положительных элементов при $n=30$
4	$\frac{1}{\sin 1} + \frac{1}{\sin 1 + \sin 2} + \dots + \frac{1}{\sin 1 + \dots + \sin n}$	Найти сумму n -четных элементов при $n=10$
5	$\left(1 + \frac{1}{1^2}\right) \cdot \left(1 + \frac{1}{2^2}\right) \cdot \dots \cdot \left(1 + \frac{1}{n^2}\right)$	Найти сумму элементов при $n=100$
6	$\frac{\cos 1}{\sin 1} \cdot \frac{\cos 1 + \cos 2}{\sin 1 + \sin 2} + \dots + \frac{\cos 1 + \cos n}{\sin 1 + \dots + \sin n}$	Найти сумму n -нечетных элементов при $n=20$
7	$(1 + \sin 0,1) + (1 + \sin 0,2) + \dots + (1 + \sin 10)$.	Найти сумму n -четных элементов
8	$\frac{1}{a} + \frac{1}{a \cdot (a+1)} + \dots + \frac{1}{a \cdot (a+1) \cdot \dots \cdot (a+n)}$	Найти сумму элементов при $n=50$
9	$a \times (a+1) \times \dots \times (a+n-1)$	Найти сумму элементов при $n=40$
10	$a \times (a-n) \times (a-2n) \times \dots \times (a-n^2)$	Найти сумму элементов при $n=10$

Задание 2

Выполните табулирование функции на множестве значений аргумента x от 1 до 10 с шагом 1,5.

Вариант 1

$$y = (x^2 + 1) \cdot \sin(3x)$$

Вариант 2

$$y = \ln(x+1) \cdot \sin(x)$$

Вариант 3

$$y = (x^2 + 1) \cdot \ln(x+1)$$

Вариант 7

$$y = (x^3 + 2) \cdot \ln(2x+3)$$

Вариант 8

$$y = \lg(3x+10) \cdot \cos(2x)$$

Вариант 9

$$y = (x^4 + 3) \cdot \sin(6x)$$

Вариант 10

$$y = \ln(4x+8) \cdot \sin(10x)$$

Вариант 4

$$y = \lg(x+1) \cdot \sin(x)$$

Вариант 5

$$y = (x^3 + 2) \cdot \cos(4x)$$

Вариант 6

$$y = \ln(2x+3) \cdot \cos(5x)$$

Вариант 11

$$y = (x^4 + 3) \cdot \ln(3x+8)$$

Вариант 12

$$y = \lg(6x+1) \cdot \sin(4x)$$

Вариант 13

$$y = \sin(2x) \cos(3x)$$

Вариант 14

$$y = (x^4 + 3x) \cdot \sin(5x)$$

Задание 3

1. Установите на форме следующие объекты и измените им свойства:

Объект	Свойство	Значение
Командная кнопка	Name	Cmd1
	Caption	Вывести
Рамки	Caption	Параметры оператора цикла
	Caption	Введите текст
	Caption	Текст наоборот
Текстовые окна	Name	Txt1
	Name	Txt2
	Name	TxtStep
	Name	Text1
	Name	Text2
Командные кнопки	Caption	Вывести
	Name	Cmd1
	Caption	Перевернуть
	Name	Cmd2

2. Приведите свою форму в соответствие с образцом

3. В окне кода процедуры обработки события **Click** для объекта **Cmd1** набрать:

```
n1 = Val(Txt1.Text)
```

```
n2 = Val(Txt2.Text)
```

```

n3 = Val(TxtStep.Text)
Cls
For i = n1 To n2 Step n3
Print i
Next i

```

4. Проверьте работу программы для различных значений параметров оператора цикла

5. В окне кода процедуры обработки события **Click** для объекта **Cmd2** набрать:

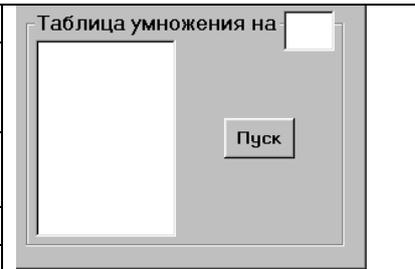
```

s = Text1.Text
k = Len(s)
b = ""
For i = k To 1 Step -1
b = b + Mid(s, i, 1)
Next i
Text2.Text = b

```

6. Проверьте работу программы на различных фразах.

7. Добавьте на форму следующие объекты и измените им свойства.

Объект	Свойство	Значение	
Командная кнопка	Name	Cmd3	
	Caption	Пуск	
Рамка	Caption	Таблица умножения на	
Текстовое окно	Name	TxtNum	
Графическое окно	Name	PicTabl	

8. В окне кода процедуры обработки события **Click** для объекта **Cmd3** набрать

```

n = TxtNum.Text
For i = 1 To 10
PicTabl.Print i; " * "; n; " = "; n * i
Next i

```

9. Проверить работу программы.

Задание 4

Подсчитать сумму заданной числовой последовательности с точностью до 0,000001 согласно вариантам задания.

Вариант	Задание	Вариант	Задание
1	$S(x) = \sum_{n=1}^{\infty} \frac{1}{n^2}$	5	$S(x) = \sum_{n=1}^{\infty} (-1)^{2n+1} \frac{1}{n}$
2	$S(x) = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{1}{2n+1}$	6	$S(x) = \sum_{n=1}^{\infty} \frac{1}{n(n-1)}$
3	$S(x) = \sum_{n=1}^{\infty} \frac{1}{n(n+2)}$	7	$S(x) = \sum_{n=1}^{\infty} \frac{1}{n(n+1)(n+2)}$
4	$S(x) = \sum_{n=0}^{\infty} \frac{1}{n!}$	8	$S(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!}$

Задание 5

Решить задачу нахождения корня нелинейного уравнения методом деления пополам с точностью $\epsilon=0.001$.

Варианты заданий для самостоятельной работы

№ варианта	Уравнение	Интервал
1	$x^4+x^2-2x-2=0$	[-1;0]
2	$\text{tg}(2x) - 1 - x=0$	[0.3;0.7]
3	$2^x-2x^2+1=0$	[1;3]
4	$e^x+x^2-2=0$	[0;1]
5	$\exp(-3x^2)-x=0$	[0;1]
6	$2x-3\ln(x)-3=0$	[3;4]
7	$\ln(x)-1/x=0$	[1;2]
8	$\cos(x)-\ln(x+1)=0$	[0;2]
9	$\ln(x)-\sin^2(x)=0$	[1;3]
10	$x+x^3-5=0$	[1;2]

3.6 Лабораторная работа №6 "РАБОТА С МАССИВАМИ"

Цель работы: научиться в среде Visual Basic разрабатывать проекты, реализующие циклические алгоритмы и использующие для хранения дан-

ных одномерные и двумерные массивы.

Методика выполнения работы

Элементы массива нумеруются, начиная с нуля или единицы. Массив - это набор значений, которые называются *элементами*, логически связанными друг с другом. Элементы массива нумеруются, начиная с нуля или единицы.

Используя массив, можно ссылаться на эти связанные значения по одному имени и использовать число, которое называется *индексом*, для идентификации отдельного элемента на основе его положения в массиве. Индексы массива варьируются от 0 до одного меньше общего числа элементов в массиве.

При использовании синтаксиса Visual Basic для определения размера массива указывается самый высокий индекс, а не общее количество элементов в массиве.

Массивы всегда обрабатываются в цикле. Причем счетчик цикла используется в качестве индекса массива.

В зависимости от количества индексов массивы делятся на одномерные (с одним индексом) и многомерные (с двумя и более индексами).

Одномерный массив – это средство языка программирования, которое позволяет обращаться к любому элементу пронумерованного множества значений. При этом все элементы множества должны иметь одинаковый тип данных, а их количество не должно превосходить некоторого заранее заданного числа, которое называется размером массива.

Объявление массива

Объявление массива выполняется аналогично объявлению переменной. Для этого используются операторы Dim, Static, Public и Private. По способу описания массивы делятся на две основные группы:

- массив с заранее известным числом элементов;
- массив, число элементов которого заранее неизвестно.

При описании массива указывается его имя, затем ставятся круглые скобки, показывающие Visual Basic 2005, что мы организуем массив, и задается тип данных, к которому будут принадлежать все элементы массива.

Используются два случая объявления массива.

Dim a() As Integer.

Эта конструкция описывает одномерный целочисленный массив a типа Integer, размер которого заранее неизвестен. Изначально в нем нет ни одного элемента. Размер этого массива будет определен позднее с помощью оператора ReDim.

Dim b(10) As Single

Такая запись определяет одномерный массив b типа Single, в котором будет содержаться 11 элементов, пронумерованных от 0 до 10. То есть при описании массива в круглых скобках указывается номер его последнего эле-

мента. Нумерация элементов массива всегда начинается с нуля независимо от способа объявления массива.

Для того чтобы обратиться к некоторому элементу массива, необходимо указать имя массива и в круглых скобках номер нужного элемента. Например, $a(3) = 5$ $b(5) = (a(0) + a(4)) / b(1)$.

Ввод массива

Существует два основных способа заполнения массива.

3. В первом случае значения элементов последовательно вводятся с клавиатуры. На первом шаге указывается количество элементов в массиве и соответствующим образом переопределяется размер массива. На втором шаге организуется цикл, на каждом шаге которого вводится значение одного элемента. Размер массива храниться в переменной n . Для организации цикла For потребуется счетчик i .

```
Dim n, i As Integer
Dim a() As Integer
n= Val(TextBox("Введите размерность массива"))
For i = 0 To n
a(i) = Val(TextBox("Введите " + i + "-й элемент массива"))
Next i
```

4. Второй способ ввода массива – это заполнение случайными числами. В Visual Basic есть специальная функция, которая по определенным правилам генерирует рациональные случайные числа в диапазоне (0; 1). Она называется Rnd(). Используя эту функцию, можно заполнить массив случайными числами из любого диапазона. Начало и конец диапазона значений задаются с клавиатуры.

```
start = Val(TextBox("Введите начало диапазона"))
fin = Val(TextBox("Введите конец диапазона"))
Randomize()
For i = 0 To n
a(i) = Math.Round(start + (fin - start) * Rnd())
Next i
```

Вывод массива

Массивы можно выводить в окно списка или в текстовое поле.

Вывод массива в окно списка практически не отличается от вывода совокупности.

Сначала очищаем окно списка. `lstA.Items.Clear()`

Выводим в окно списка поясняющий текст и заголовки колонок.

```
lstA.Items.Add("Исходный массив")
```

`lstA.Items.Add("Номер" + vbTab + "Значение")` Константа `vbTab` позволяет организовать вывод в две колонки.

Организуем цикл для вывода значений элементов массива

```
For i = 0 To n
```

```
lstA.Items.Add(Str(i) + vbTab + Str(a(i)))
Next i
```

На каждом шаге цикла в окно списка выводим номер элемента – i и его значение – $a(i)$.

Для вывода массива в текстовое поле нам потребуется специальная переменная, в которой будет формироваться выводимая строка. Так как массив выводится в текстовое поле, то эта переменная будет иметь символьный тип данных, а точнее – String. Dim s As String.

Сначала в этой строке нет никакой информации. Она пустая. Пустая строка обозначается парой кавычек, между которыми нет ни одного символа. Организуем цикл для формирования строки выводимой информации.

```
s = ""
For i = 0 To n
s = Str(a(i)) + " "
Next i
txtA.Text = s
```

Матрицы – двумерные массивы вводятся и выводятся с помощью вложенных циклов: один перебирает строки, другой – столбцы матрицы. Какой из циклов внешний – неважно, главное – вложенные циклы позволяют перечислить все сочетания номеров строк и столбцов, т.е. учесть все элементы матрицы. Рассмотрим, например, матрицу M размера 5×7 .

Ввод и вывод двумерного массива (матрицы)

Ввод матрицы с клавиатуры:

```
For i = 1 To 5 'перебираем строки
For j = 1 To 7 'перебираем столбцы
M(i,j) = InputBox("Введите элемент матрицы")
Next j: Next i
```

Задание 1

Ввод элементов одномерного массива с клавиатуры. Вывод массива с помощью MSFlexGrid.

№	Задание
1	Найти сумму положительных элементов массива и их количество.
2	Найти произведение нечетных элементов массива и их количество.
3	Найти сумму элементов массива с нечетными индексами.
4	Найти сумму всех элементов кратных трем и их количество.
5	Найти среднее арифметическое отрицательных элементов и их количество.
6	В одномерном массиве произвольных чисел вычислить произведение отрицательных элементов, имеющих нечетные индексы.

7	Найти среднее арифметическое положительных элементов массива и их количество.
8	Для целочисленного массива $A[1..15]$ выяснить, является ли сумма его элементов четным числом. Вывести "ДА" или "НЕТ".
9	Найти произведение положительных, сумму отрицательных элементов.
10	Найти сумму всех элементов кратных двум и их количество.
11	Найти общее количество нулевых элементов и сумму всех элементов.
12	Найти сумму всех элементов кратных пяти и их количество.

Задание 2

1. Даны целые числа a_1, a_2, a_3 . Получить целочисленную матрицу $B[3 \times 3]$, для которой $b_{ij} = a_i - 3a_j$.

2. Даны действительные числа $a_1, \dots, a_{10}, b_1, \dots, b_{20}$. Получить действительную матрицу $C[20 \times 10]$, для которой $c_{ij} = a_j / (1 + |b_i|)$.

3. Получить $A[10 \times 12]$ – целочисленную матрицу, для которой $a_{ij} = i + 2j$.

4. Даны натуральное число n , действительная матрица размера $n \times 9$. Найти среднее арифметическое:

- а) каждого из столбцов;
- б) каждого из столбцов, имеющих четные номера.

5. Дано натуральное число n . Выяснить, сколько положительных элементов содержит матрица $A[n \times n]$, если $a_{ij} = \sin(i^2 - j^2/n)$.

6. Дана действительная матрица размера $n \times m$, в которой не все элементы равны нулю. Получить новую матрицу путем деления всех элементов данной матрицы на ее наибольший по модулю элемент.

7. Дана действительная квадратная матрица порядка 12. Заменить нулями все ее элементы, расположенные на главной диагонали и выше нее.

8. Дана действительная матрица размера $n \times m$. Определить числа b_1, \dots, b_n , равные соответственно:

- а) суммам элементов строк;
- б) произведениям элементов строк;
- в) наименьшим значениям элементов строк;
- г) значениям средних арифметических элементов строк;
- д) разностям наибольших и наименьших значений элементов строк.

Задание 3

Ввод элементов массива с клавиатуры. Вывод массива с помощью MSFlexGrid.

№	Задание на двумерные массивы
---	------------------------------

1	Дана матрица $A(n, n)$. Вычислить сумму диагональных элементов
2	Вычислить сумму граничных элементов матрицы $A(n, m)$
3	Определить число отрицательных и положительных элементов матрицы $A(n, m)$.
4	Определить максимальный элемент в каждой строке матрицы $A(n, m)$.
5	В каждой строке двумерной матрицы (n, m) , состоящей из произвольных чисел, вычислить количество, сумму и среднее арифметическое отрицательных элементов.
6	В двумерной матрице $A(N, N)$, состоящей из произвольных чисел, вычислить сумму элементов, сумма индексов которых равна N .
7	В двумерной матрице $A(N, N)$, состоящей из произвольных чисел, вычислить среднее арифметическое положительных чисел и количество нулевых элементов.
8	Определить минимальный элемент в каждой строке матрицы $A(n, m)$.
9	В двумерной матрице $A(N, M)$, состоящей из произвольных чисел, найти количество положительных элементов в каждом столбце.
10	В двумерной матрице $A(N, M)$, состоящей из произвольных чисел, в каждой строке найти сумму и количество элементов кратных $K1$ ($K1$ - вводится с клавиатуры).

3.7 Лабораторная работа №7 "ПРОЦЕДУРЫ И ФУНКЦИИ"

Цель работы: рассмотреть в Visual Basic различные типы подпрограмм. Научиться использовать описание, вызов подпрограмм функций и процедур.

Методика выполнения работы

Программы, реализующие вспомогательный алгоритм, называются подпрограммами. Подпрограмму можно оформить как процедуру или как функцию.

Процедура - это некий блок кода, который будет выполняться всякий раз при вызове этой процедуры. Каждая процедура начинается зарезервированным словом Sub и заканчивается End.

Функция - это некий блок кода, который будет возвращать значение. Этим, и только этим функции отличаются от процедур.

Процедура Function — это ряд операторов Visual Basic, заключенных в операторы Function и End Function. Процедура Function выполняет задачу, а затем возвращает управление вызывающему коду. При возврате элемента управления он также возвращает значение вызывающему коду.

При каждом вызове процедуры выполняются ее инструкции, начиная с первого исполняемого оператора после Function оператора и заканчивая первым End Function встречным оператором, Exit Function или Return.

Процедуру Function можно определить в модуле, классе или структуре. По умолчанию это Public означает, что вы можете вызывать его из любого места в приложении с доступом к модулю, классу или структуре, в которой он определен.

Процедура Function может принимать аргументы, такие как константы, переменные или выражения, которые передаются ей вызывающим кодом.

Основные отличия функции от процедуры:

1. Функция имеет тип (аналогично переменной) и может возвращать значение через имя функции. Ее можно использовать качестве операнда в выражениях. Процедура возвращает результаты через параметры, ее нельзя использовать в выражениях.

2. Вызов функции осуществляется посредством указания ее имени в каком либо операторе языка программирования. Процедура вызывается с помощью отдельного оператора.

Прежде чем использовать процедуру или функцию в программе, ее нужно объявить (описать).

Подпрограмма типа Sub

При описании процедуры, как и при описании функции, ей дается уникальное название – имя процедуры и устанавливается список параметров. Параметры разделяются между собой запятой и содержат описание типов.

Общий вид описания процедуры:

Sub название_процедуры ([параметры])

операторы процедуры

End Sub

Здесь:

название процедуры — имя создаваемой вами процедуры Sub;

параметры — необязательный список формальных параметров, используемых в процедуре Sub;

операторы процедуры — блок операторов, который выполняет работу процедуры.

Общий вид описания функции:

Function название_функции ([параметры]) As тип

операторы функции

[Return значение]

End Function

Здесь: название_функции — имя создаваемой вами функции.

As тип — пара ключевых слов, которые определяют тип возвращаемого функцией значения.

параметры — необязательный список параметров, используемых в данной функции.

операторы функции — это блок операторов, который выполняет работу функции.

Return — новый оператор Visual Basic .NET. Он позволяет указать, когда нужно выйти из функции, чтобы вернуть значение в вызывающую программу, и каково это возвращаемое значение. В качестве альтернативы можно использовать синтаксис Visual Basic 6 и возвращать значение в вызывающий код, присвоив значение имени функции.

Вызов подпрограммы типа Function

Для вызова функции в программе, необходимо поместить в оператор программы имя этой функции и все требуемые для нее аргументы (фактические параметры).

Пример вызова функции в операторе присваивания:

Имя_переменной = название_функции(фактические параметры)

Возврат значений

Значение, отправляемое процедурой Function обратно в вызывающий код, называется его возвращаемым значением.

Процедура возвращает это значение одним из двух способов:

Он использует оператор **Return** указания возвращаемого значения и немедленно возвращает управление вызывающей программе. Это показано в следующем примере.

```
Function FunctionName [(ParameterList)] As ReturnType
    Return Expression
End Function
```

Он присваивает значение собственному имени функции в одном или нескольких инструкциях процедуры. Элемент управления не возвращается вызывающей программе, пока не **Exit Function** будет выполнен оператор или **End Function**. Это показано в следующем примере.

```
Function FunctionName [(ParameterList)] As ReturnType
    FunctionName = Expression
End Function
```

Преимущество назначения возвращаемого значения имени функции заключается в том, что элемент управления не возвращается из процедуры до тех пор, пока не будет обнаружен **Exit Function** оператор или **End Function**. Это позволяет назначить предварительное значение и при необходимости скорректировать его позже.

Вызов процедуры

Процедура вызывается из другого места в коде. Это называется *вызовом процедуры*. После завершения процедуры она возвращает управление в код, из которого она была вызвана (*вызывающий код*). Вызывающий код — это оператор или выражение в составе оператора, в которых указывается имя процедуры и передается управление процедуре.

Возврат из процедуры

После завершения процедуры она возвращает управление вызывающему коду. Для этого он может использовать оператор `Return`, соответствующий оператор `exit` для процедуры или оператор конечного (ключевого слова) процедуры. Затем управление передается в вызывающий код, следующий за точкой вызова процедуры.

При использовании оператора `Return` управление немедленно возвращается в вызывающий код. Операторы, следующие за оператором `Return`, не выполняются. В одной и той же процедуре можно использовать несколько операторов `Return`.

При использовании операторов `Exit Sub` или `Exit Function` управление немедленно возвращается в вызывающий код. Операторы, следующие за оператором `Exit`, не выполняются. В одной и той же процедуре можно использовать несколько операторов `Exit` и одновременно использовать операторы `Return` и `Exit`.

Если процедура не содержит операторов `Return` или `Exit`, она завершается оператором `End Sub`, `End Function`, `End Get` или `End Set`, который следует за последним оператором в составе процедуры. При использовании оператора `End` управление немедленно возвращается в вызывающий код. В процедуре можно использовать только один оператор `End`.

Параметры и аргументы

В большинстве случаев процедура должна работать с разными данными при каждом вызове. Эти данные можно передать в процедуру при ее вызове. В процедуре можно определить *параметры* (ноль или более), каждый из которых представляет одно из передаваемых значений. Каждому параметру в определении процедуры соответствует *аргумент* при вызове процедуры. Аргумент представляет значение, которое передается в соответствующий параметр для заданного вызова процедуры.

Вызов процедуры `Function` в выражении

`Function` Используйте имя процедуры так же, как и переменную. Вызов процедуры можно использовать `Function` везде, где можно использовать переменную или константу в выражении.

Следуйте имени процедуры с круглыми скобками, чтобы заключить список аргументов. Если аргументы отсутствуют, при необходимости можно

опустить круглые скобки. Однако использование круглых скобок упрощает чтение кода.

Поместите аргументы в список аргументов в круглые скобки, разделенные запятыми. Убедитесь, что аргументы указаны в том же порядке, в Function чем процедура определяет соответствующие параметры.

Значение, возвращаемое процедурой, участвует в выражении так же, как и значение переменной или константы.

Вызов процедуры функции в инструкции присваивания Function

Используйте имя процедуры после знака равенства (=) в инструкции присваивания.

Следуйте имени процедуры с круглыми скобками, чтобы заключить список аргументов. Если аргументы отсутствуют, при необходимости можно опустить круглые скобки. Однако использование круглых скобок упрощает чтение кода.

Поместите аргументы в список аргументов в круглые скобки, разделенные запятыми. Убедитесь, что аргументы указаны в том же порядке, Function в чем процедура определяет соответствующие параметры, если вы не передаете их по имени.

Значение, возвращаемое процедурой, хранится в переменной или свойстве в левой части инструкции присваивания.

Задание 1

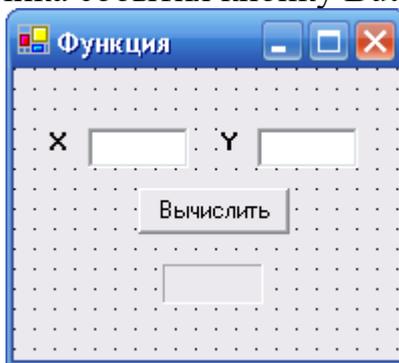
Создать проект, в котором определяется функция (например, умножения двух чисел $F = x * y$), выводится на надпись значение функции.

Разместить на форме:

- для вывода значений аргументов два текстовых поля TextBox1 и TextBox20;

- для вывода значений функции надпись label1;

- для создания обработчика события кнопку Button1.



Создать код вызываемой функции:

```
Function F(ByVal x, ByVal y) As Integer
```

```
    F = x * y
```

```
End Function
```

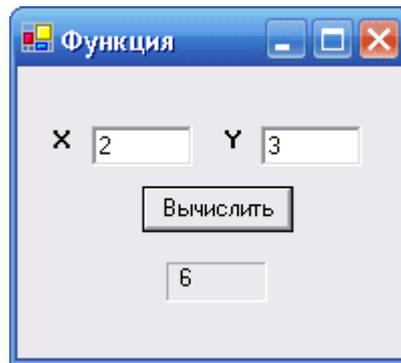
Объявить переменные и создать код процедуры-обработчика:

```

Private Sub Button1_Click(ByVal sender As System.Object,
    _ByVal e As System.EventArgs) Handles Button1.Click
    Dim x, y As Integer
    x = Val(TextBox1.Text)
    y = Val(TextBox2.Text)
    Label1.Text = Str(F(x, y))
End Sub

```

Запустить проект на выполнение и ввести в текстовые поля значения переменных (например, 2 и 3). На надписи будет выведено вычисленное значение.



Задание 2

Возьмём задачу нахождения максимального из вводимых в компьютер чисел.

Разобьём программу на последовательность процедур:

```

Dim max, Ch As Double
Dim i, n As Integer
Private Sub VMax_click()
:VVOD: USL: PECH:
End Sub
    Private Sub VVOD()
        max = InputBox("Введите предполагаемый максимум")
        n = InputBox("Введите количество рассматриваемых чисел")
    End Sub
Private Sub USL()
For i = 1 To n
Ch = InputBox("Введите число", i)
If Ch > max Then max = Ch
Next i
End Sub
    Private Sub PECH()
        Debug.Print "Максимум равен "; max
    End Sub

```

Вышеизложенная методика, используемая для программ, содержащих большое количество операций, позволяет оптимизировать размер программы и упростить процесс её написания.

Задание 3

Создание процедур и функций

1. Даны действительные числа s, t . Получить $f(t, -3s, 3.22) + f(5.2, t, s-t)$,

$$\text{где } f(a, b, c) = \frac{2a - b - \sin c}{5 + |c|}$$

2. Даны действительные числа s, t . Получить $g(1.2, s) + g(t, s) - g(2s-1, st)$,

$$\text{где } g(a, b) = \frac{a^2 + b^2}{a^2 + 2ab + 3b^2 + 4}$$

3. Даны действительные числа a, h , натуральное число n . Вычислить

$$f(a) + 2f(a + h) + \dots + 2f(a + (n-1)h) + f(a + nh),$$

$$\text{где } f(x) = (x^2 + 1) \cos^2 x.$$

4. Даны действительные числа a, b, c . Получить

$$\frac{\max(a, a + b) + \max(a, b + c)}{1 + \max(a + bc, 6.23)}$$

5. Даны действительные числа a, b . Получить

$$u = \min(a, b), v = \min(ab, a + b), x = \min(u^2 + v, 3.14)$$

6. Даны действительные числа s, t . Получить

$$h(s, t) + \max(h^2(s-t, st), h^3(s-t, s+t)) + h(1, 1),$$

$$\text{где } h(a, b) = \frac{a}{1+b^2} + \frac{b}{1+a^2} - (a-b)^2.$$

7. Даны действительные числа $x_1, y_1, x_2, y_2, \dots, x_{10}, y_{10}$. Найти периметр десятиугольника, вершины которого имеют соответственно координаты $(x_1, y_1), (x_2, y_2), \dots, (x_{10}, y_{10})$. (Определить функцию вычисления расстояния между двумя точками, заданными своими координатами).

8. Даны натуральное число n , действительные числа $x_1, y_1, x_2, y_2, \dots, x_n, y_n$. Найти площадь n -угольника, вершины которого при некотором последовательном обходе имеют координаты $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. (Определить функцию вычисления площади треугольника по координатам его вершин.)

9. Составить функцию, позволяющую определить позицию самого правого вхождения заданного символа в исходную строку. Если строка не содержит символа, результатом работы процедуры должна быть -1.

10. Составить процедуру, изменяющую в исходной строке символов все единицы нулями и все нули единицами. Замена должна выполняться, начиная с заданной позиции строки.

3.8 Лабораторная работа №8 "ИСПОЛЬЗОВАНИЕ ГРАФИКИ В VISUAL BASIC"

Цель работы: изучить графические возможности языка Visual Basic. Научиться пользоваться различными графическими примитивами и методами. Освоить построение графика функции в графическом окне.

Методика выполнения работы

На форме или в графическом поле можно рисовать различные графические примитивы с использованием графических методов. В таблице приведены примеры использования этих методов. В качестве объекта *object*, куда выводятся графические примитивы, может служить сама форма (в этом случае имя объекта *Form* можно не указывать) или графическое окно *PictureBox*.

Таблица - Графические примитивы

Наименование	Синтаксис и комментарии
Точка	<code>object.Pset (X,Y), C</code> X,Y – координаты точки, C – цвет.
Окружность	<code>object.circle (X, Y) , R, C</code> X, Y – координаты центра в выбранной системе координат, R – радиус, C – цвет.
Дуга окружности	<code>object.circle (X, Y) , R, C, A, B</code> X, Y – координаты центра, R – радиус, C – цвет. A, B – углы дуги в радианах. дуга строится против часовой стрелки от A к B.
Круговой сектор	<code>object.circle (X, Y) , R, C, -A, -B</code> Минус перед углами означает, что из центра к концам дуги строятся отрезки прямых, образуя угловой сектор.
Овал	<code>object.circle (X, Y) , R, C, , , K</code>

	К – коэффициент сжатия овала. При $0 < K < 1$ сжатие по горизонтали, при $K > 1$ – по вертикали.
Отрезок линии	object.Line (X1,Y1) –(X2,Y2), C X1, Y1 – координаты точки начала отрезка, X2, Y2 – его конца, C – цвет.
Прямоугольник	object.Line(X1,Y1) –(X2,Y2), C, B X1, Y1 — координаты левой верхней вершины прямоугольника, X2, Y2 — координаты правой нижней вершины, C – цвет.
Прямоугольник закрасенный	object.Line (X1,Y1) –(X2,Y2), C, BF X1, Y1 — координаты левой верхней вершины прямоугольника, X2, Y2 — координаты правой нижней вершины, C – цвет.
Очистка нарисованного	object.Cls
Возвращение цвета точки с указанными координатами	object.Point (X,Y)
Вывод строки символов	object.Print [output] В качестве output может быть строковое или числовое выражение. Вывод осуществляется от последней построенной точки изображения. Для указания точки вывода можно использовать метод Pset(X,Y).
Масштабирование окна вывода	object.Scale (X1, Y1) – (X2,Y2) (X1, Y1) и (X2,Y2) – «мировые» координаты выводимого изображения, левой верхней и правой нижней вершины окна соответственно

При построении изображения в графическом окне или на самой форме важным является выбор масштаба по вертикальной и горизонтальной осям. Масштаб устанавливается с помощью метода **Scale**.

В Visual Basic единицей измерения в графике по умолчанию являются **ТВИПЫ**.

Графические методы

1. Line(x1,y1)-(x2,y2),Color - рисует линию; x1 и y1 –это начальные координаты линии, x2 и y2 – это конечные координаты, Color- это цвет (может

задаваться функцией RGB(R,G,B), R-красный, G-зеленый, B-синий. Цвета могут принимать значение от 0 до 255.)

Пример: Line(100,150)-(250,300),RGB(255,0,0)

2. Line(x1,y1)-(x2,y2),Color,B - рисует квадрат x1 и y1 –это начальные координаты линии, x2 и y2 – это конечные координаты, Color- это цвет (может задаваться функцией RGB(R,G,B), R-красный, G-зеленый, B-синий. Цвета могут принимать значение от 0 до 255.)

B-параметр для рисования квадрата.

Пример:

– Line (2000, 1000)-(5000, 3000) , vbRed , B '(Прямоугольник красного цвета)

– Line (2000, 1000)-(5000, 3000) , vbRed , BF '(Прямоугольник красного цвета, залитый этим же цветом)

– Line (2000, 1000)-(5000, 3000) , , B '(Прямоугольник. Цвет его определяется свойством ForeColor, так как в операторе там, где должен быть указан цвет, стоит пустота)

3. Circle(X,Y),R,Color- рисует окружность Circle ' (x_центра , y_центра), радиус , цвет , начальный_угол , конечный_угол , сжатие X и Y - это центр окружности, R -это радиус окружности color- цвет окружности

Пример:

– Circle(200,200),100, RGB(0,255,0)

– Circle (4000, 2000), 1000 (Окружность с центром в точке с координатами (4000, 2000) и радиусом 1000)

– Circle (4000, 2000), 1000, vbRed (Красная окружность)

– Circle (4000, 2000), 1000, 1, 3 (Дуга окружности, начинающаяся от угла в 1 радиан и кончающаяся углом в 3 радиана. Угол отмеряется от направления на восток против часовой стрелки)

– Circle (4000, 2000), 1000, -1, -3 (Сектор круга, начинающийся от угла в 1 радиан и кончающийся углом в 3 радиана. Угол отмеряется от направления на восток против часовой стрелки)

– .Circle (4000, 2000), 1000, 2 (Эллипс с центром в точке с координатами (4000, 2000). Получен из окружности радиусом 1000 горизонтальным сжатием в 2 раза)

– .Circle (4000, 2000), 1000, 1/3(Эллипс с центром в точке с координатами (4000, 2000). Получен из окружности радиусом 1000 вертикальным сжатием в 3 раза)

– Circle (4000, 2000), 1000, 1, 3, 2(Дуга эллипса)

– Circle (4000, 2000), 1000, -1, -3, 2(Сектор эллипса)

4. PSet (x, y), Color-рисует точку X и Y -это координаты точки, Color-цвет.

Пример. Построение графика функции

Private Sub Command1_Click ()

```

Const Pi = 3.141529
Picture1.Scale (-1, 2) - (1, -2) 'масштабирование окна вывода
Picture1.Line (-1, 0) - (1, 0) 'построение оси «Y»
Picture1.Line (0, -2) - (0, 2) 'построение оси «X»
For I = -1 To 1 Step 0.25
Picture1.PSet (I, 0) ' вывод числовых значений под осью «X»
Picture1.Print I
Next I
For I = -3 To 3
Picture1.PSet (0, I) 'вывод числовых значений рядом с осью «Y»
Picture1.Print I
Next I
For X = -1 To 1 Step 0.002
Y = Exp (X) * Sin (2 * Pi * X) 'вывод точек графика функции Y(X)
Picture1.PSet (X, Y)
Next X
End Sub

```

Задание 1

Создать проект «Построение графических примитивов». В графическом окне построить все графические примитивы (точку, линию, прямоугольник, закрашенный прямоугольник, окружность, эллипс, дугу, сектор).

Задание 2

Заполнить форму разноцветными точками в хаотичном порядке.

Для кнопки Command1 «Звездное небо» введите программный код:

```

Private Sub Command1_Click()
Me.DrawWidth = 3
For I = 1 To 1000
PSet (Width * Rnd, Height * Rnd), 16777216 * Rnd
Next
End Sub

```

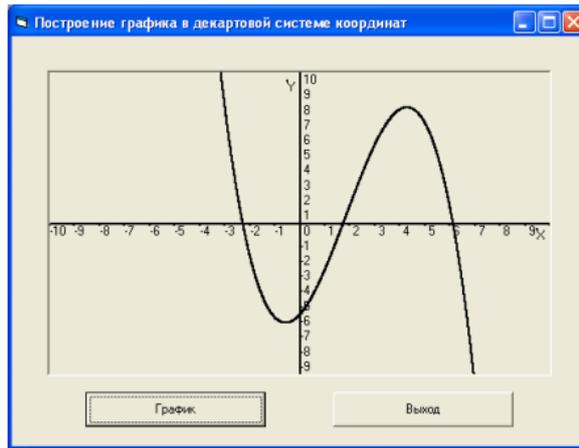
Задание 3

Построить график функции $y = -0.25x^3 + 1.4x^2 + 1.8x - 6$ на отрезке значений аргумента $[-10; 10]$.

Анализ показывает, что область значений функции на этом отрезке: $-10 < y < +10$.

На форме разместим графическое поле PictureBox и две командные кнопки для построения графика и выхода из программы.

Изменим свойство Caption у формы Form1 на «Построение графика в декартовой системе координат».



Программный код:

```

Dim x As Single, i As Integer
Private Sub Command1_Click()
Picture1.Scale (-10, 10)-(10, -10) 'Масштабирование
Picture1.DrawWidth = 2 'задание толщины рисуемых объектов
Picture1.Line (0, -10)-(0, 10) 'рисование оси Y
Picture1.Line (-10, 0)-(10, 0) 'рисование оси X
Picture1.PSet (9.5, -0.3), QBColor(0) ' подпись оси X
Picture1.Print "X"
Picture1.PSet (-0.5, 9.5), QBColor(0) ' подпись оси Y
Picture1.Print "Y"
' ось X
For i = -10 To 10
    Picture1.PSet (i, 0)
    Picture1.Print i
Next i
' ось Y
For i = -10 To 10
    Picture1.PSet (0, i)
    Picture1.Print i
Next i
'Построение графика по точкам
For x = -10 To 10 Step 0.005
    Picture1.PSet (x, f(x))
Next x
End Sub
Private Sub Command2_Click() 'Программный код для кнопки Выход
End
End Sub
' Задание пользовательской функции
Private Function f(x As Single) As Double
f = -0.25 * x ^ 3 + 1.4 * x ^ 2 + 1.8 * x - 6
End Function

```

Задание 4

Создать проект для построения графика функции $y(x)$, диапазоны изменения $x \in [-10, 10]$, y задан в таблице.

Вариант	$y(x)$
1	$x^4 - 4x^3 + 2x^2 - 20x + 10$
2	$x^4 + 6x^3 + 10x^2 - 4x - 6$
3	$x^4 - 3x^2 - 4x - 7$
4	$x^4 - x^3 + 5x^2 - 17x + 2$
5	$x^4 - x^3 - 29x^2 + 71x - 20$
6	$x^4 - 7x^3 + 5x^2 + 25x - 10$
7	$x^4 - 3x^3 - 7x^2 + 18x - 2$
8	$x^4 - 6x^3 + 4x^2 + 10x - 3$
9	$x^4 + 5x^3 - 13x^2 - 15x - 1$
10	$x^4 - 3x^3 + x^2 - 5x + 2$
11	$x^4 - 4x^3 + 3x^2 - 20x + 50$
12	$x^4 + 5x^3 + 7x^2 + 7x - 10$

3.9 Лабораторная работа №9 "НАСТРОЙКА ГРАФИЧЕСКОГО ОКНА. СПОСОБЫ ВВОДА КОМАНД. РЕЖИМЫ РИСОВАНИЯ"

Цель работы: изучить структуру графической системы AutoCAD. Научиться пользоваться различными компонентами окна программы AutoCAD, задавать формат и точность представления линейных и угловых единиц, использовать при построении объектов различные системы координат и режимы рисования.

Методика выполнения работы

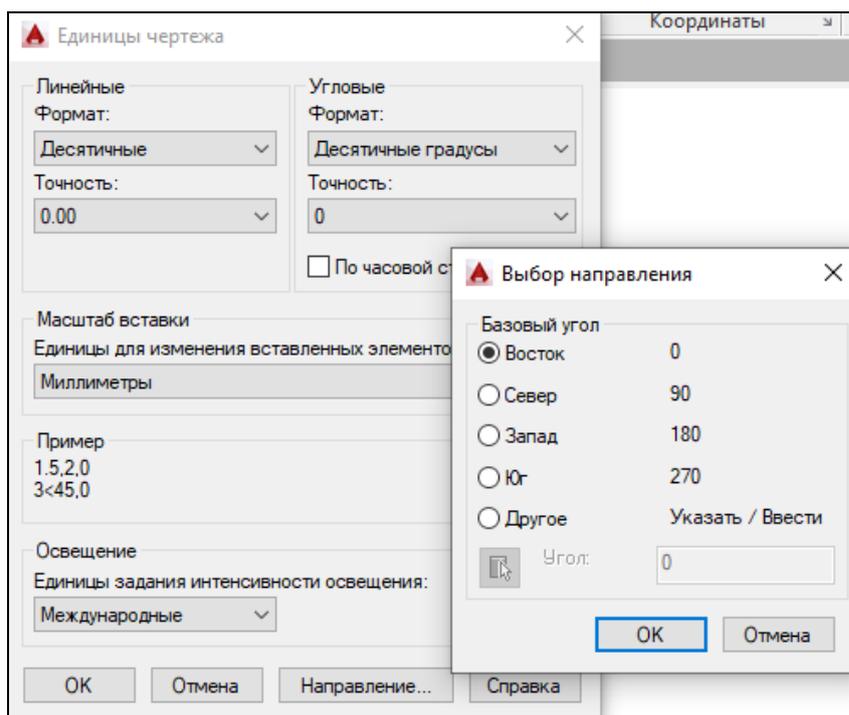
При выполнении работы использовать следующие темы лекционного курса:

- Запуск системы AUTOCAD.

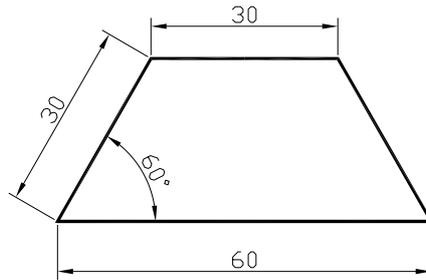
- Настройка графического окна программы.
- Точность единиц измерения.
- Системы абсолютных и относительных координат.
- Режимы рисования.

Задание 1

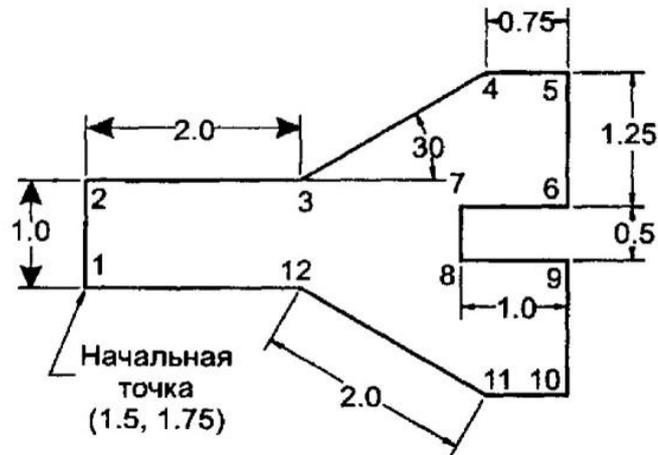
1. Установить границы чертежа формата А3 - альбомная.
2. Задать согласно рисунку формат и точность представления линейных и угловых единиц.



3. Изменить фон области рисунка на серый цвет.
4. Установить размер курсора – 6, цвет автопривязки – фиолетовый.
5. Отразить на ленте следующие панели – Редактирование-2, Зумирование и Мультиталиния.
6. Установить максимальный размер области рисования. Возвратить обычный вид окна.
7. При помощи команды *Отрезок*, используя абсолютные, относительные и полярные координаты построить трапецию. Начальная точка – т.А (10,20) от начала координат.

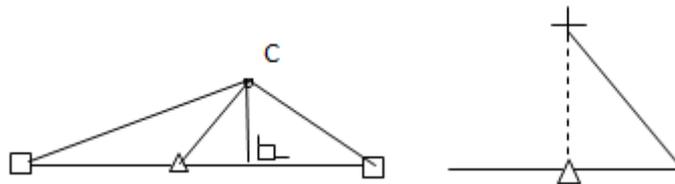


8. Построить следующий рисунок, используя абсолютные, относительные и полярные координаты.

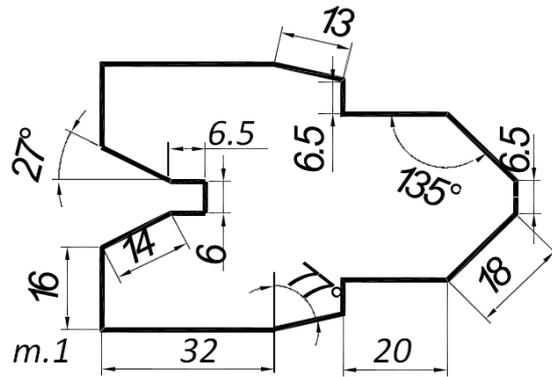


9. Включить режим ОРТО и ПРИВЯЗКА. Используя метод Направление-Расстояние построить в произвольном месте отрезок длиной 64 единицы. Выключить режим ОРТО. Построить из произвольной точки С четыре отрезка. Отключить ОРТО. Используя привязку, первые два отрезка соединить с концами отрезка АВ, третий – перпендикуляр к прямой АВ, четвертый - соединить с серединой отрезка АВ. Чтобы не вызывать 4 раза команду Отрезок, используйте процедуру вызова предыдущей команды – нажатие пробела.

10. Построить треугольник, основание – 80, высота – 50, используя объектное отслеживание.

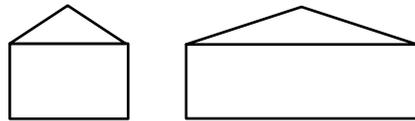


11. Построить фигуру по точкам, используя команду ОТРЕЗОК и объектное отслеживание. Для построения горизонтальных и вертикальных линий используйте метод Направление-расстояние при включенном режиме ОРТО. Для построения линий под углом используйте Полярное отслеживание или полярные координаты. Начинайте построение с т.1 по часовой стрелке.

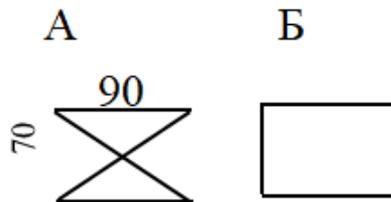


12. Нарисовать линию длиной 130 единиц и затем удлинить ее вправо на 50 единиц.

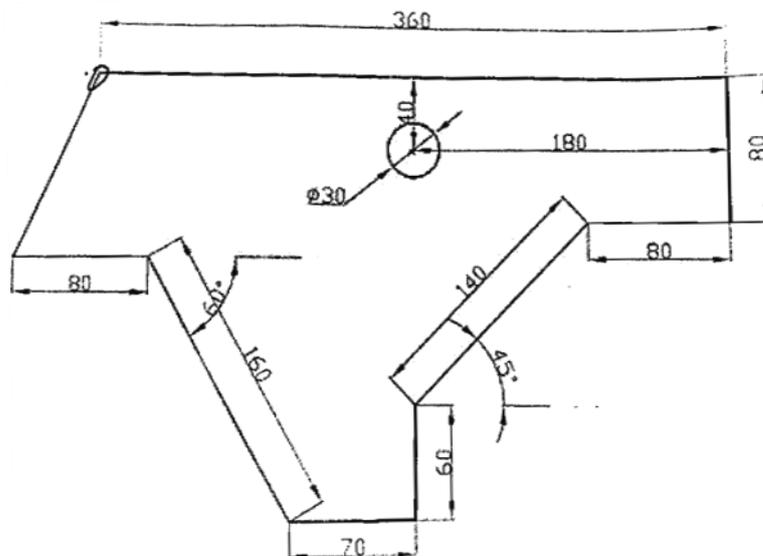
13. Нарисовать квадрат размером 80x80 и сверху треугольник высотой 40. Используя редактирование "ручками" и объектное отслеживание, растянуть рисунок вправо на 60 единиц.



14. Нарисовать фигуру А. Преобразовать ее в фигуру Б, используя редактирование "ручками".



15. Построить рисунок, используя объектную и полярную привязки.



3.10 Лабораторная работа №10 "КОМАНДЫ ПОСТРОЕНИЯ ГРАФИЧЕСКИХ ПРИМИТИВОВ"

Цель работы: изучить структуру команд AutoCAD. Научиться пользоваться командами построения отрезков, дуг, окружностей и многоугольников. Изучить команды построения и редактирования сложных примитивов. Научиться использовать в AutoCAD вставку объектов из других приложений Windows.

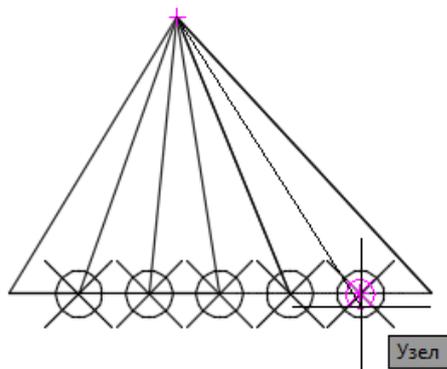
Методика выполнения работы

При выполнении работы использовать следующие темы лекционного курса:

- Режимы рисования.
- Построение графических примитивов.
- Сложные примитивы. Работа с OLE-объектами.
- Основные свойства объектов.

Задание 1

1. Построить отрезок длиной 75 единиц. Разделить его на 6 частей при помощи команды Рисование – Точка – Поделить. Изменить отображение точек. Используя привязку Узел, построить следующую фигуру.



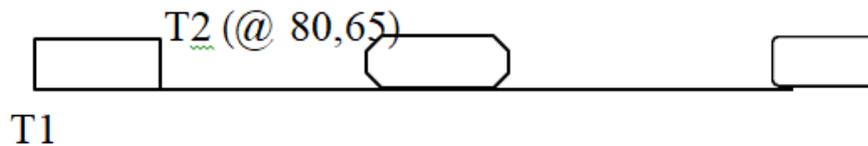
2. Построить прямые параллельно осям X и Y. Используя полярное отслеживание, построить прямые под углами 45° , 138° и 312° .

3. Нарисовать линию длиной 240 с начальной точкой T1(90,100).

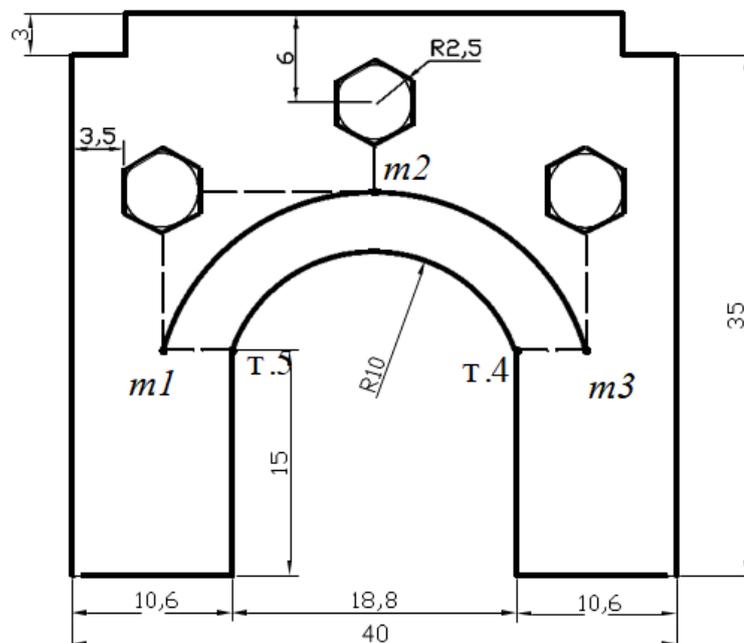
– Нарисовать прямоугольник размером 80x65, привязываясь первым углом к началу линии.

– Нарисовать прямоугольник такого же размера с фаской 10, привязываясь первым углом к середине линии.

– Нарисовать прямоугольник такого же размера с радиусом сопряжения 20, привязываясь первым углом к концу линии.



- После построения присвоить радиусу сопряжения значение 0.
- 4. Построить вписанный многоугольник с радиусом 65. Центр многоугольника - середина построенного выше прямоугольника с фаской.
- Построить описанный многоугольник с радиусом 65. Центр многоугольника тот же.
- Построить восьмиугольник, у которого длина каждой стороны равна 30.
- 5. Построить эллипс по следующим размерам: большая ось – 120 единиц, малая ось – 40 единиц.
- 6. Нарисовать чертеж. Пунктирные линии не рисовать, они указывают линии отслеживания. Многоугольник – описанный. Большую дугу построить по трем точкам: *m.1*, *m.2* и *m.3*. Эти точки определить путем отслеживания объектов. Меньшую дугу построить при помощи опции Начало, конец, радиус. Начало – *m.4*, конец – *m.5*, радиус – 10.

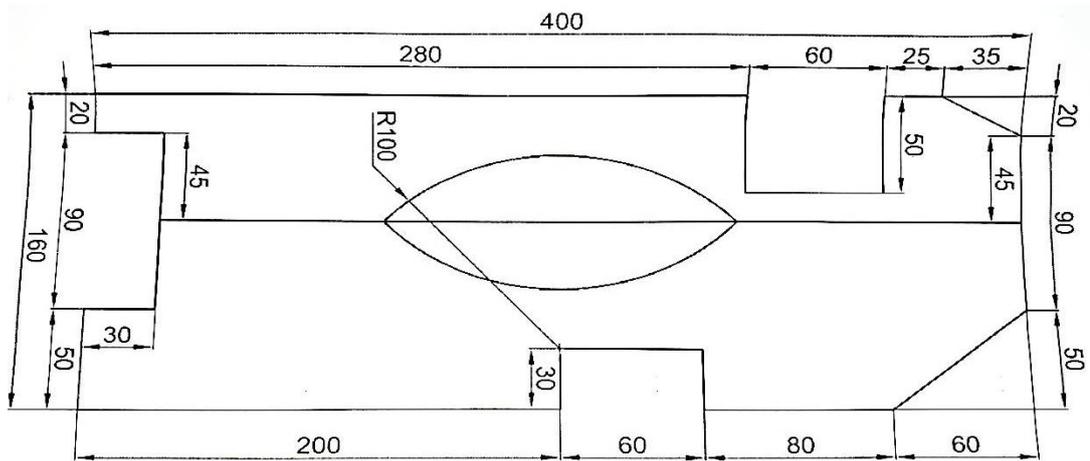


- 7. Построить окружности с диаметром 50 и радиусом 50 из одного центра – середины скругленного прямоугольника.
- Используя команду Многоугольник построить равносторонний треугольник с длиной стороны 100. Вписать в него окружность. Используя параметр 3Т, построить окружность, указывая в качестве трех точек вершины треугольника.

– Между концов двух произвольных отрезков построить окружность, используя параметр 2Г.



8. Построить рисунок, используя объектную и полярную привязки.



9. Построить полилинию произвольной длины, состоящую из 4 сегментов, изменяя начальную и конечную ширину сегмента от 0 до 10. Использовать для построения линейные и дуговые сегменты.



10. Вставить в рисунок AutoCAD формулу путем внедрения объекта Microsoft Equation.

$$H = \sqrt[3]{\left(\frac{Q_{p.n.}}{B \cdot \sigma_n \cdot \varepsilon \cdot m_0 \cdot \sqrt{2g}}\right)^2}$$

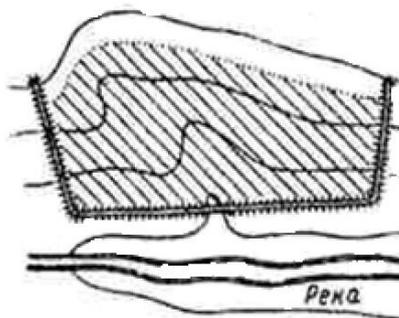
Преобразовать формулу к следующему виду:

$$H = \sqrt[3]{\left(\frac{B \cdot \sigma_n \cdot Q_{p.n.}}{\varepsilon \cdot m_0 + \sqrt{\frac{2g}{10}}}\right)^2}$$

11. Вставить в AutoCAD скопированный из папки Слайн рисунок при помощи команды Файл - Вставка, указав при этом следующие параметры:

- Точка вставки: 100,100
- Масштаб: 6

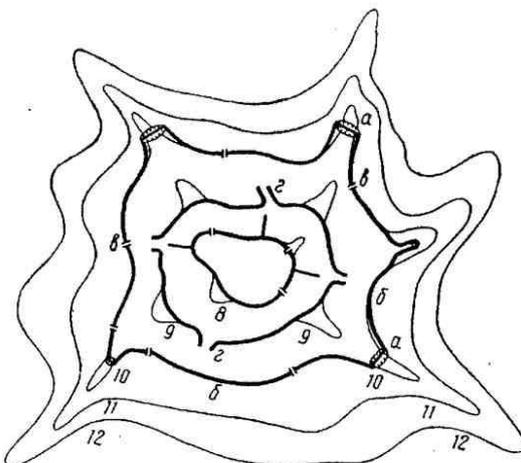
- Угол поворота: 0



Обвести все линии рисунка примитивом Полилиния. Толщина полилинии должна совпадать с толщиной линии на рисунке. Затем отредактировать полилинию при помощи опции Сплайн или Сгладить. Удалить исходный вставленный рисунок.

12. При помощи команды Вставка - Растровое изображение вставить в AutoCAD рисунок из папки Горизонтали на Рабочем столе со следующими параметрами:

- Точка вставки: 250,200
- Масштаб: 2
- Угол поворота: 0



При помощи команды Сплайн обвести все горизонтали. Отредактировать при помощи ручек полученный рисунок. Удалить исходный вставленный рисунок

13. Любым способом вставить в AutoCAD заданное изображение, находящееся в папке Сплайн на Рабочем столе. Используя команды Полилиния и Сплайн, преобразовать изображение в рисунок AutoCAD, игнорируя текст.



3.11 Лабораторная работа №11 "КОМАНДЫ РЕДАКТИРОВАНИЯ"

Цель работы: изучить команды изменения положения и формы объектов, копирования и удаления объектов AutoCAD. Научиться использовать рациональные способы построения и редактирования объектов.

Методика выполнения работы

При выполнении работы использовать следующие темы лекционного курса:

- Изменения положения объектов.
- Построение подобного объекта.
- Копирование набора объектов массивом.
- Пропорциональное изменение размеров объектов чертежа.

Задание 1

1. Построить отрезок длиной 60 единиц (координаты первой точки – 30,40).

- Построить в начале этого отрезка наклонную линию длиной 10 и под углом 225° .

- Скопировать наклонную линию на расстоянии 5 единиц вправо 2 раза, включив *Ортогональный* режим.

- Затем, выделив три наклонные линии, скопировать их за указанную на рисунке базовую точку к середине и концу построенного горизонтального отрезка.

- Исходный отрезок удлинить при помощи "ручки" до последней наклонной линии.



2. Нарисовать прямоугольник размером 60x100, используя команду "Прямоугольник". Координаты первого угла -50,80, в качестве координат второго угла указать его размер.

- Переместить прямоугольник вправо на расстояние 100 единиц.

– Переместить полученный прямоугольник, приняв за базовую точку середину нижней грани прямоугольника, к середине горизонтальной линии, нарисованной выше.



3. Нарисовать равносторонний треугольник (длина стороны – 60), используя опцию "Сторона" команды "Многоугольник".

– Описать вокруг него и вписать в него окружность, используя параметр $3T$.

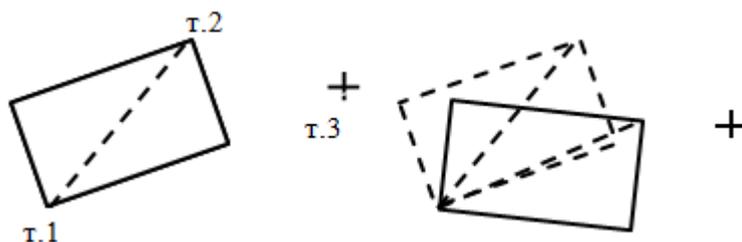
– Осуществить поворот всего объекта относительно базовой точки на угол 38° .



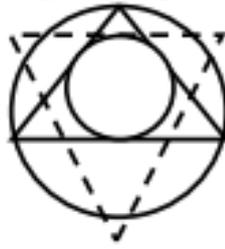
4. Совместить диагонали прямоугольника при повороте его вокруг одной из вершин с направлением к точке т.3

Команда: `_rotate`

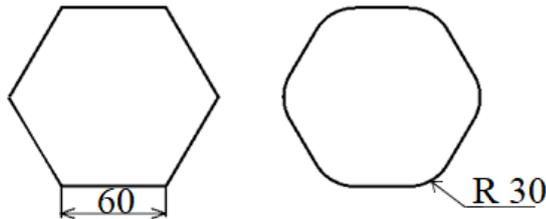
- Выберите объекты: выбрать прямоугольник и нажать ENTER.
- Базовая точка: указать точку т.1
- Угол поворота или [Копия/Опорный угол] <0>: O
- Опорный угол: указать точку т.1, а затем точку т.2
- Новый угол: указать точку т.3



5. Скопировать построенную выше фигуру. Нарисовать произвольный треугольник. При помощи команды Обрезать удалить те части объекта, которые выступают за режущие кромки – пунктирный треугольник.



6. Дан шестиугольник с длиной стороны 60. Выполнить сопряжение его вершин радиусом 30 единиц.



7. Построить эллипс, у которого большая ось - 86, малая - 40, и внутри его небольшой отрезок под углом 25° . При помощи команды Удлинить продлить концы отрезка до пересечения с эллипсом.

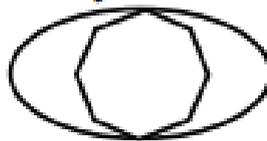
До удлинения



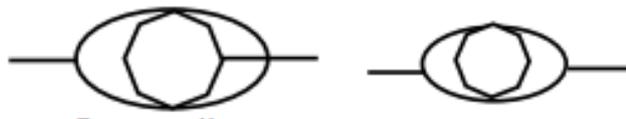
После удлинения



8. Построить вписанный восьмиугольник, указав мышкой в качестве центра - центр построенного выше эллипса; в качестве радиуса - половину малой оси эллипса.



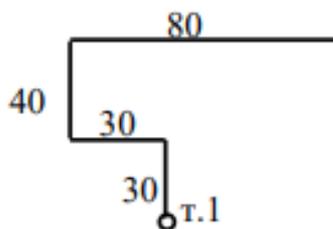
- Скопировать построенную фигуру вправо на 200 единиц.
- При помощи команды Масштаб уменьшить исходную фигуру в 2 раза.
- К скопированной фигуре добавить два отрезка следующим образом:



- При помощи опции Опорный отрезок команды Масштаб уменьшить эллипс и восьмиугольник таким образом, чтобы эллипс находился между концами двух отрезков.

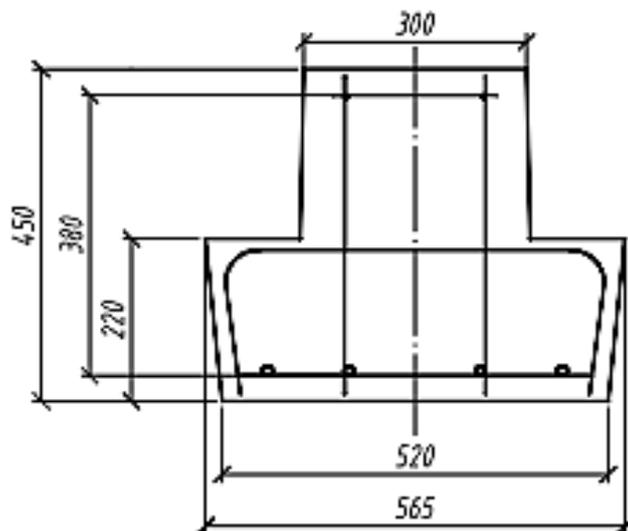
9. Построить по заданным размерам следующую фигуру. Отразить

ее зеркально сначала сверху вниз, затем слева направо.



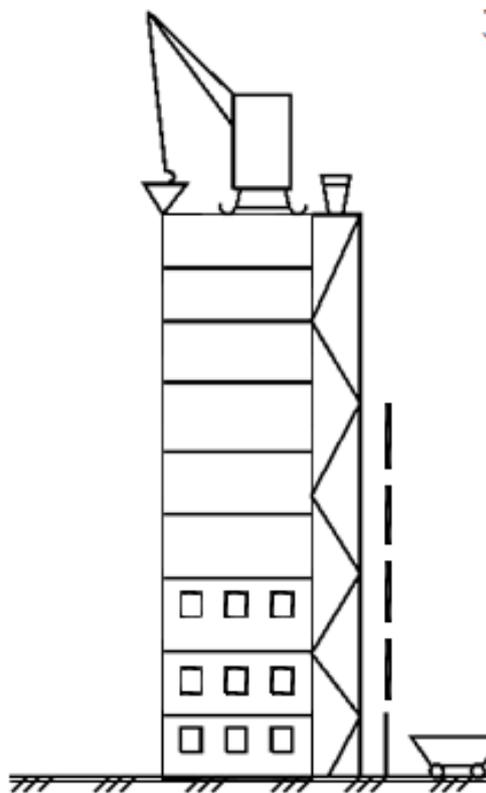
- Полученную фигуру отразить зеркально под углом 45° , указав первую точку оси отражения на расстоянии 40 единиц от нижнего правого угла фигуры. Для указания угла использовать режимы Полярное отслеживание либо Динамический ввод.

10. Построить половину заданной фигуры, не проставляя размеров. При помощи команды Зеркало построить вторую половину фигуры.



11. Нарисовать заданный рисунок в произвольном масштабе.

- Изменить размер всего рисунка при помощи опции Опорный отрезок команды Масштаб таким образом, чтобы высота крана совпадала с длиной пунктирной линии. В данном случае Опорный отрезок – высота крана, Новая длина – пунктирная линия.



12. Построить отрезок длиной 80. При помощи команды ПОДОБИЕ построить отрезки, находящиеся друг от друга на следующем расстоянии: 20, 10, 30.

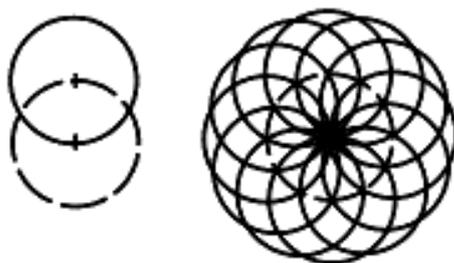
- Построить окружность диаметром 60. Применяв команду ПОДОБИЕ, построить окружность больше заданной на 20 единиц, и окружность меньше заданной на 20 единиц. Пунктирной линией указаны исходные объекты.



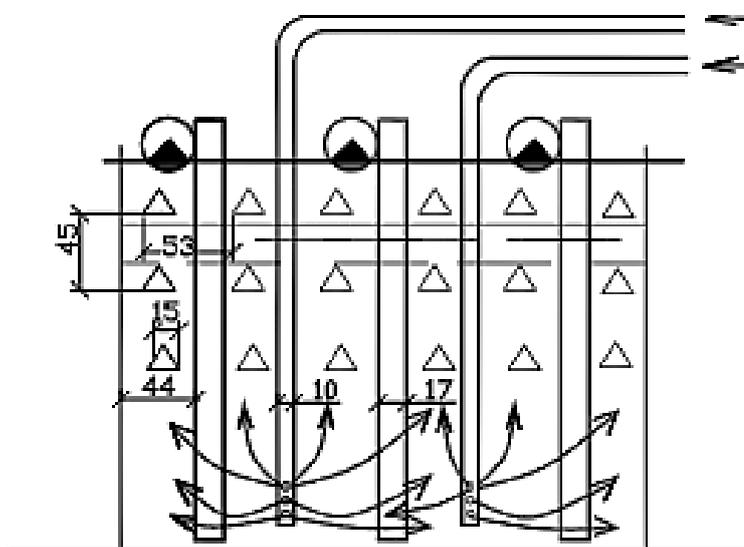
13. Построить окружность с радиусом 50 (указана на рисунке пунктирной линией).

- Скопировать полученную окружность вверх, как показано на рисунке.

- Размножить верхнюю окружность по кругу в количестве 12 единиц, указав в качестве центра массива центр нижней окружности.



14. Нарисовать следующую схему, в которой треугольник построить при помощи команды Многоугольник, опция Сторона. Треугольники размножить, используя команду Прямоугольный массив. Расстояние между столбцами и строками указано на рисунке. Использовать команду Сопряжение для изогнутых труб.



3.12 Лабораторная работа №12 "РАБОТА С ТЕКСТОМ. СЛОИ. СВОЙСТВА ОБЪЕКТА"

Цель работы: ознакомиться с видами текстовой информации, способами создания и редактирования текста в Autocad. Научиться создавать и редактировать текстовые стили и слои, изменять свойства объекта.

Методика выполнения работы

При выполнении работы использовать следующие темы лекционного курса:

- Создание однострочного и многострочного текста в AUTOCAD.
- Настройка текстового стиля.
- Свойства объекта.
- Диспетчер слоев.

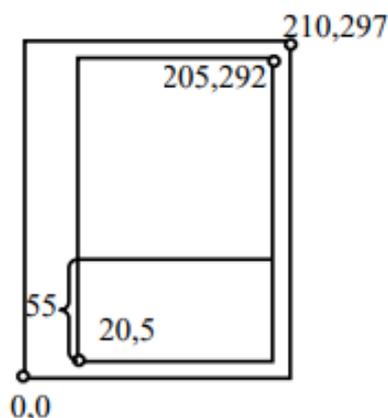
Задание 1

1. Создать новый файл. Открыть Диспетчер свойств слоев и установить следующие слои:

Имя	Цвет	Тип	Вес
Основная	Черный(белый)	Continuous	0.9
Тонкая	Голубой	Continuous	0.5
Осевая	Красный	Осевая	0.5
Текст	Синий	Continuous	0.3
Штриховка	Зеленый	Continuous	0.3
Пунктирная	Фиолетовый	Штриховая	0.5

2. Задать границы чертежа формата А4 (210x297) командой Формат-Лимиты.

3. При помощи команды ПРЯМОУГОЛЬНИК построить рамку. Координаты углов прямоугольника указаны на рисунке.



– Построить отрезок на расстоянии 55 мм параллельно нижней границе внутренней рамки.

– При помощи команды Прямоугольный массив создать 11 копий построенного отрезка, расположенных на расстоянии 5 мм друг от друга.

– Построить один вертикальный отрезок на расстоянии 7 единиц от внутренней рамки. Вертикальные отрезки штампа расположены на разном расстоянии друг от друга, поэтому лучше воспользоваться командой ПОДОБИЕ с опцией Через.

– Отредактировать полученное изображение штампа при помощи команд Обрезать, Разорвать в точке, а также редактирование ручками.

– В соответствии со стандартом выделить на чертеже тонкие линии и присвоить им слой Тонкая, а толстые линии отнести к слою Основная

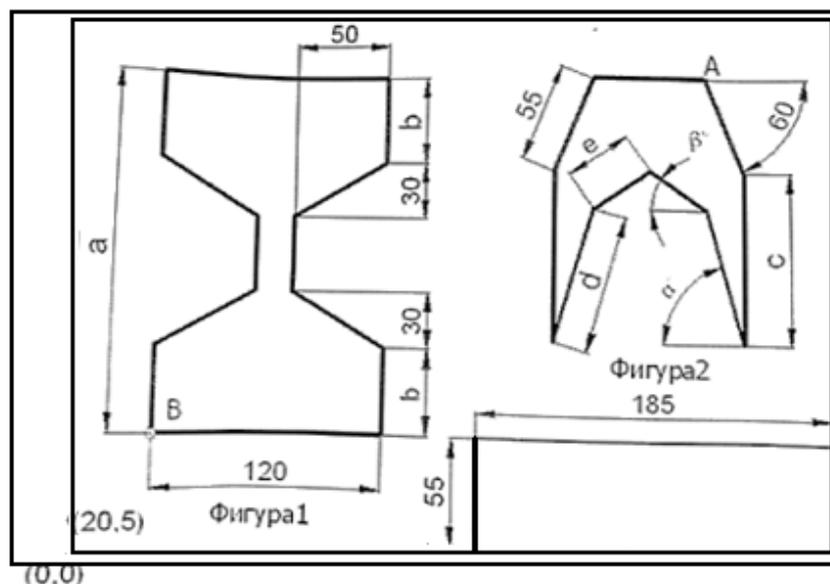
Задание 1

1. Открыть файл А4_Фамилия. Преобразовать его в формат А3 - альбомная.

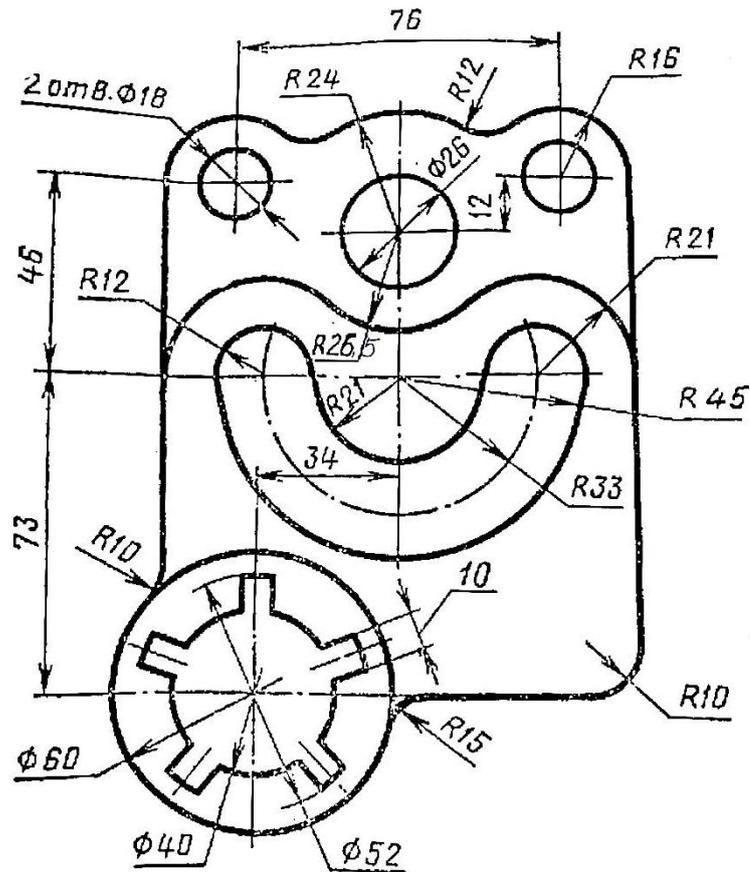
2. Нарисовать чертеж, используя данные согласно варианту. Проставить размеры. Первую фигуру начинать рисовать из точки В(60,40), вторую – из А(350,250).

Варианты заданий

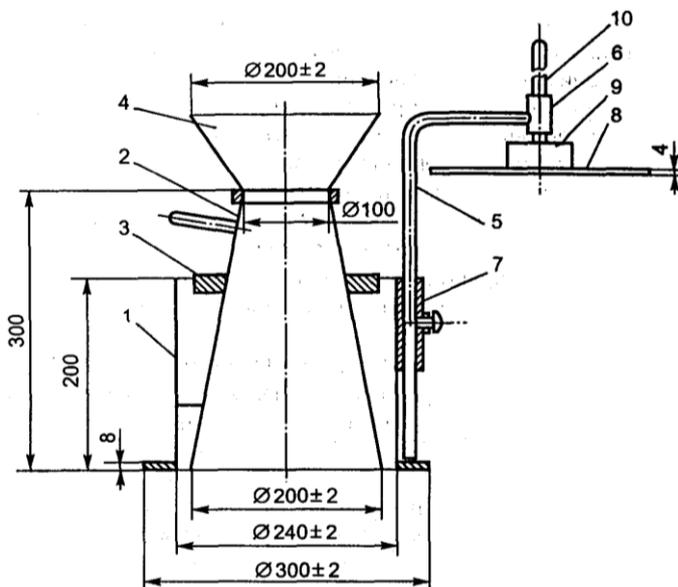
Размер	Вариант									
	1	2	3	4	5	6	7	8	9	10
а, мм	190	200	220	180	190	185	200	195	210	200
b, мм	45	40	65	30	35	45	55	55	60	20
с, мм	90	100	110	120	130	95	105	115	125	135
d, мм	73	80	85	90	95	75	83	95	98	105
угол α , градусы	74	75	78	78	80	74	75	78	75	80
е, мм	36	40	42	45	40	35	40	45	42	45
угол β , градусы	34	35	38	38	45	35	35	40	40	45



3. Нарисовать чертеж. Создать новый размерный стиль, в котором у размеров вместо стрелок поставить засечки. В этом же размерном стиле установить подразмерный стиль для радиусов и диаметров со стрелками и полочкой.



4. Нарисовать схему Технического вискозиметра по слоям, про-
ставить размеры.
- При помощи команды Масштаб привести чертеж к указанному
в варианте масштабу.
 - Создать и применить новый размерный стиль.
 - Разместить полученный чертеж на листе формата А4_Фамилия
и выдать на печать с опцией Просмотр.



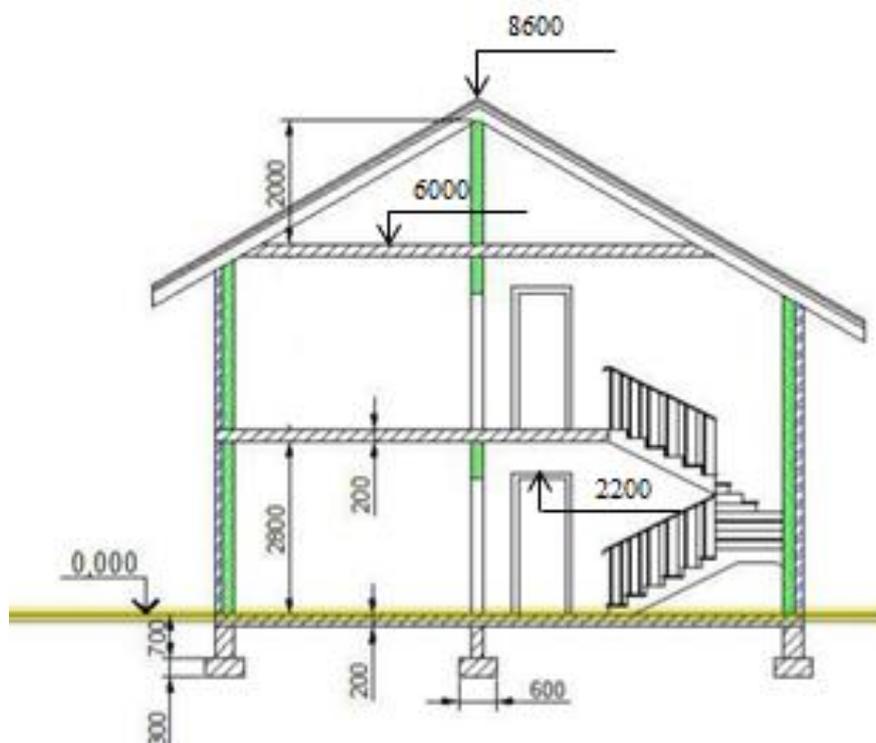
Вариант №1 – 1:2

Вариант №2 – 1:10

Вариант №3 – 1:5

Вариант №4 – 2:1

5. Создать новый стиль мультивыноски, рассмотренный в лекции. Нарисовать чертеж и проставить размеры вместе с выноской.



3.14 Лабораторная работа №14 "ШТРИХОВКА. МУЛЬТИЛИНИИ"

Цель работы: научиться использовать расширенные возможности нанесения штриховки, ознакомиться с различными способами создания и редактирования мультилинии в AutoCAD, рассмотреть определения стилей мультилинии. Изучить состав диалогового окна Штриховка и градиент. Научиться копировать фрагменты чертежа, используя Центр управления.

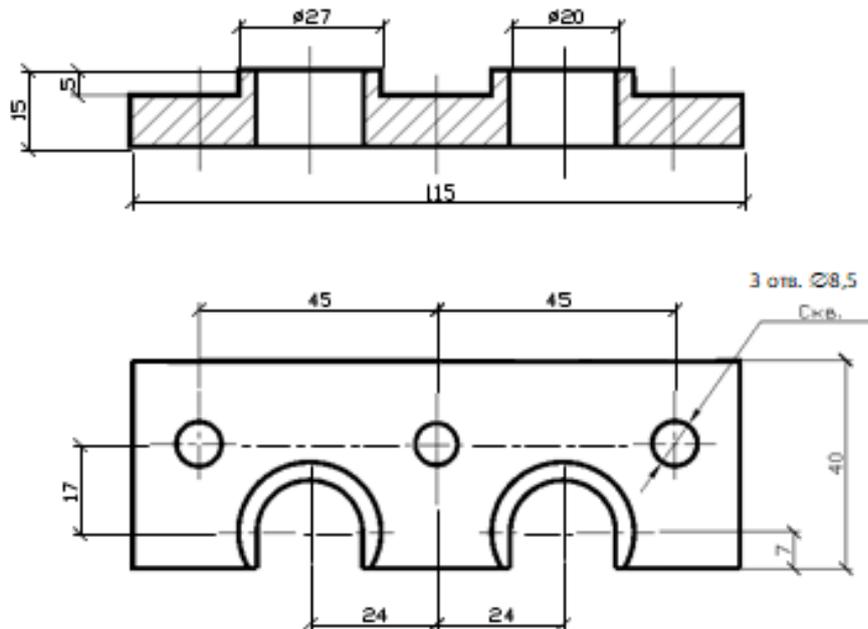
Методика выполнения работы

При выполнении работы использовать следующие темы лекционного курса:

- Штриховка и градиент. Способы указания области для нанесения штриховки.
- Копирование слоев и размерного стиля через Центр управления.
- Свойства объекта.
- Создание новой панели на ленте.
- Создание и редактирование мультилинии.

Задание 1

1. Создать новый файл. При помощи команды Центр управления перенести в этот файл слои и свой размерный стиль.
2. Нарисовать следующий чертеж по слоям.



3. Выполнить следующие задания построения штриховки. Основные примитивы чертить в основном слое, штриховку – в слое штриховка.

– Нарисовать в осевом слое отрезок длиной 10 единиц с координатами в начальной точке (20,20). При помощи команды Свойства изменить масштаб этого отрезка так, чтобы были видны штрихи в осевой линии.

– Нарисовать отрезок длиной 61 единица с координатами в начальной точке (50,50). Нарисовать окружность по конечным точкам отрезка, используя соответствующую опцию, после построения удалить отрезок. При помощи команды Массив размножить окружность в 2 столбца и 2 строки, расстояние между которыми 140 единиц. Заполнить окружности штриховкой:

- первую окружность штриховкой ANSI35;
- вторую окружность этой же штриховкой, но увеличить масштаб в 2 раза;
- третью окружность штриховкой ISO10W100 с углом поворота 10° .
- первую окружность штриховкой ARSAND, масштаб выбрать самостоятельно.

– Отредактировать штриховку первой окружности, изменив угол поворота с 0° на 45° .

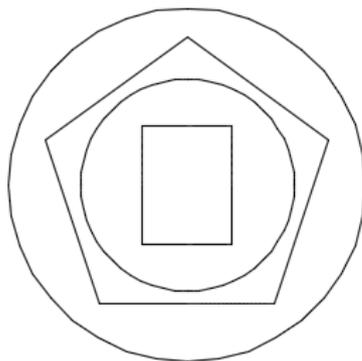
– Выделить все объекты и при помощи команды Свойства сделать все контуры окружностей желтого цвета.

– Нарисовать окружность радиусом 60 в произвольном месте.

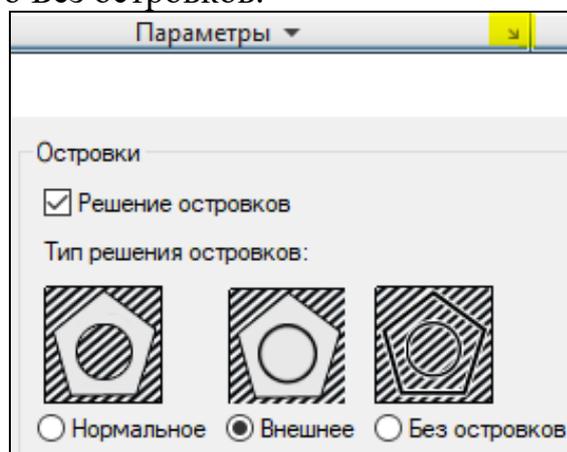
При помощи команды Многоугольник нарисовать вписанный пятиугольник радиусом 50.

При помощи команды Подобие сместить исходную окружность внутрь на расстояние 24.

Построить прямоугольник 30x40 и переместить внутрь окружности.



- Размножить полученный рисунок вниз 3 раза на расстоянии 130 единиц.
- Используя параметры штриховки Островки заполнить штриховкой Gravel полученные рисунки следующими способами:
Первый рисунок – способ Нормальный;
Второй – способ Внешний;
Третий – способ Без островков.



- С помощью команды Слайн построить рядом две замкнутые фигуры произвольного контура. Заполнить их штриховкой "по бетону", при этом к первой фигуре применить ассоциативную штриховку, ко второй - не ассоциативную.

Задание 2

1. Создать новую панель МУЛЬТИ с командами Мультилиния, Диспетчер стиля мультилинии, Мультивыноски и Диспетчер стиля мультивыноски.
2. Создать новый слой MULTI, в котором установить тип линий – пунктирный, цвет – синий, вес – по умолчанию. В этом слое нарисовать горизонтальный отрезок длиной 60 единиц, начальная точка A(60, 34.5). Размножить созданный отрезок 4 раза с помощью массива вниз на расстоянии 18 единиц.
3. Переключиться на нулевой слой. Создать четыре мультилинии с разными параметрами (меняя их в командной строке), привязываясь при по-

строении к концу каждой синей линии, длиной 60 единиц:

- расположение **Центр**, Масштаб **5**;
- расположение **Верх**, Масштаб **0**;
- расположение **Низ**, Масштаб **-2.5**;
- расположение **Верх**, Масштаб **3**.

Начальная точка или [Расположение/Масштаб/Стиль]: М

МЛИНИЯ Масштаб мультилинии <20.00>: 5

4. Создать четыре новых стиля мультилиний, набрав в командной строке `_mlstyle`:

- **СТИЛЬ0**: три пунктирные линии, удаленные друг от друга на расстояние 1 единицы. В свойствах мультилинии установить флажок Показывать стыки.

- **СТИЛЬ1**: две линии синего цвета, удаленные друг от друга на расстояние 8 единиц, с линией изоляции между ними (тип – изоляция, цвет – красный). В поле Торцы установить флажки: начало – отрезок, угол 90 и конец – отрезок, угол 45.

Торцы		Элементы	
	Начало	Конец	
Отрезок:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Смещение
Внешняя дуга:	<input type="checkbox"/>	<input type="checkbox"/>	Цвет
Внутренние дуги:	<input type="checkbox"/>	<input type="checkbox"/>	Тип линий
Угол:	90.00	45.00	4
			0
			-4
			Смещение
			Цвет
			Тип линий

- **СТИЛЬ2**: четыре линии разного цвета и разных произвольных типов, удаленные друг от друга на расстояние 1 единицы, без стыков. В поле Торцы установить флажки: начало – внешняя дуга, угол 90 и конец – внутренняя дуга, угол 20.

- **СТИЛЬ3**: первая линия: смещение - 0, тип – непрерывный, цвет – черный; вторая линия: смещение - 0, тип – скрытый, цвет – голубой.

Нарисовать свое имя командой Мультилиния, используя для каждой буквы разные стили, созданные выше.

5. Нарисовать план квартиры в масштабе 1:100. Основная стена – 510 мм, перегородка – 250 мм.

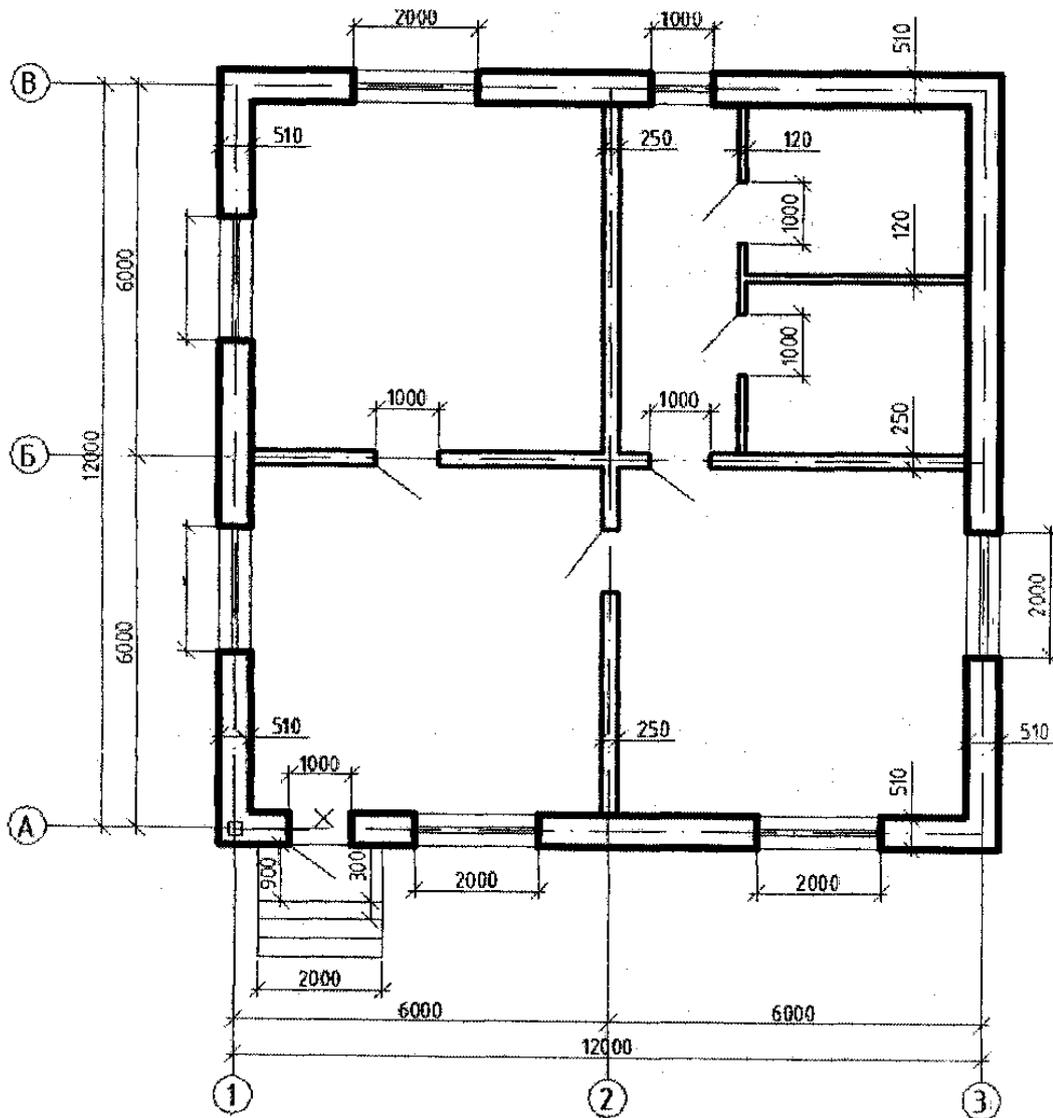
– Вначале создать новый стиль мультитинии (на базе стиля STANDARD) из трех элементов (средняя – осевая), название стиля – фамилия студента.

– Нарисовать основные стенки (расположение – центр). Затем аналогично - перегородки.

– Отредактировать стыки и пересечения. В местах расположения дверей разорвать мультитинию. Окна вставить позже.

– Проставить размеры от осевой линии.

– Вычислить площадь каждой комнаты и вставить с помощью Поля.



3.15 Лабораторная работа №15 "ПОСТРОЕНИЕ СЛОЖНЫХ ОБЪЕКТОВ. ТАБЛИЦА И ВИДОВЫЕ ЭКРАНЫ"

Цель работы: научиться создавать и редактировать таблицы, владеть навыками создания и подрезки видовых экранов в пространстве Листа, вставлять чертежи в другие приложения.

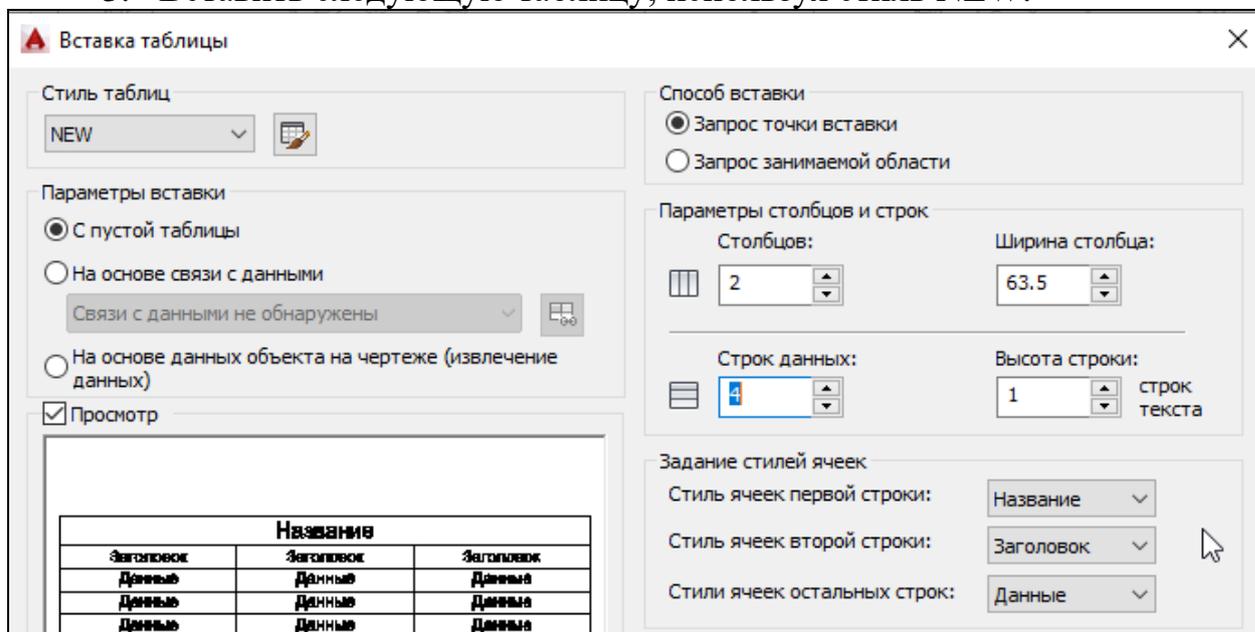
Методика выполнения работы

При выполнении работы использовать следующие темы лекционного курса:

- Создание и редактирование таблицы.
- Видовые экраны модели и листа.
- Работа с OLE-объектами.

Задание 1

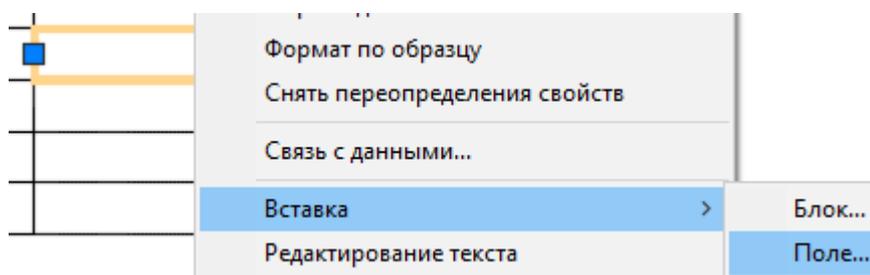
1. Создать новый стиль таблицы NEW, у которого изменить данные:
 - наименование таблицы: шрифт 26 пт, Arial, цвет фона – салатовый, букв – бордовый;
 - заголовки столбцов: шрифт 20 пт, Arial, цвет фона – желтый, букв – синий;
 - данные: шрифт 14 пт, Times New Roman, цвет фона – нет, букв – черный;
2. Нарисовать семиугольник радиусом 115.
3. Вставить следующую таблицу, используя стиль NEW.



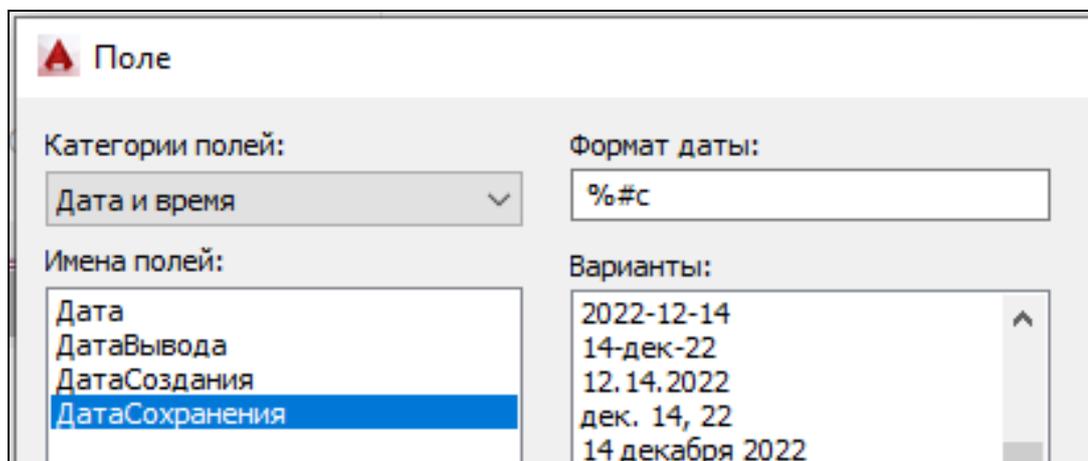
Данные о чертеже

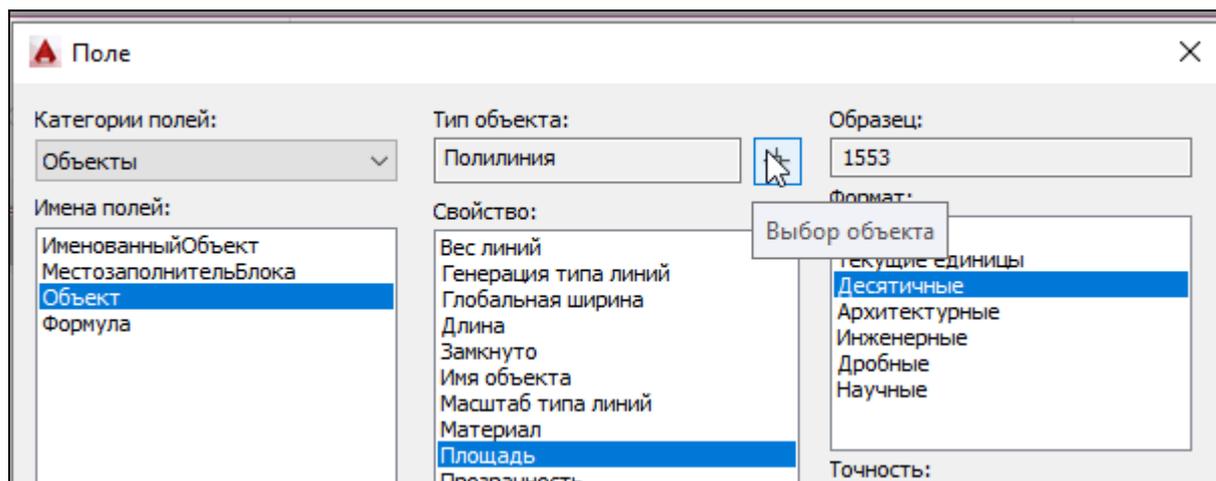
Наименование	Категория
Имя файла (полный путь)	Документ
Текущая дата	Дата сохранения
Текущее время	Время
Площадь	Объект

- Наименование таблицы и заголовков, а также данные столбца Наименование набрать на клавиатуре
- Данные в столбец Категория вставить, щелкнув правой мышкой и выбрав в контекстном меню Вставить поле.



- В верхнем поле Категория полей выбрать соответствующую категорию.
- В поле Имена полей выбрать соответствующее наименование.





– Вычислить площадь нарисованного выше семиугольника, убрав в Формате хвостовые нули.

– Изменить контур семиугольника и для перерасчета формул ввести в командной строке команду РЕГЕН.

Задание 2

1. Создать таблицу, указанную в задании. В столбце Стоимость использовать формулу.

2. Щелкнув по ячейке, выбрать в контекстном меню Выбрать поле. Затем установить:

- Категория - Объекты
- Имена полей - Формула
- Формат - Десятичные
- Формат поля - убрать хвостовые нули
- Точность - три знака после запятой
- В поле Формула поставить знак =, затем щелкнуть по кнопке Ячейка

и выбрать в таблице соответствующую ячейку. В конце нажать кнопку Вычислить.

- Выделить ячейку с результатом в первой строке, нажать CTRL+C.

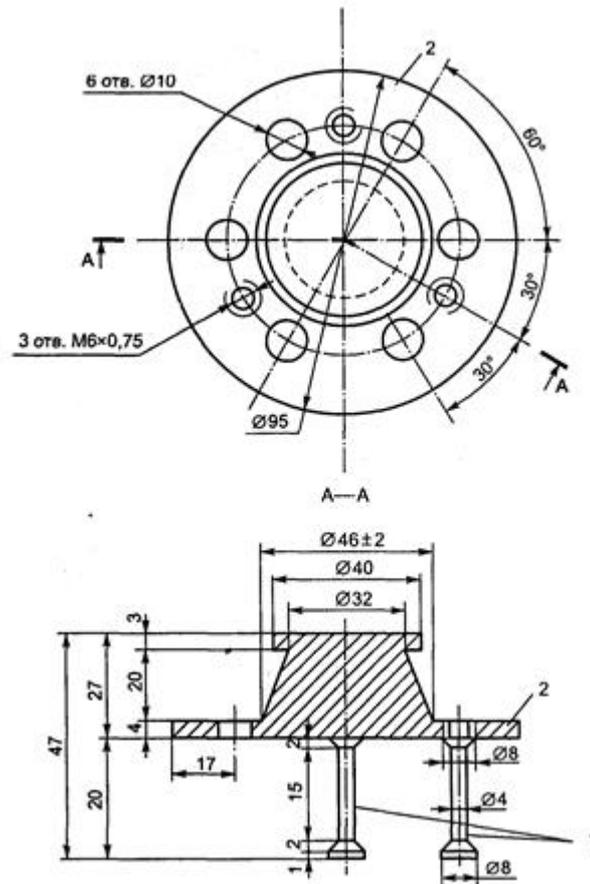
– Выделить диапазон следующих вычисляемых ячеек и нажать CTRL+V.

Наименование	Един. измер.	Цена, руб.	Норма расхода		Стоимость, руб.	
			Заводская	Проектная	Заводская	Проектная
Цемент М500	т/м ³	96800	0.39	0,32		
Песок	м ³ /м ³	10615	0.42	0,45		
Щебень	м ³ /м ³	1/856	0.7	0.67		
Вода	м ³ /м ³	1071	0.2	0.162		

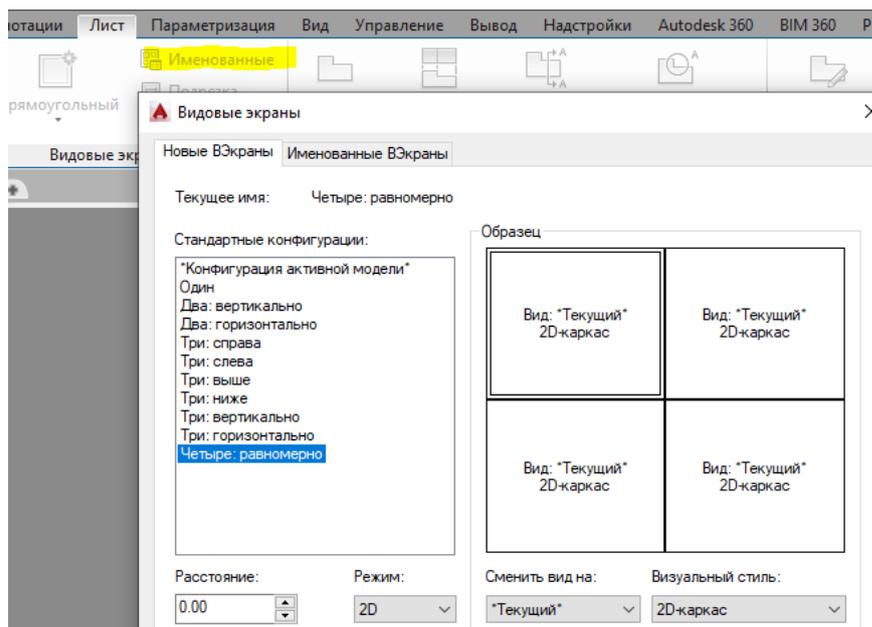
Добавка	т/м ³	670000	0	0.00192		
Сталь	т/м ³	935000	0.054	0.054		
ИТОГО:						
Тепловая энергия	Г кал/м ³	48000	0.2	0.072		
Электроэнергия	кВт/ч	105	19.9	18		

Задание 3

1. Нарисовать заданный чертеж, используя слои и свой размерный стиль в натуральную величину. Размеры проставлять в слое Текст.



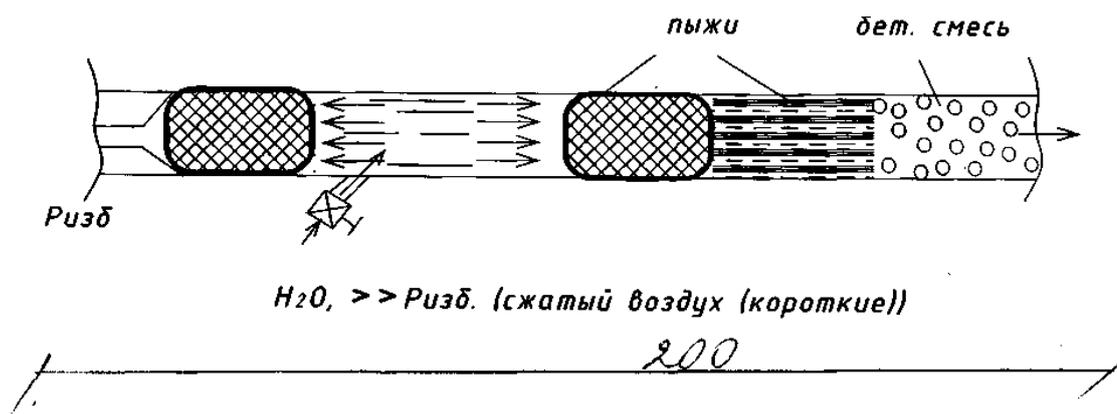
2. Перейти в пространство Листа и удалить текущий ВЭ. Создать 4 ВЭ в пространстве листа, используя параметр Вписать.



3. Создать новый слой Граница и перенести на этот слой контуры ВЭ.
4. На 1 ВЭ с помощью команды Текст вставить название "Прибор Красного".
5. Только на 1 ВЭ оставить размеры, заморозив слой Текст на остальных ВЭ, используя кнопку Заморозить на текущем ВЭ.
6. На 2 ВЭ уменьшить текущий масштаб чертежа в 2 раза.
7. На 3 ВЭ подрезать чертеж многоугольником, выбрав (оставив на экране) нижний вид.
8. На 4 ВЭ нарисовать "облако" на верхнем виде, и подрезать ВЭ объектом, используя в качестве подрезающего объекта "облако".
9. Сделать невидимыми границы всех видовых экранов.
10. Отправить лист на печать, используя опцию Предварительный просмотр.

Задание 4

1. Открыть новый файл. Перенести туда свои слои и свой размерный стиль через центр управления.
2. Нарисовать заданную схему по слоям.



3. Вставить этот фрагмент чертежа в Word двумя способами.

1 Способ

- Выделить в AutoCAD фрагмент чертежа и выполнить команду Правка - Копировать.
- Перейти в Word и вставить скопированный фрагмент.
- Отредактировать его, обрезав лишнее белое поле.
- Сделать сверху рисунка надпись «Способ №1».

2 Способ

- В Word выбрать команду Вставка – Объект – Чертеж AutoCAD.
- Откроется AutoCAD с новым чертежом, который связан с Word. Скопировать в этот пустой чертеж фрагмент построенной схемы.
- Перейти на лист с единственным прямоугольным ВЭ на нем.
- Отрегулировать пропорции ВЭ таким образом, чтобы он отмечал будущие границы рисунка.
- Убрать информацию о цвете – всем слоям назначить белый цвет.
- Включить веса линий.
- Максимизировать чертеж внутри ВЭ двойным нажатием ролика мыши.
- Регенерировать чертеж (РЕГЕН).
- Закрыть чертеж. На запрос "Обновить Microsoft Word перед закрытием?" ответить "Да".

3.16 Лабораторная работа №16 "ПРОСТЫЕ И ДИНАМИЧЕСКИЕ БЛОКИ"

Цель работы: научиться создавать и редактировать простые и динамические блоки, а также блоки с атрибутами. Уметь извлекать из блоков текстовую информацию. Освоить работу с Палитрой вариаций блоков в редакторе блоков.

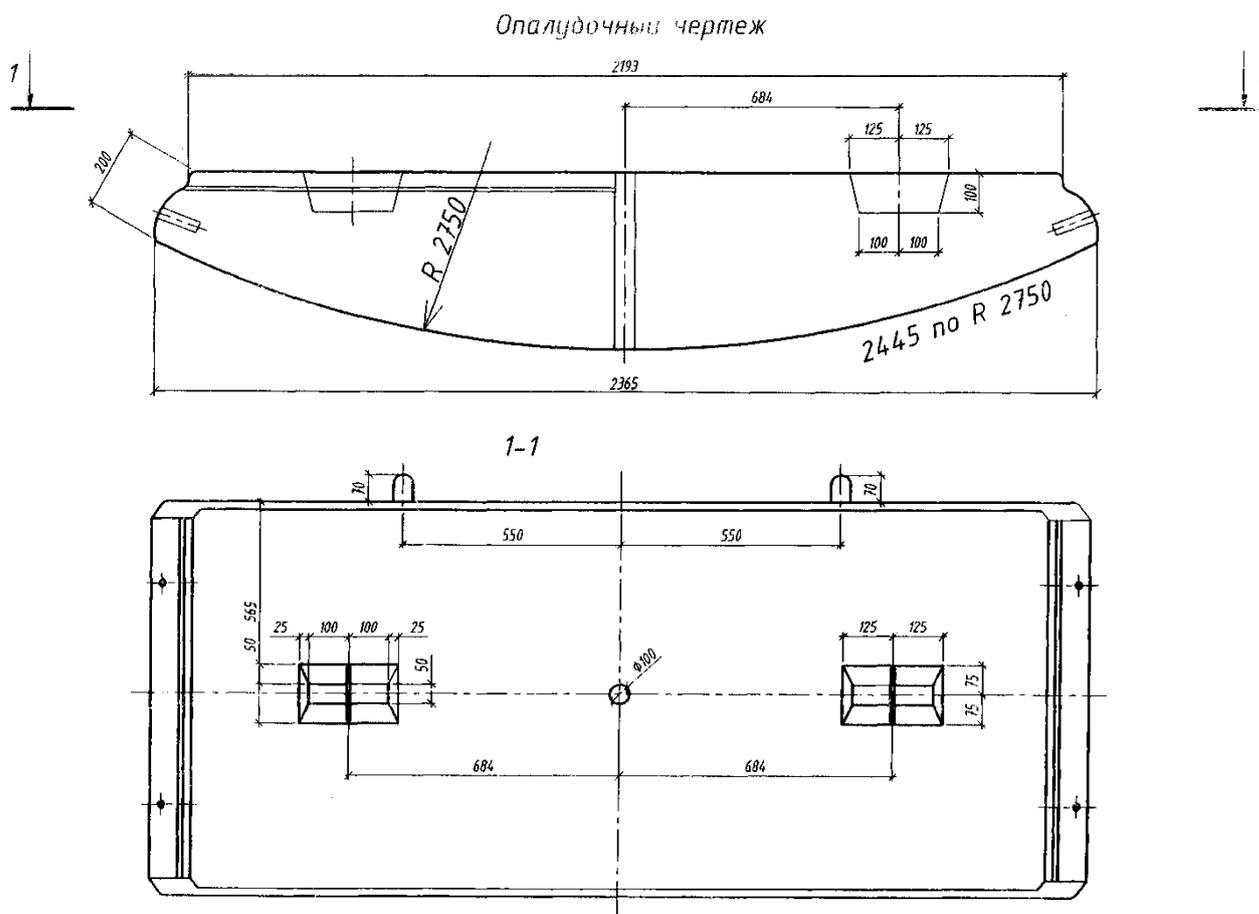
Методика выполнения работы

При выполнении работы использовать следующие темы лекционного курса:

- Создание определения блока.
- Запись блока в файл.
- Способы вставки и редактирование блока.
- Атрибуты блока.
- Динамические блоки.

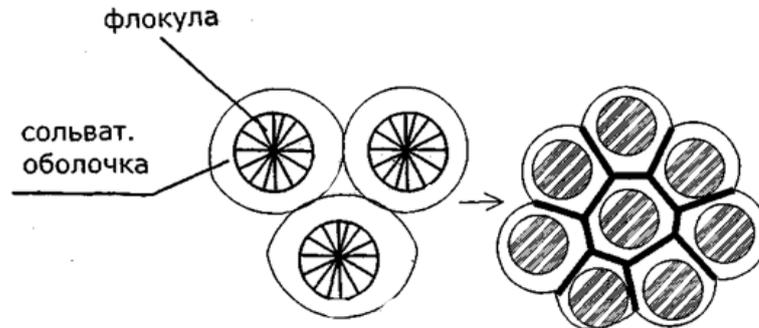
Задание 1

1. Открыть новый файл, выбрав в папке Template шаблон A4_имя.
2. Преобразовать рамку A4 в формат листа A1 (альбомный 840x594).
Сохранить полученный чертеж под именем A1.dwg.
3. Нарисовать заданный чертеж.
4. Создать из этого чертежа блок Block2, оставив исходные объекты без изменений
5. Вставить в рамку файла A1 блок Block2 3 раза с масштабами 2:1, 1:5 и 1:2. Сверху над блоками сделать надписи соответствующего масштаба
6. Зайти в редактор блоков и вставить надпись с названием конструкции в текстовом слое.



Задание 2

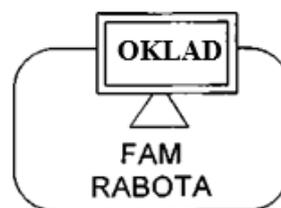
1. В файле A1.dwg создать новый слой NEW желтого цвета с типом линии «зигзаг» и сделать его текущим.
2. Открыть новый файл. Перенести туда свои слои и нарисовать по слоям следующую схему.



3. Один из контуров перенести на нулевой слой.
4. Создать из полученного чертежа автономный блок Влок3 (записать в файл) и вставить его в файл A1 со следующими параметрами:
 - с масштабом 1:2 (по X и Y одинаковые масштабы);
 - с масштабом по X (-2), а по Y(+2);
 - с масштабом по X (+3), а по Y (-2).
5. Зайти в редактор блоков и перенести контур с нулевого слоя на основной слой.

Задание 3

1. Создать в новом файле чертеж плана комнаты 6000x10000. Ширина стенки - 380, дверной проем - 900, оконный проем - 1200.
2. Вне плана комнаты нарисовать прямоугольник размером 1300x800 с закругленными углами радиусом сопряжения 200. На нем нарисовать монитор.

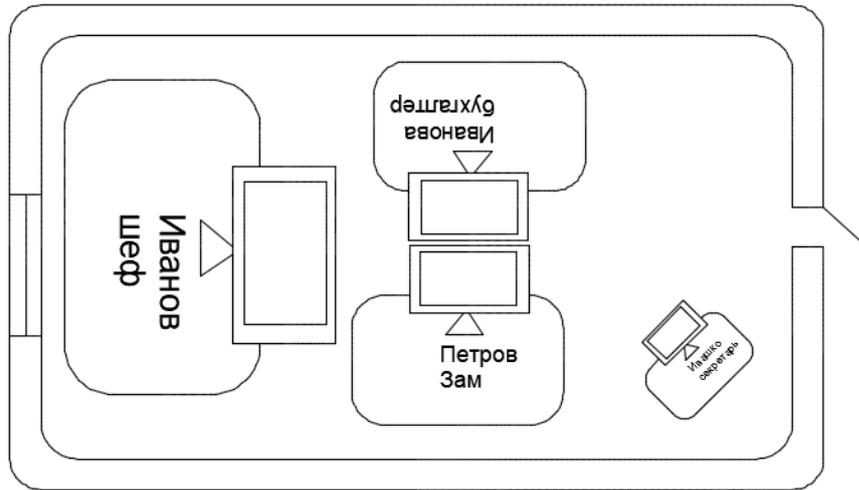


3. Создать три атрибута блока со следующими параметрами.

Режим	Имя	Подсказка	По умолчанию	Высота текста
Контролируемый	FAM	Фамилия	Иванов	100
Без режима Выровнять по предыдущему	RABOTA	Должность	Директор	100
Скрытый	OKLAD	Оклад	5000	100

4. Из полученного рисунка и атрибутов создать блок СТОЛ, оставив исходные объекты без изменений.

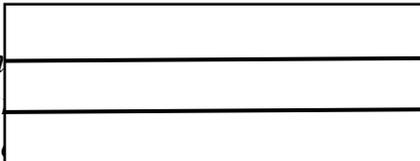
5. Вставить блок СТОЛ в комнату 4 раза с масштабами 3:1, 2:2 и 1:1, меняя должности, оклады и фамилии согласно рисунку.



6. Извлечь атрибуты всех вхождений блока в таблицу. Вставить таблицу рядом с чертежом, увеличив ее в 50 раз.

Задание 4

1. Открыть файл с планом квартиры из лабораторной работы №6.
2. Создать динамический бок Окно, в котором *высота – фиксированная, а длина будет изменяться по указанным размерам.*
3. Нарисовать прямоугольник 900x510.

4. Создать  размера и установить галочку Открыть в редакторе  - левая нижняя.

5. На панели инструментов  инструмент Маскировка, который скрывает все, что находится под замаскированным объектом. Маскировка должна находиться поверх всех объектов, кроме нашего окна.

6. После вызова команды Маскировка указать четыре точки нашего окна и нажать ВВОД. Маскировка накладывается поверх всех объектов.

7. Чтобы расположить маскировку подокном, необходимо выделить окно и задать в редактировании выбрать параметр "на задний план".

8. Вызвать команду Маскировка и включить параметр русское "к" (вКл).

9. Выставить Линейный параметр, добавить операцию Растянуть за правую нижнюю точку

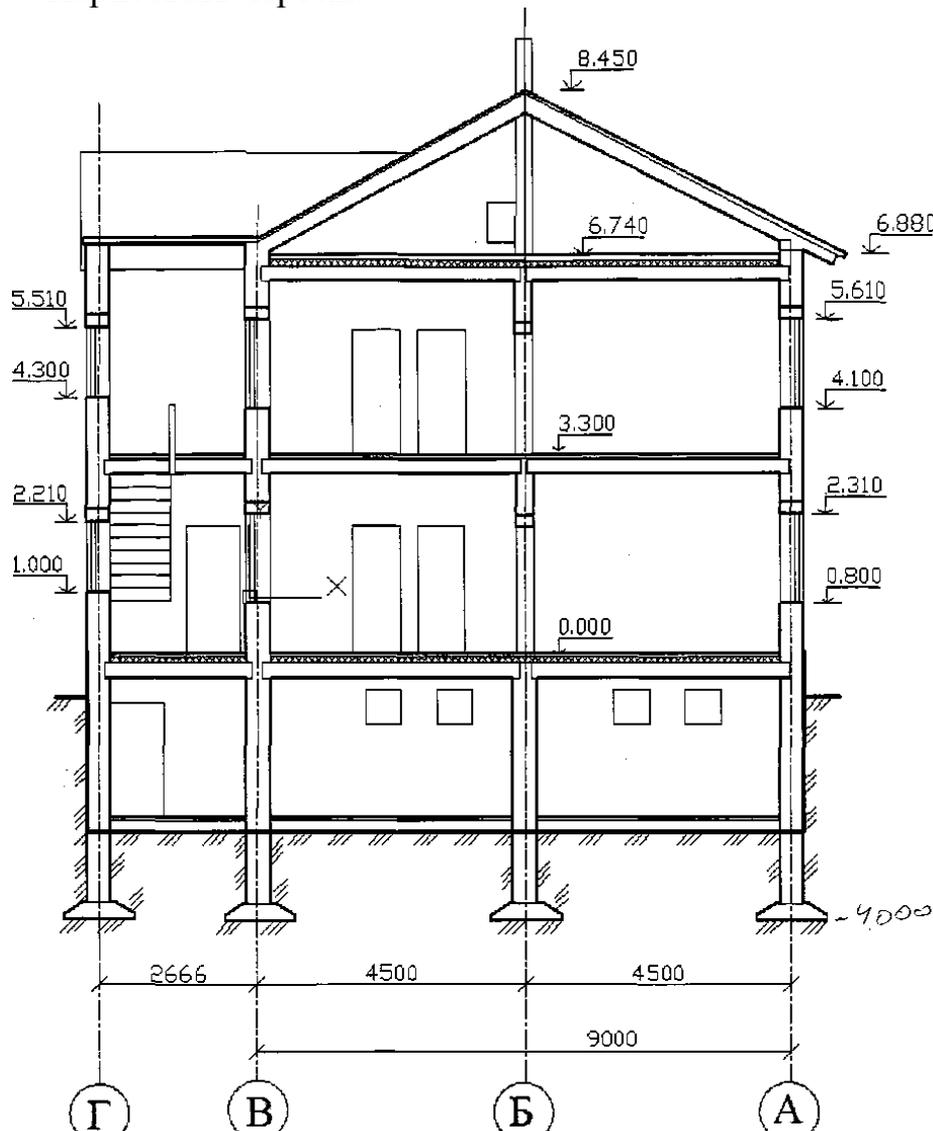
10. Вызвать команду Свойства. Выделив параметр, добавить Список значений расстояний: 1000, 2000, 2500.

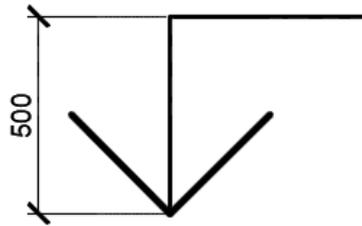
11. Выставить параметр Поворота, добавить к этому параметру операцию Повернуть за правую верхнюю точку.

12. Вызвать команду Свойства. Выделив параметр, добавить список значений угла.
13. Перенести объекты блока на слой Пунктирная и закрыть блок ОКНО.
14. Вставить в чертеж вне комнаты блок «Окно». Скопировать его три раза.
15. Каждому задать соответственно расстояния 2500, 2000, 1000.
16. Скопировать блоки на план согласно рисунку.

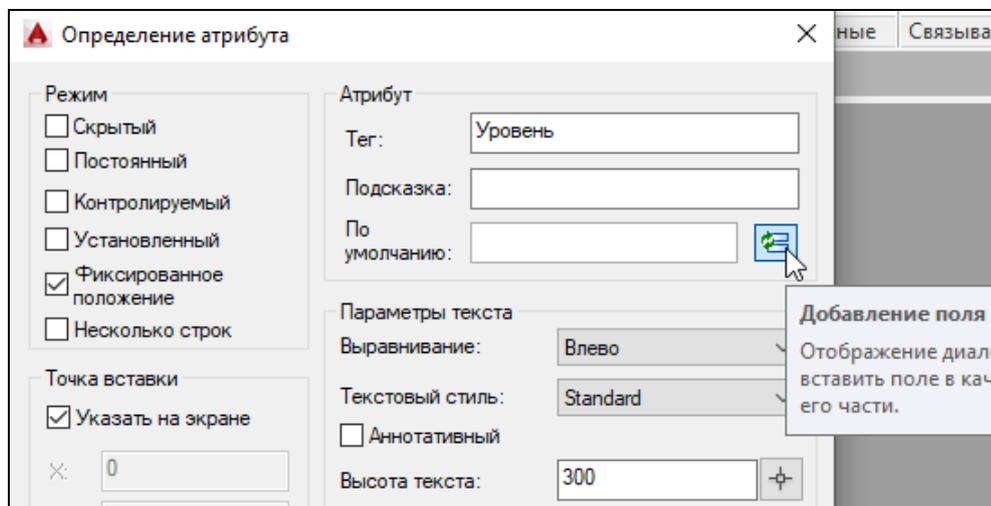
Задание 5

1. Нарисовать заданный чертеж. Проставить линейные размеры, не проставляя размер отметки уровня, но учитывать его при построении.
2. Перенести начало координат на линию, где находится нулевая отметка.
3. Нарисовать чертеж.

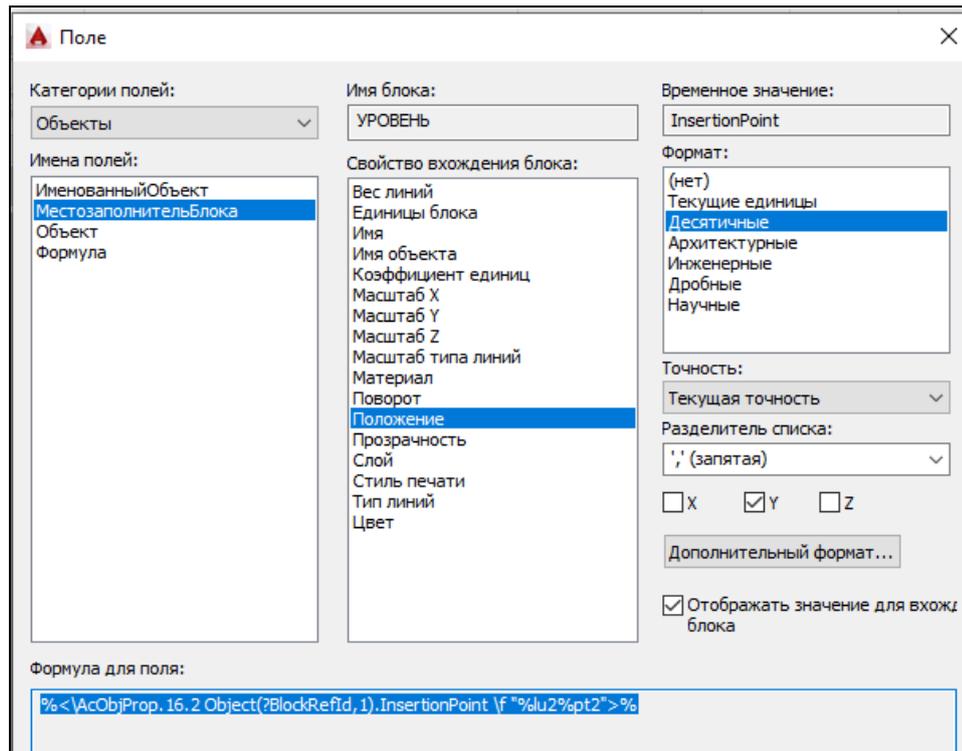




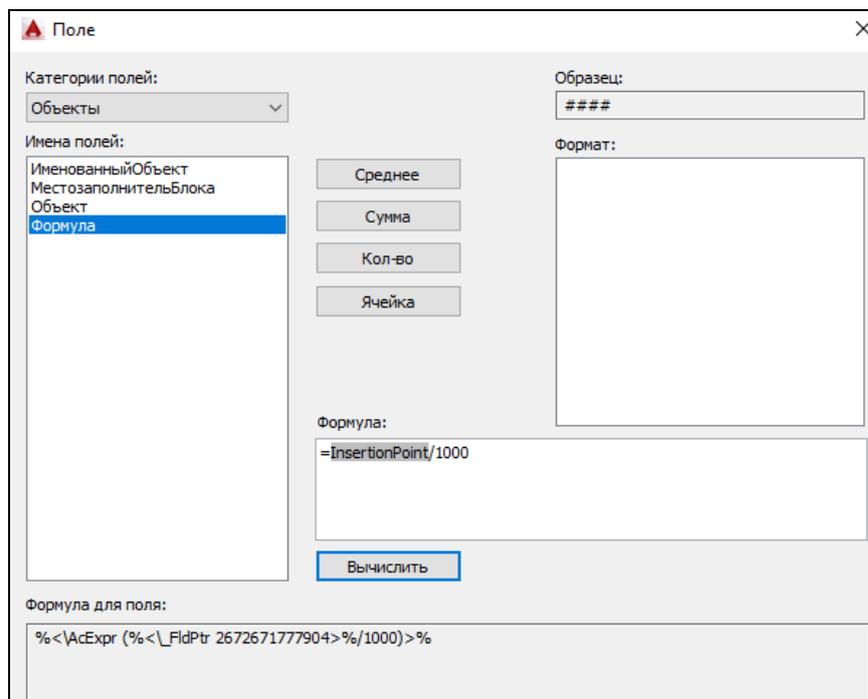
4. Создать из рисунка без размера блок УРОВЕНЬ, указав в качестве базовой точкой острие стрелки, и поставить флажок Открыть в редакторе блоков.
5. В редакторе блоков вставить атрибут блока Уровень с указанными параметрами.



6. В поле Атрибут нажать на кнопку Добавление поля и установить следующие параметры:
7. Выделить и скопировать текстовую информацию в Формуле для поля.
В поле Имена полей выбрать Формула. Ниже в поле Формула набрать "=" и вставить скопированный текст. Для перевода в метры разделить на 1000 и нажать кнопку вычислить. Установить необходимый формат



8.



9. На панели вариаций установить параметр и операцию зеркального отображения слева – направо и сверху – вниз.

4 РАЗДЕЛ 3. КОНТРОЛЬ ЗНАНИЙ

Вопросы для зачета по дисциплине "Решение технологических задач на ЭВМ":

1. Из каких компонентов состоит система программирования Visual Basic? Их назначение.
2. Опишите технологию визуального проектирования.
3. Понятие и назначение объекта.
4. Назначение элемента управления Label (Метка).
5. Назначение элемента управления TextBox (текстовое поле).
6. Что такое метод? Как метод обозначается в программном коде?
7. Что такое событие объекта? Как оно обозначается в программном коде?
8. Какие действия может совершать процедура обработки события?
8. Опишите вид окна системы программирования Visual Basic, назовите компоненты окна, как их вызвать или убрать.
9. Назначение окна Project Explorer. Что в нем отображается? Какие действия можно совершать в этом окне? Его использование для открытия форм и модулей.
10. Назначение окна Toolbox (палитра объектов). Что в нем отображается и каков порядок вставки объектов на поверхность формы?
11. Окно Properties (свойств), назначение, правила использования.
12. Как изменить значение свойства объекта в период разработки проекта и в программе?
13. Назначение окна конструктора форм. Что в нем отображается? Как управлять содержимым этого окна?
14. Окно редактирования кода. Что может содержаться в этом окне?
15. Назначение элементов управления, прототипы которых расположены на панели элементов управления при открытии нового проекта. 37. Что такое модуль? Какие виды модулей вы знаете? Как их ввести в проект?
16. Требования к идентификаторам.
17. Как объявить переменную? Типы данных.
18. Напишите операторы объявления переменных числового типа, строки переменной и фиксированной длины, пользовательского типа, массива.
19. Область видимости переменной. Объясните значения ключевых слов Private, Public и Static, используемых при объявлении переменных.
20. Назначение функции Val(...). Какой символ функция Val(...) воспринимает как разделитель дробной и целой частей числа?
21. Как ввести в программу информацию, введенную в текстовое окно TextBox? Как вывести в текстовый блок TextBox информацию из программы?
22. Назначение элемента управления CommandButton (Командная кнопка).
23. Операции сравнения и логические операции.

24. Синтаксис оператора If...Then в однострочной и блочной форме. Всегда ли они взаимозаменяемы?

25. Как работает оператор If...Then? Нарисуйте соответствующий фрагмент блок-схемы.

26. Синтаксис оператора If...Then...Else. Как он работает? Нарисуйте соответствующий фрагмент блок-схемы.

27. Синтаксис оператора If...Then...ElseIf. Как он работает? Нарисуйте блок-схему.

28. Как вывести результат вычислений в текстовый блок TextBox с использованием функции Format?

29. Типы условных операторов в VBA.

30. Назначение и синтаксис встроенной процедуры MsgBox.

31. Синтаксис оператора For ... Next

32. Синтаксис оператора DoLoop While

33. Синтаксис оператора Do UntilLoop

34. Какие команды служат для принудительного выхода из циклов?

35. Какое значение имеет счетчик после выхода из цикла For ... Next?

36. Как определяется массив переменных в программе?

37. Как занести в память значения для элементов массива?

38. Как записываются значения элементов массива в расчетных формулах?

39. Как определить количество элементов в массиве?

40. Как задать нумерацию элементов массива в нужном интервале?

41. Как найти сумму элементов массива?

42. Как найти произведение элементов массива?

43. Как найти минимальный элемент массива? 10. Как найти максимальный элемент массива?

44. Какие массивы данных можно представить, как двумерный массив?

45. Как определяется общее количество элементов двумерного массива?

46. Правила использования вложенных циклов.

47. Как осуществляется ввод двумерных массивов?

48. Как вывести двумерный числовой массив в виде матрицы?

49. Как запрограммировать вычисление суммы элементов матрицы?

50. Как запрограммировать вычисление суммы элементов в столбцах матрицы?

51. Как запрограммировать вычисление произведения элементов матрицы?

52. Что такое сортировка массива? Для чего она производится?
53. Каким образом производится удаление элементов массива?
54. Каким образом производится вставка элементов массива?
55. Что такое графический объект? Назовите графические объекты.
56. Что такое графический метод? Назовите графические методы.
57. Какие графические элементы управления Вам известны?
58. Какие единицы измерения применяются в графических объектах?
59. Для чего используется свойство ScaleMode?
60. Расскажите назначение метода Scale. Приведите синтаксис метода.
61. Что нужно сделать, чтобы объект класса Графическое окно с именем Picture1, который Вы уже разместили на экранной форме, имел ширину 14 (в условных единицах), высоту 10 и чтобы координатные оси располагались привычным для Вас образом: ось x была бы направлена слева направо, а ось y — снизу вверх, а начало координат располагалось в центре данного объекта?
62. Что такое объект Screen и каковы его основные свойства?
63. Перечислите основные свойства элемента управления Line.
64. Перечислите основные свойства элемента управления Shape.
65. Назовите графические свойства объектов.
66. Как можно добавить рисунок на форму?
67. Файлы каких графических форматов можно использовать в элементе управления рисунок (Image)?
68. Какой метод используется для вывода текста на форме (Form) или рисунке (PictureBox)?
69. Как в программном коде можно изменить параметры шрифта для вывода текста на форме (Form)?
70. Какие способы используются в VB для задания цвета?
71. Приведите формат RGB – функции.
72. Приведите синтаксис метода PSet.
73. Какой синтаксис метода Line и что означают его параметры?
74. Какой синтаксис метода Circle и что означают его параметры?
75. Особенности программного пакета AutoCAD и его возможности.
76. Интерфейс AutoCAD: элементы рабочего окна.
77. Настройка панелей инструментов.
78. Построение двухмерных объектов: абсолютные и относительные значения координат; типы двухмерных координат (декартовы и полярные).

79. Метод построений «направление – расстояние».
80. Настройка единиц измерения, границ рисунка, параметров сетки.
81. Команды черчения простейших примитивов (круг, дуга, прямоугольник и т. п.).
82. Объектные привязки – постоянные и временные. Объектное слежение.
83. Назначение слоев. Создание и удаление слоев. Настройка параметров слоя.
84. Вставка текстовой информации, формул и растровых изображений.
85. Команды модифицирования: перемещение, копирование, массив, зеркало и прочие.
86. Копирование объектов в прямоугольных или круглых областях. Создание эквидистантных объектов.
87. Команды черчения сложных объектов: полилиний, сплайнов, мультилиний и их редактирование.
88. Штриховка: выбор типа и настройка параметров; выбор заштриховываемых объектов. Ассоциативная и неассоциативная штриховка.
89. Типы текста (однострочный и многострочный).
90. Создание и редактирование текста.
91. Создание и редактирование текстовых стилей.
92. Создание и редактирование размерных стилей.
93. Особенности настройки режимов черчения и размерных стилей.
94. Преимущества использования блоков в чертежах.
95. Создание и редактирование блоков и атрибутов.
96. Создание и редактирование таблиц.
97. Вставка формул и полей в таблицу.
98. Изменение стилей таблиц.
99. Назначение центра управления AutoCAD.
100. Создание динамического блока.
101. Видовые экраны в пространстве модели и в пространстве листа.
102. Подрезка блоков и внешних ссылок.

5 РАЗДЕЛ 4. ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ

5.1 Литература

Основная литература

1. Система автоматизированного проектирования AutoCAD. Практикум: в 2 ч. / С. В. Гиль [и др.]; под общ. ред. С. В. Гиль; Белорусский национальный технический университет, Кафедра «Инженерная графика машиностроительного профиля». – Минск: БНТУ, 2020. – Ч. 1. – 84 с.

2. Компьютерная графика: учебно-методическое пособие для обучающихся по специальности 1-27 01 01 «Экономика и организация производства (по направлениям)», 1-27 01 01-10 «Экономика и организация производства (энергетика)» / Белорусский национальный технический университет, Энергетический факультет; сост. А. В. Манюкевич. – Минск: БНТУ, 2022. – 36 с.

3. Проектирование технологических схем по получению минеральных вяжущих и изделий на их основе [Электронный ресурс]: методические указания к выполнению технологической части курсового проекта «Вяжущие вещества» для студентов специальности 1-70 01 01 «Производство строительных изделий и конструкций» / Белорусский национальный технический университет, Кафедра "Строительные материалы и технология строительства"; сост.: Т. А. Чистова, О. Б. Сенько. – Минск: БНТУ, 2020.

4. Решение задач на VBA: учебно-методическое пособие по выполнению лабораторных работ / Т.Н. Литвинович; кол. авт. Белорусский государственный университет транспорта. – Гомель: БелГУТ, 2018. – 33 с.

5. Языки программирования. Python : учебно-методическое пособие для студентов специальности 1-40 01 01 «Программное обеспечение информационных технологий» : в 2 ч. / В. В. Иванченко [и др.] ; Белорусский национальный технический университет, Кафедра "Программное обеспечение информационных систем и технологий". – Минск : БНТУ, 2021. – Ч. 1. – 91 с.

Дополнительная литература

1. Соколова Т.Ю. AutoCAD 2016: двухмерное и трехмерное моделирование: учебный курс. - ДМК Пресс, 2016.

2. Программные средства создания Интернет-приложений : пособие для студентов учреждений высшего образования, обучающихся по специальности 1-08 01 01 «Профессиональное обучение (по направлениям)», направление специальности 1-08 01 01-07 «Профессиональное обучение (информатика)» / Белорусский национальный технический университет, Кафедра «Технология и методика преподавания» ; сост. Н. И. Астапчик. – Минск : БНТУ, 2021. – 108 с.

5.2 Учебная программа по учебной дисциплине "Решение технологических задач на ЭВМ"

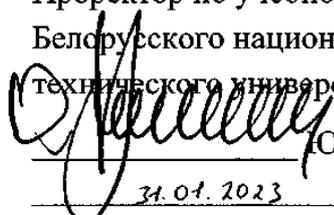
Белорусский национальный технический университет

УТВЕРЖДАЮ

Проректор по учебной работе

Белорусского национального

технического университета



О.А. Николайчик

31.01.2023

Регистрационный № УД-СФ67-30 /уч.

**РЕШЕНИЕ ТЕХНОЛОГИЧЕСКИХ
ЗАДАЧ НА ЭВМ**

Учебная программа учреждения высшего образования

по учебной дисциплине для специальности

1-70 01 01 «Производство строительных изделий и конструкций»

Минск 2022 г.

Учебная программа составлена на основе образовательного стандарта ОСВО 1-70 01 01-2018 «Производство строительных изделий и конструкций» и учебного плана специальности 1-70 01 01 «Производство строительных изделий и конструкций» рег.№ СФ 70д-1, СФ 70з-1, утв. 30.08.2018г.

СОСТАВИТЕЛИ:

Ж.Л. Зеленковская, старший преподаватель кафедры «Строительные материалы и технология строительства» Белорусского национального технического университета;

О.Б. Сенько, старший преподаватель кафедры «Строительные материалы и технология строительства» Белорусского национального технического университета.

РЕЦЕНЗЕНТЫ:

Т.М. Уласик, и.о. заведующего кафедрой «Геотехника и строительная механика» Белорусского национального технического университета, кандидат технических наук, доцент;

Д. М. Пикус, доцент кафедры «Экономика, организация строительства и управление недвижимостью» Белорусского национального технического университета, кандидат технических наук.

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой "Строительные материалы и технология строительства" Белорусского национального технического университета

(протокол № __ от _____ 2022 г.)

Заведующий кафедрой _____ С.Н. Леонович

Методической комиссией строительного факультета Белорусского национального технического университета

(протокол № ____ от _____ 2022 г.)

Председатель методической комиссии _____ Э.И. Батяновский

Научной библиотекой БНТУ _____ Т.И. Бирюкова

Научно-методическим советом Белорусского национального технического университета (протокол № ____ секции № 1 от _____ 2022 г.)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебная программа учреждения высшего образования дисциплины "Решение технологических задач на ЭВМ" разработана для специальности 1-70 01 01 "Производство строительных изделий и конструкций".

Целью изучения дисциплины является формирование у студентов компьютерной культуры и практических навыков записи математической модели поставленной задачи, составления принципиальной схемы алгоритма, реализации алгоритма на алгоритмическом языке программирования, комплекса знаний по современным методам технологического проектирования.

Учебная дисциплина базируется на знаниях, полученных при изучении курсов: "Математика", "Информатика".

Задача дисциплины – проведение сложных расчетов и научных исследований, использующих комбинации самых разных численных алгоритмов и аналитических преобразований при решении технологических задач; научить студентов самостоятельно проектировать технологические схемы и применять средства автоматизации проектирования при решении конкретных задач на базе современных достижений в компьютерной технике на всех этапах последующего обучения, в том числе при курсовом и дипломном проектировании.

В результате изучения дисциплины "Решение технологических задач на ЭВМ" студент должен:

знать:

- краткую характеристику математического пакета и его возможности;
- основы алгоритмизации инженерных задач;
- принципы программирования на алгоритмическом языке;
- технологии применения стандартных программ для компьютерного моделирования технологических задач;
- общие принципы о процессе, технологии и организации технологического проектирования с применением современных технических и программных средств;

уметь:

- ставить прикладные задачи, строить их математические модели, разрабатывать алгоритмы решения;
- реализовывать построенный алгоритм в виде собственной программы на алгоритмическом языке или с использованием стандартных программ;
- применять средства автоматизации проектирования для создания проектной документации;

владеть:

- численными методами при проведении сложных расчетов и научных исследований для решения технологических задач;
- методами использования компьютерных технологий для построения чертежей;
- приемами построения технологических схем в приложении AutoCAD .

Освоение данной учебной дисциплины обеспечивает формирование следующей компетенции:

СК-1. Уметь использовать программные средства для решения технологических задач при производстве сборных бетонных и железобетонных изделий и конструкций. Владеть основными программными средствами автоматизации разработки технологической и конструкторской документации.

Согласно учебным планам на изучение учебной дисциплины отведено:
- для очной формы получения высшего образования всего 230 ч., из них аудиторных - 132 часов;

- для заочной формы получения высшего образования всего 230 ч., из них аудиторных – 28 часов.

Распределение аудиторных часов по курсам, семестрам и видам занятий приведено в таблице 1.

Таблица 1.

Очная форма получения высшего образования					
Курс	Семестр	Лекции, ч.	Лабораторные занятия, ч.	Практические занятия, ч.	Форма текущей аттестации
2	4	32	32		зачет
3	5	34	34		зачет

Таблица 2.

Заочная форма получения высшего образования					
Курс	Семестр	Лекции, ч.	Лабораторные занятия, ч.	Практические занятия, ч.	Форма текущей аттестации
2	4	6	8		зачет
3	5	6	8		зачет

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел I. АЛГОРИТМИЧЕСКИЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ. ВОЗМОЖНОСТИ АЛГОРИТМИЧЕСКИХ ЯЗЫКОВ

Тема 1.1 Эффективность применения информационных технологий в области строительства

Программное обеспечение, как наличие готовых пакетов системных и прикладных программ для решения конкретных строительно-технологических задач. Математическое моделирование процессов является одним из основных способов получения новых знаний и технологических решений.

Тема 1.2. Объектно-ориентированное программирование

Основные понятия объектно-ориентированного программирования. Среда разработки приложений, в которую включено все, что необходимо для создания, модификации, тестирования, корректирования и компиляции программ. Проект приложения. Объекты управления и их свойства. События. Окно кода процедуры обработки события. Понятие метода. Компиляция проекта.

Тема 1.3. Данные, выражения, функции

Основные типы данных, константы, переменные. Арифметические, логические, строковые выражения. Встроенные математические, финансовые, системные функции. Функции обработки строк, даты и времени, преобразования типов, работы с графикой.

Тема 1.4. Выбор в программе, организация циклов

Условные операторы. Программирование алгоритмов разветвляющейся структуры. Способы организации циклов. Программирование алгоритмов циклической структуры.

Тема 1.5. Массивы данных, создание и использование вызываемых процедур и функций

Понятие массива данных. Виды массивов. Объявление массива. Способы создания массивов данных. Свойства элементов массивов. Организация процедур и функций. Программный код процедур и функций.

Раздел II. ГРАФИЧЕСКИЕ ПРИМИТИВЫ, СВОЙСТВА ОБЪЕКТОВ

Тема 2.1. Развитие систем автоматизированного проектирования. Общие сведения о системе AutoCAD

Сущность процесса автоматизированного проектирования. Основные требования, которые закладываются в основу создания и совершенствования систем автоматизированного проектирования (далее – САПР). Исторические этапы зарождения, становления и развития САПР. Схемы взаимодействия человека и ЭВМ на различных этапах. Структурная схема САПР. Технические средства САПР. Классификация, требования, предъявляемые к средствам графического моделирования в САПР.

Назначение пакета, его возможности. Основные различия векторной и растровой графики. Загрузка системы. Стартовое диалоговое окно. Графический интерфейс системы. Открытие существующих чертежей. Создание новых чертежей. Действия при возникновении проблем. Меню, строки и панели инструментов. Первоначальная настройка. Завершение работы и сохранение изображений.

Тема 2.2. Команды системы AutoCAD

Настройка пользовательского интерфейса. Командные строки, текстовое окно, диалог с программой. Технология работы с командами. Способы ввода команд. Структура команд. Команды зуммирования и панорамирования изображений.

Тема 2.3. Ввод координат и настройка параметров чертежа

Абсолютные и относительные координаты. Полярные координаты. Механизм объектных привязок. Назначение и настройка границ изображения. Стандартные форматы чертежей. Режимы рисования.

Тема 2.4. Графические примитивы как основа изображений

Простые примитивы и их построение. Построение прямоугольников, правильных многоугольников, окружностей, вспомогательных линий и эллипсов. Использование опций при построении графических примитивов.

Тема 2.5. Сложные примитивы. Работа с OLE-объектами. Основные свойства объектов

Команды построения и редактирования сложных примитивов. Вычерчивание отрезков и дуг с постоянной или переменной по длине шириной. Вставка текстовой информации, формул и растровых изображений. Редактирование вставленных объектов. Построение технологической схемы обжига цементного клинкера.

Назначение слоев. Создание слоев и работа с ними. Выбор и загрузка типа линии. Назначение типа линии объектам. Работа с масштабами линий.

Возможности редактирования свойств объектов. Импортирование слоев и типов линий из других чертежей.

Раздел III. РЕДАКТИРОВАНИЕ ОБЪЕКТОВ, РАЗМЕРНЫЕ СТИЛИ

Тема 3.1. Команды редактирования

Выбор объектов для редактирования. Команды изменения положения, копирования, удаления, и изменения формы объектов. Редактирование с использованием ручек. Копирование объектов в прямоугольных или круглых областях. Создание эквидистантных объектов. Принципиальное компоновочное решение генерального плана завода железобетонных изделий общественного назначения

Тема 3.2. Размеры и размерные стили

Настройка размерных стилей. Выполнение одиночных размеров, размеров от общей базы и размерных цепей. Редактирование размеров. Особенности настройки режимов черчения и размерных стилей.

Тема 3.3. Работа с текстом

Создание текстового стиля. Нанесение однострочного и многострочного текстов. Создание текстовых полей. Редактирование текста. Создание файлов-шаблонов. Использование текста при создании принципиальной схемы длинного протяжного стенда.

Тема 3.4. Штриховка и градиент

Параметры штриховки. Определение контуров штриховки. Режим обработки замкнутых объектов внутри контура штриховки. Копирование свойств. Редактирование штриховки.

Тема 3.5. Получение твердой копии чертежа

Устройство печати. Стили печати. Подготовка чертежа к печати. Возможности вывода на принтер/плоттер изображений, созданных в AutoCAD. Настройка опций диалогового окна Печать.

Раздел IV. ПОСТРОЕНИЕ ТЕХНОЛОГИЧЕСКИХ СХЕМ И СПЕЦИФИКАЦИЙ

Тема 4.1. Построение сложных объектов

Палитры. Особенности работы с мультилинией. Создание стилей мультилиний. Редактирование мультилиний. Геометрический калькулятор. Получение информации о чертеже и его элементах. Построение плана помещения при помощи команды Мультилиния.

Тема 4.2. Создание таблиц

Создание и редактирование таблиц. Вставка формул и полей в таблицу. Изменение стилей таблиц.

Тема 4.3. Блоки, внешние ссылки и динамические блоки

Преимущества использования блоков в чертежах. Создание и вставка блоков. Атрибуты блоков, их создание и редактирование. Построение плана квартиры с использованием атрибута блока Площадь. Использование внешних блоков. Внедрение ссылок. Присоединение чертежей. Подрезка блоков и внешних ссылок. Динамические блоки. Создание динамического блока на примере несущей конструкции пилона.

Тема 4.4. Видовые экраны в пространстве модели и в пространстве листа

Пространство модели и пространство листа. ВЭ в пространстве модели. Упорядоченное расположение и объединение смежных ВЭ. Плавающие видовые экраны. Редактирование ВЭ. Управление видимостью слоев ВЭ.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ
очная форма получения высшего образования

Номер раздела, темы	Наименование раздела, темы, занятия	Количество аудиторных часов				Количество часов УСР	Форма контроля знаний
		Лекции	Практические занятия	Лабораторные занятия	Семинарские занятия		
1	2	3	4	5	6	7	8
	4 семестр						
I.	АЛГОРИТМИЧЕСКИЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ. ВОЗМОЖНОСТИ АЛГОРИТМИЧЕСКИХ ЯЗЫКОВ						
1.1.	Эффективность применения информационных технологий в области строительства.	2					
	Лабораторная работа №1. "Интерфейс и настройки среды разработки"			2			
1.2.	Объектно-ориентированное программирование.	6					
	Лабораторная работа №2. "Объекты управления"			6			
1.3.	Данные, выражения, функции.	8					
	Лабораторная работа №3. "Данные, выражения, функции"			8			
1.4.	Выбор в программе, организация циклов.	8					
	Лабораторная работа №4. "Выбор в программе, организация циклов"			8			
1.5.	Массивы данных, создание и использование вызываемых процедур и функций.	8					

1	2	3	4	5	6	7	8
	Лабораторная работа №5. " Массивы данных, создание и использование вызываемых процедур и функций"			8			
	Итого за семестр	32		32			зачёт
	5 семестр						
II.	ГРАФИЧЕСКИЕ ПРИМИТИВЫ, СВОЙСТВА ОБЪЕКТОВ						
2.1.	Развитие систем автоматизированного проектирования. Общие сведения о системе AutoCAD	2					
2.2.	Команды системы AutoCAD.	2					
	Лабораторная работа №6. " Команды системы AutoCAD"			4			
2.3.	Ввод координат и настройка параметров чертежа	2					
2.4.	Графические примитивы как основа изображений.	2					
	Лабораторная работа №7. " Графические примитивы как основа изображений"			4			
2.5.	Сложные примитивы. Работа с OLE-объектами. Основные свойства объектов	2					
III.	РЕДАКТИРОВАНИЕ ОБЪЕКТОВ, РАЗМЕРНЫЕ СТИЛИ						
3.1.	Команды редактирования.	4					
	Лабораторная работа №8. " Команды редактирования"			6			
3.2.	Размеры и размерные стили.	4					
	Лабораторная работа №9. " Размеры и размерные стили"			4			

1	2	3	4	5	6	7	8
3.3.	Работа с текстом.	2					
	Лабораторная работа №10. " Работа с текстом."			2			
3.4.	Штриховка и градиент.	2					
	Лабораторная работа №11. " Штриховка и градиент."			2			
3.5.	Получение твердой копии чертежа	2					
IV.	ПОСТРОЕНИЕ ТЕХНОЛОГИЧЕСКИХ СХЕМ И СПЕЦИФИКАЦИЙ						
4.1.	Построение сложных объектов.	2					
	Лабораторная работа №12. " Построение сложных объектов"			2			
4.2.	Создание таблиц.	2					
	Лабораторная работа №13. " Создание таблиц"			2			
4.3.	Блоки, внешние ссылки и динамические блоки.	4					
	Лабораторная работа №14. " Блоки, внешние ссылки и динамические блоки"			4			
4.4.	Видовые экраны в пространстве модели и в пространстве листа.	2					
	Лабораторная работа №15. "Видовые экраны в пространстве модели и в пространстве листа"			4			
	Итого за семестр	34		34			зачёт
	ИТОГО:		132				

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ*
заочная форма получения высшего образования

Номер раздела, темы	Наименование раздела, темы, занятия	Количество аудиторных часов				Количество часов УСР	Форма контроля знаний
		Лекции	Практические занятия	Лабораторные занятия	Семинарские занятия		
1	2	3	4	5	6	7	8
	4 семестр						
I.	АЛГОРИТМИЧЕСКИЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ. ВОЗМОЖНОСТИ АЛГОРИТМИЧЕСКИХ ЯЗЫКОВ						
1.2.	Объектно-ориентированное программирование.	2					
	Лабораторная работа №1. "Интерфейс и настройки среды разработки"			2			
	Лабораторная работа №3. " Данные, выражения, функции"			2			
1.4.	Выбор в программе, организация циклов.	2					
	Лабораторная работа №4. " Выбор в программе, организация циклов"			2			

1	2	3	4	5	6	7	8
1.5.	Массивы данных, создание и использование вызываемых процедур и функций.	2					
	Лабораторная работа №5. " Массивы данных, создание и использование вызываемых процедур и функций"			2			
	Итого за семестр	6		8			зачёт
	5 семестр						
II.	ГРАФИЧЕСКИЕ ПРИМИТИВЫ, СВОЙСТВА ОБЪЕКТОВ						
2.3.	Команды системы AutoCAD	2					
	Лабораторная работа №6. " Команды системы AutoCAD"			2			
III.	РЕДАКТИРОВАНИЕ ОБЪЕКТОВ, РАЗМЕРНЫЕ СТИЛИ						
3.1.	Команды редактирования.	2					
	Лабораторная работа №8. " Команды редактирования"			2			
IV.	ПОСТРОЕНИЕ ТЕХНОЛОГИЧЕСКИХ СХЕМ И СПЕЦИФИКАЦИЙ						
4.1.	Построение сложных объектов.	2					
	Лабораторная работа №12. " Построение сложных объектов"			2			
	Лабораторная работа №13. " Создание таблиц"			2			
	Итого за семестр	6		8			зачёт
	ИТОГО:		28				

* Темы учебного материала, не указанные в Учебно-методической карте, отводятся на самостоятельное изучение студентом

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Список литературы

Основная литература

6. Система автоматизированного проектирования AutoCAD. Практикум: в 2 ч. / С. В. Гиль [и др.]; под общ. ред. С. В. Гиль; Белорусский национальный технический университет, Кафедра «Инженерная графика машиностроительного профиля». – Минск: БНТУ, 2020. – Ч. 1. – 84 с.

7. Компьютерная графика: учебно-методическое пособие для обучающихся по специальности 1-27 01 01 «Экономика и организация производства (по направлениям)», 1-27 01 01-10 «Экономика и организация производства (энергетика)» / Белорусский национальный технический университет, Энергетический факультет; сост. А. В. Манюкевич. – Минск: БНТУ, 2022. – 36 с.

8. Проектирование технологических схем по получению минеральных вяжущих и изделий на их основе [Электронный ресурс]: методические указания к выполнению технологической части курсового проекта «Вяжущие вещества» для студентов специальности 1-70 01 01 «Производство строительных изделий и конструкций» / Белорусский национальный технический университет, Кафедра "Строительные материалы и технология строительства"; сост.: Т. А. Чистова, О. Б. Сенько. – Минск: БНТУ, 2020.

9. Решение задач на VBA: учебно-методическое пособие по выполнению лабораторных работ / Т.Н. Литвинович; кол. авт. Белорусский государственный университет транспорта. – Гомель: БелГУТ, 2018. – 33 с.

10. Языки программирования. Python : учебно-методическое пособие для студентов специальности 1-40 01 01 «Программное обеспечение информационных технологий» : в 2 ч. / В. В. Иванченко [и др.] ; Белорусский национальный технический университет, Кафедра "Программное обеспечение информационных систем и технологий". – Минск : БНТУ, 2021. – Ч. 1. – 91 с.

Дополнительная литература

1. Соколова Т.Ю. AutoCAD 2016: двухмерное и трехмерное моделирование: учебный курс. - ДМК Пресс, 2016.

2. Программные средства создания Интернет-приложений : пособие для студентов учреждений высшего образования, обучающихся по специальности 1-08 01 01 «Профессиональное обучение (по направлениям)», направление специальности 1-08 01 01-07 «Профессиональное обучение (информатика)» / Белорусский национальный технический университет, Кафедра «Технология и методика преподавания» ; сост. Н. И. Астапчик. – Минск : БНТУ, 2021. – 108 с.

Средства диагностики результатов учебной деятельности

Оценка уровня знаний студента производится по десяти бальной шкале в соответствии с критериями, утвержденными Министерством образования Республики Беларусь.

Для оценки достижений студента рекомендуется использовать следующий диагностический инструментарий:

- защита выполненных на лабораторных занятиях заданий;
- защита выполненных в рамках самостоятельной работы индивидуальных заданий;
- выступление студента на конференции по подготовленному реферату;
- сдача зачета по дисциплине.

Перечень контрольных вопросов и заданий для самостоятельной работы студентов

1. Основные понятия объектно-ориентированного программирования.
2. Проект приложения. Объекты управления и их свойства.
3. События. Понятие метода. Компиляция проекта.
4. Основные типы данных, константы, переменные.
5. Условные операторы. Программирование алгоритмов разветвляющейся структуры.
6. Способы организации циклов. Программирование алгоритмов циклической структуры.
7. Организация процедур и функций. Программный код процедур и функций.
8. Особенности программного пакета AutoCAD и его возможности. Интерфейс AutoCAD: элементы рабочего окна. Настройка панелей инструментов.
9. Построение двухмерных объектов: абсолютные и относительные значения координат; типы двухмерных координат (декартовы и полярные). Метод построений «направление – расстояние».
10. Настройка единиц измерения, границ рисунка, параметров сетки.
11. Команды черчения простейших примитивов (круг, дуга, прямоугольник и т. п.). Объектные привязки – постоянные и временные. Объектное слежение.
12. Назначение слоев. Создание и удаление слоев. Настройка параметров слоя.
13. Команды модифицирования: перемещение, копирование, массив, зеркало и прочие.

14. Команды черчения сложных объектов: полилиний, сплайнов, мультилиний и их редактирование.

15. Штриховка: выбор типа и настройка параметров; выбор заштриховываемых объектов. Ассоциативная и неассоциативная штриховка.

16. Типы текста (однорочный и многорочный). Создание и редактирование текста. Создание и редактирование текстовых стилей.

17. Создание и редактирование размерных стилей.

18. Создание и редактирование блоков и атрибутов. Назначение центра управления AutoCAD.

19. Создание динамического блока.

20. Видовые экраны в пространстве модели и в пространстве листа.

Методические рекомендации по организации и выполнению самостоятельной работы студентов

При изучении дисциплины рекомендуется использовать следующие формы самостоятельной работы:

- решение индивидуальных заданий;
- подготовка сообщений, тематических докладов, презентаций по заданным темам;
- составление тематической подборки литературных источников, интернет-источников;
- проработка тем (вопросов), вынесенных на самостоятельное изучение.

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола заседания кафедры)
Согласование не требуется	Кафедра «Строительные материалы и технология строительства»	—	Содержание данной учебной программы не требует согласования с другими учебными дисциплинами специальности. Протокол № <u>10</u> от <u>30.05.2022</u>