

КОНСОЛЬНЫЕ ПРИЛОЖЕНИЯ: ПРЕИМУЩЕСТВА И НЕДОСТАТКИ

БНТУ, Минск

Научный руководитель: Дробыш А.А.

В настоящее время, когда у разработчиков программного обеспечения появляется всё больше возможностей для создания приложений со сложным графическим интерфейсом, консольные приложения по-прежнему прочно удерживают свои позиции, даже в такой, казалось бы, им среде, как современная операционная система Windows. Это объясняется в первую очередь простотой их исполнения и некоторыми специфическими особенностями консольных приложений, которые делают их в ряде случаев более подходящими для решения задачи, чем приложения с графическим интерфейсом. Один минус: выглядят такие приложения очень уныло и однообразно, как безликая серая масса букв и цифр на чёрном фоне окна консоли. Консольным приложением называется «*программа*», которая не имеет графического интерфейса – окон, и которая работает в текстовом режиме в черно-белой консоли. Команды в такой программе приходится вводить с клавиатуры, результаты работы консольные приложения также выводят на экран в текстовом режиме

Создание проекта приложения консоли (CLR) среды CLR

В строке меню Файл выберите пункты «Создать Проект».

В диалоговом окне «Создать проект» в поле Установленные шаблоны выберите узел Visual C++ выделите узел CLR, а затем выберите шаблон консольного приложения.

В поле Имя введите уникальное имя приложения.

Можно указать другие параметры проекта и решения, но это не обязательно.

Нажмите кнопку ОК.

При использовании шаблона консольного приложения, чтобы создать проект, автоматически добавляет эти ссылки и файлы.

Ссылки на следующие пространства имен .NET Framework:

Система – Содержит основные классы и базовые классы, которые определяют часто используемых значений и ссылочные типы данных, события и обработчики событий, интерфейсы, атрибуты и обработке исключений.

DLL сборки – Mscorlib, которое поддерживает разработку платформы .NET Framework.

Исходные файлы: Консоль (сpp-файл) — основной файл источника и точка входа в приложение, созданные. Он указывает DLL-файл проекта и пространство имен проекта. Поместите свой код в этот файл.; Атрибуты AssemblyInfo.cpp-Contains, файлов ресурсов, типы данных системы управления версиями, подписывая информацию и т.д., с помощью которых можно изменить метаданные сборки проекта. Для получения дополнительной информации см. Содержимое сборок.

Stdafx.cpp-Used для построения с именем Win32.pch предкомпилированного файла заголовка и предварительно компилируются с именем файла StdAfx.obj типов.

Файлы заголовков: Stdafx.h-Used для построения с именем Win32.pch предкомпилированного файла заголовка и предварительно компилируются с именем файла StdAfx.obj типов. созданные resource.h-А включают файл для app.rc.

Файлы ресурсов: файл сценария ресурсов app.rc-The программы.; файл Значка app.ico-The программы.; ReadMe.txt – описание файлов в проекте.

Преимущества и недостатки

Преимущества:

1. Скорость работы итоговой программы (по сравнению с Java/Python).

2. Прямое управление динамической памятью, что опять же важно для скорости работы.

Недостатки:

1. Трудный для понимания синтаксис, который вызывает ошибки. Близкие по обозначению разнонаправленные операции (можно спутать).

2. 3 основных операции (присваивание, инкрементация, декрементация) в сочетании с другими операциями предполагает создание сложных для чтения выражений (усложняет диагностику).

3. Необходимо очень внимательно работать с указателями, ссылками, динамической памятью, проверять не происходит ли выход за границы массива.

УДК 372

Чайко Е.Ю.

АЛФАВИТ ЯЗЫКА ПРОГРАММИРОВАНИЯ C++

БНТУ, Минск

Научный руководитель: Дробыш А.А.

Алфавит (или множество литер) языка программирования C++ основывается на множестве символов таблицы кодов ASCII. Алфавит C++ включает:

- строчные и прописные буквы латинского алфавита (мы их будем называть буквами),
- цифры от 0 до 9 (назовём их буквами-цифрами),
- символ '_' (подчерк – также считается буквой),
- набор специальных символов: " { } , | [] + - % / \ ; ' : ? < > = ! & # ~ ^ . * и прочие символы.

В тесте программы можно использовать комментарии. Комментарий – это последовательность символов, которая игнорируется компилятором языка C++. Комментарий имеет следующий вид: /*<символы>*/. Комментарии могут