

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Белорусский национальный технический университет

ОПТИМИЗАЦИЯ ПРОЕКТНЫХ РЕШЕНИЙ

Методические указания к лабораторной работе
**«Использование методов динамического
программирования
для решения задач оптимизации»**

для студентов специальности

1 40 01 02 – «Информационные системы и технологии (по направлениям)»
направления 1 40 01 02 -01 – «Информационные системы и технологии в
проектировании и производстве»

Учебное электронное издание

Минск ◊ БНТУ ◊ 2008

УДК: 681.3.07

Авторы:

*И.Л. Ковалева,
Л.В. Федосова*

Рецензенты:

Т. А. Долгова, кандидат физико-математических наук, доцент БГТУ;
Н.А.Разоренов, кандидат технических наук, заведующий кафедрой «ПОВТ
и АС» БНТУ

Методические указания содержат рекомендации по выполнению оптимального проектирования на основе детерминированных моделей динамического программирования в системе 1С: Предприятие. Изложена краткая теория, описывающая фундаментальный принцип динамического программирования, составляющий основу декомпозиции задачи на этапы. Проанализированы результаты, полученные при решении тестовой задачи о загрузке. Приведен пример кода.

Разработанные задания направлены на закрепление студентами изложенного материала в ходе самостоятельного выполнения ими оптимального планирования предложенной задачи.

Белорусский национальный технический университет
пр-т Независимости, 65, г. Минск, Республика Беларусь
Тел.(017) 292-67-85 факс (017) 292-71-53
E-mail: il05kov@mail.ru
Регистрационный № _____

© БНТУ, 2008

© Ковалева И.Л., Федосова Л.В., 2008

© Ковалева И.Л., компьютерный дизайн, 2008

Содержание

1 КРАТКАЯ ТЕОРИЯ	4
2 ЗАДАЧА О ЗАГРУЗКЕ	5
2.1 Постановка задачи в общем виде	5
2.2 Пример решения	7
2.3 Решение задачи о загрузке в системе <i>1С: Предприятие 8.0</i>	10
Задание	31

Цель работы: разработка конфигурации для автоматизированного решения задач оптимального планирования методами динамического программирования в системе 1С: Предприятие и анализ полученных результатов.

1 КРАТКАЯ ТЕОРИЯ

Динамическое программирование (или, иначе, «динамическое планирование») представляет собой особый математический аппарат, позволяющий осуществлять оптимальное планирование управляемых процессов. Под «управляемыми» понимаются процессы, на ход которых мы можем в той или другой степени влиять.

Динамическое программирование – это вычислительный метод для решения задач определенной структуры. Возникло и сформировалось в 1950-1953 гг. благодаря работам Р. Беллмана над динамическими задачами управления запасами. В упрощенной формулировке динамическое программирование представляет собой направленный последовательный перебор вариантов, который обязательно приводит к глобальному максимуму (минимуму).

Пусть предполагается к осуществлению некоторое мероприятие или серия мероприятий (короче говоря, «операция»), преследующая определенную цель. Спрашивается: как нужно организовать (спланировать) операцию для того, чтобы она была наиболее эффективной, т.е. наилучшим образом удовлетворяла поставленным перед ней требованиям? Современная математическая наука располагает целым арсеналом методов, позволяющих решить задачу оптимального планирования. Среди них особое место занимает метод динамического программирования. Специфика этого метода в том, что для отыскания оптимального управления планируемая операция разделяется на ряд последовательных «шагов» или «этапов». Соответственно и самый

процесс планирования становится ”многошаговым” и развивается последовательно, от этапа к этапу, причем каждый раз оптимизируется управление только на одном шаге.

Чтобы поставленная задача оптимального планирования приобрела количественный, математический характер, необходимо ввести в рассмотрение некоторый численный критерий, которым мы будем характеризовать качество, успешность, эффективность операции.

2 ЗАДАЧА О ЗАГРУЗКЕ

2.1 Постановка задачи в общем виде

Одной из задач, решаемых методом динамического программирования, является задача об оптимальной загрузке. Задача о загрузке — это задача о рациональной загрузке судна (самолета, автомашины и т.п.), которое имеет ограничения по объему или грузоподъемности. Каждый помещенный на судно груз приносит определенную прибыль. Задача состоит в определении загрузки судна такими грузами, которые приносят наибольшую суммарную прибыль.

Перед тем, как представить соотношения динамического программирования, заметим, что рассматриваемая здесь задача известна также как задача о снаряжении, где пилот реактивного самолета должен определить наиболее ценные (необходимые) предметы, которые следует взять на борт самолета, или как задача о рюкзаке, в которой солдат (или турист) должен определить наиболее ценные предметы, подлежащие загрузке в ранец (рюкзак). Кажется, что три упомянутых названия для одной и той же задачи были выбраны для того, чтобы гарантировать равное представительство военно-морского флота, воздушных сил и армии.

Рекуррентное уравнение процедуры обратной прогонки выводится для общей задачи загрузки судна грузоподъемностью W предметов

(грузов) n наименований. Пусть m_i количество предметов i -го наименования, подлежащих загрузке, r_i — прибыль, которую приносит один загруженный предмет i -го наименования, w_i — вес одного предмета i -го наименования. Общая задача имеет вид следующей целочисленной задачи линейного программирования.

Максимизировать

$$z = r_1 m_1 + r_2 m_2 + \dots + r_n m_n$$

при условии, что

$$w_1 m_1 + w_2 m_2 + \dots + w_n m_n \leq W$$

$$m_1, m_2, \dots, m_n \geq 0 \text{ и целые.}$$

Три основные элемента модели динамического программирования — этапы, варианты решения и состояния определяются для этой задачи следующим образом:

1. Этап i ставится в соответствие предмету i -го наименования, $i = 1, 2, \dots, n$.
2. Варианты решения на этапе i описываются количеством m_i , предметов i -го наименования, подлежащих загрузке. Соответствующая прибыль равна $r_i m_i$. Значение m_i заключено в пределах от 0 до $[W/w_i]$, где $[W/w_i]$ — целая часть числа W/w_i .
3. Состояние x_i на этапе i выражает суммарный вес предметов, решения о погрузке которых приняты на этапах $i, i+1, \dots, n$. Это определение отражает тот факт, что ограничение по весу является единственным, которое связывает n этапов вместе.

Пусть $f_i(x_i)$ — максимальная суммарная прибыль от этапов $i, i+1, \dots, n$ при заданном состоянии x_i . Проще всего рекуррентное уравнение определяется с помощью следующей двухшаговой процедуры.

Шаг 1. Выразим $f_i(x_i)$ как функцию $f_{i+1}(x_{i+1})$ в виде

$$f_i(x_i) = \max\{r_i m_i + f_{i+1}(x_{i+1})\}, i = 1, 2, \dots, n,$$

$$m_i = 0, 1, \dots, \left\lfloor \frac{w}{w_i} \right\rfloor$$

$$x_i = 0, 1, \dots, W$$

где $f_{n+1}(x_{n+1}) \equiv 0$.

Шаг 2. Выразим x_{i+1} как функцию x_i для гарантии того, что левая часть последнего уравнения является функцией лишь x_i . По определению $x_i - x_{i+1}$ представляет собой вес, загруженный на этапе i , т.е. $x_i - x_{i+1} = w_i m_i$ или $x_{i+1} = x_i - w_i m_i$. Следовательно, рекуррентное уравнение приобретает следующий вид.

$$f_i(x_i) = \max\{r_i m_i + f_{i+1}(x_i - w_i m_i)\}, i = 1, 2, \dots, n,$$

$$m_i = 0, 1, \dots, \left\lfloor \frac{w}{w_i} \right\rfloor$$

$$x_i = 0, 1, \dots, W$$

2.2 Пример решения

Рассмотрим конкретный пример. В 4-тонный самолет загружаются предметы трех наименований. Приведенная ниже таблица 1 содержит данные о весе одного предмета w_i (в тоннах) и прибыли r_i (в тысячах долларов), получаемой от одного загруженного предмета, Как необходимо загрузить самолет, чтобы получить максимальную прибыль?

Таблица 1 Исходные данные

Предмет i	w_i	r_i
1	2	31
2	3	47
3	1	14

Так как вес одного предмета w_i для всех наименований и максимальный вес W принимают целочисленные значения, состояние x_i может принимать лишь целочисленные значения.

Этап 3. Точный вес, который может быть загружен на этапе 3 (предмет наименования 3), заранее неизвестен, но он должен принимать одно из значений 0, 1, ..., 4 (так как $W = 4$ тонны). Состояния $x_3 = 0$ и $x_3 = 4$ представляют собой крайние случаи, когда предметы третьего наименования совсем не загружаются или загружают самолет полностью. Остальные значения $x_3 (= 1, 2 \text{ или } 3)$ предполагают частичную загрузку самолета предметами третьего наименования. Действительно, при этих значениях x_3 все оставшиеся емкости самолета могут быть заполнены предметами третьего наименования.

Так как вес w_3 одного предмета третьего типа равен 1 тонне, максимальное количество единиц этого типа, которое может быть загружено, равно $[4/1] = 4$. Это означает, что возможными значениями x_3 будут 0, 1, 2, 3 и 4. Решение m_3 является допустимым лишь при условии, что $w_3 m_3 \leq x_3$. Следовательно, все недопустимые альтернативы (те, для которых $w_3 m_3 > x_3$) исключены. Следующее уравнение является основой для сравнения альтернатив на этапе 3.

$$f_3(x_3) = \max\{14m_3\}, \quad \max\{m_3\} = \left[\frac{4}{1} \right] = 4$$

В таблице 2 сравниваются допустимые решения для каждого значения x_3 .

Таблица 2

Данные этапа 3

x_3	$14m_3$						Оптимально решение
	$m_3 = 0$	$m_3 = 1$	$m_3 = 2$	$m_3 = 3$	$m_3 = 4$	$f_3(x_3)$	m_3
0	0	–	–	–	–	0	0
1	0	14	–	–	–	14	1
2	0	14	28	–	–	28	2
3	0	14	28	42	–	42	3
4	0	14	28	42	56	56	4

Этап 2.

$$f_2(x_2) = \max \left\{ 47m_2 + f_3(x_2 - 3m_2) \right\}, \quad \max \left\{ m_2 \right\} = \left[\frac{4}{3} \right] = 1$$

В таблице 3 сравниваются допустимые решения для каждого значения x_2

Таблица 3

Данные этапа 2

x_2	$47m_2 + f_3(x_2 - 3m_2)$		Оптимально решение	
	$m_2 = 0$	$m_2 = 1$	$f_2(x_2)$	m_2
0	0+0=0	–	0	0
1	0+14=14	–	14	0
2	0+28=28	–	28	0
3	0+42=42	47+0=47	47	1
4	0+56=56	47+14=61	61	1

Этап 1.

$$f_1(x_1) = \max \left\{ 31m_1 + f_2(x_1 - 2m_1) \right\}, \quad \max \left\{ m_1 \right\} = \left[\frac{4}{2} \right] = 2$$

В таблице 4 сравниваются допустимые решения для каждого значения x_1

Таблица 4

Данные этапа 1

x_1	$31m_1 + f_2(x_1 - 2m_1)$			Оптимально решение	
	$m_1 = 0$	$m_1 = 1$	$m_1 = 2$	$f_1(x_1)$	m_1
0	0+0=0	–	–	0	0
1	0+14=14	–	–	14	0
2	0+28=28	31+0=31	–	31	1
3	0+47=47	31+14=45	–	47	0
4	0+61=61	31+28=59	62+0=62	62	2

Оптимальное решение определяется теперь следующим образом. Из

условия $W = 4$ следует, что первый этап решения задачи при $x_1 = 4$ дает

оптимальное решение $m_1^* = 2$, которое означает, что два предмета первого наименования будут загружены в самолет. Эта загрузка оставляет $x_2 = x_1 - 2m_1^* = 4 - 2 \times 2 = 0$. Решение на втором этапе при $x_2 = 0$ приводит к оптимальному решению $m_2^* = 0$, которое, в свою очередь, дает $x_3 = x_2 - 3m_2^* = 0 - 3 \times 0 = 0$. Далее этап 3 при $x_3 = 0$ приводит к $m_3^* = 0$. Следовательно, оптимальным решением задачи является $m_1^* = 2$, $m_2^* = 0$ и $m_3^* = 0$. Соответствующая прибыль равна 62000 долларов,

В таблице 4 для первого этапа нам, по существу, необходимо получить оптимальное решение лишь для $x_1 = 4$, так как это последний этап, подлежащий рассмотрению. Однако в таблицу включены также вычисления для $x_1 = 0, 1, 2$ и 3 , которые позволяют провести анализ чувствительности решения. Например, что произойдет, если максимальная грузоподъемность самолета будет 3 тонны вместо 4? Новое оптимальное решение может быть определено, начиная с $x_1 = 3$ на первом этапе и продолжая так, как мы поступали при $x_1 = 4$.

Задача о загрузке является типичным представителем задачи *распределения ресурсов*, в которой ограниченный ресурс распределяется между конечным числом видов (экономической) деятельности. При этом целью является максимизация соответствующей функции прибыли. В таких моделях определение состояния на каждом этапе будет аналогично приведенному для задачи о загрузке: состоянием на этапе i является суммарное количество ресурса, распределяемого на этапах $i, i+1, \dots, n$.

2.3 Решение задачи о загрузке в системе 1С: Предприятие 8.0

Подробно рассмотрим основные этапы автоматизации решения задачи о загрузке в системе 1С: Предприятие 8.0.

1. Запускаем *1С: Предприятие 8.0* в режиме «Конфигуратор». Последовательность шагов для запуска представлена на рис.1.

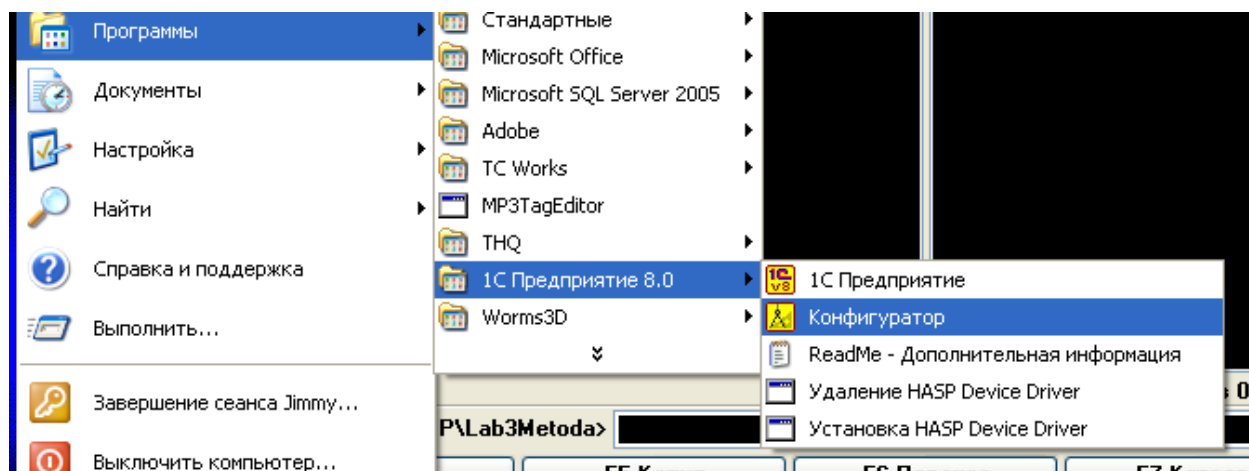


Рис.1. Запуск *1С: Предприятия 8.0* в режиме «Конфигуратор»

2. Далее необходимо создать новую информационную базу (далее ИБ). Для этого в открывшемся окне нажимаем «Добавить...» (рис.2):

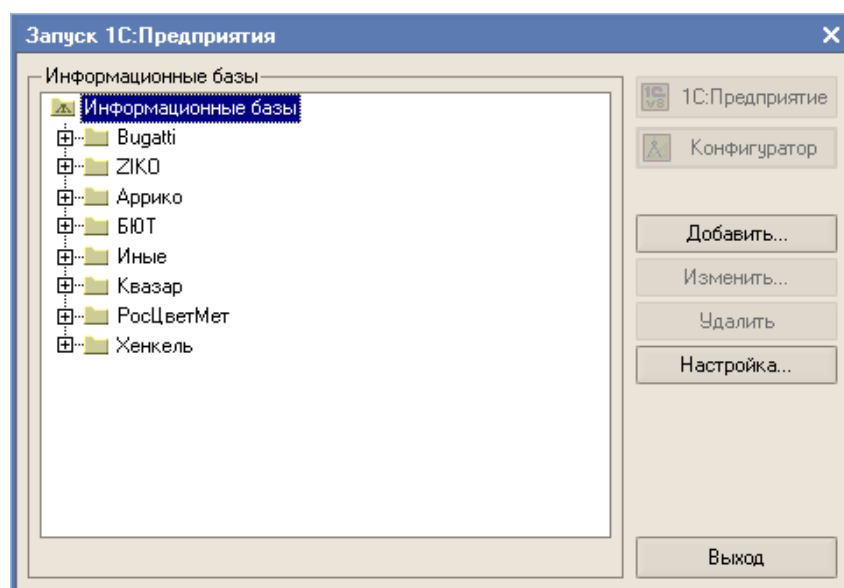


Рис.2. Первый шаг создания новой информационной базы

3. На следующем шаге производится выбор режима добавления ИБ в список. Т.к. создается новая ИБ необходимо выбрать «Создание новой ИБ» и нажимаем «Далее» (рис.3):

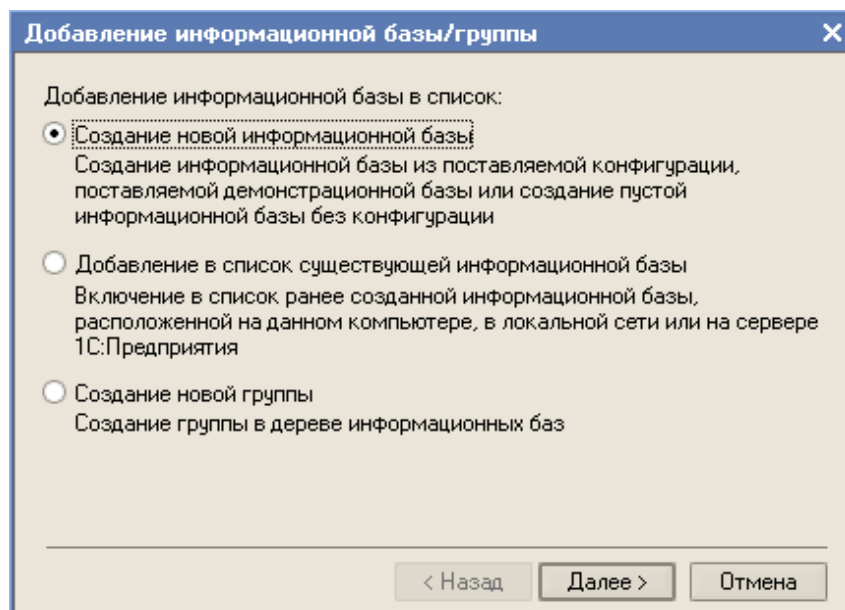


Рис.3. Второй шаг создания новой информационной базы

4. В появившемся окне выбираем режим «Создание информационной базы без конфигурации...» и переходим к следующему шагу, нажимая кнопку «Далее» (рис.4):

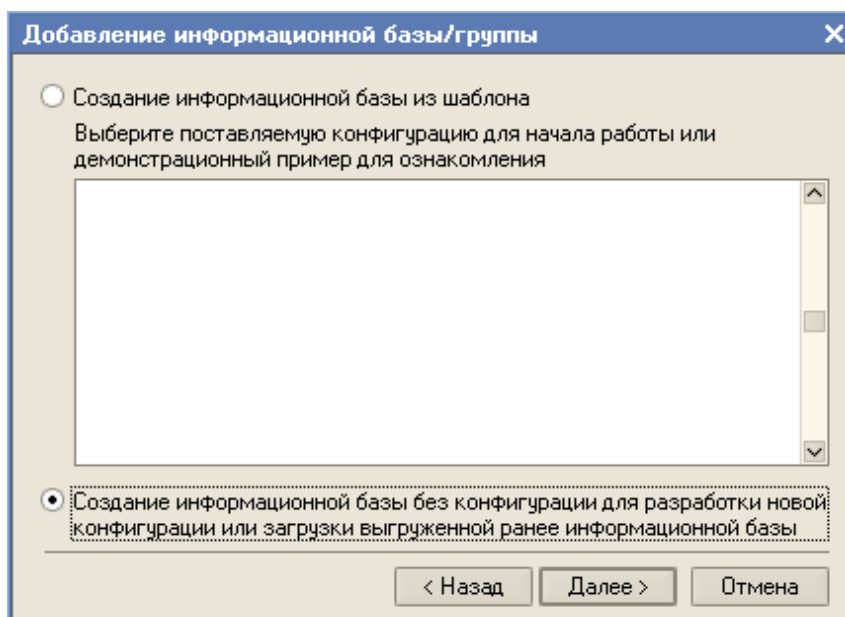


Рис.4. Создание новой информационной базы без конфигурации

5. Задаем необходимо ввести наименование (имя) ИБ (в нашем случае – ЗагрузкаСамолета), указать тип ее расположения и перейти далее нажатием кнопки «Далее» (рис.5):

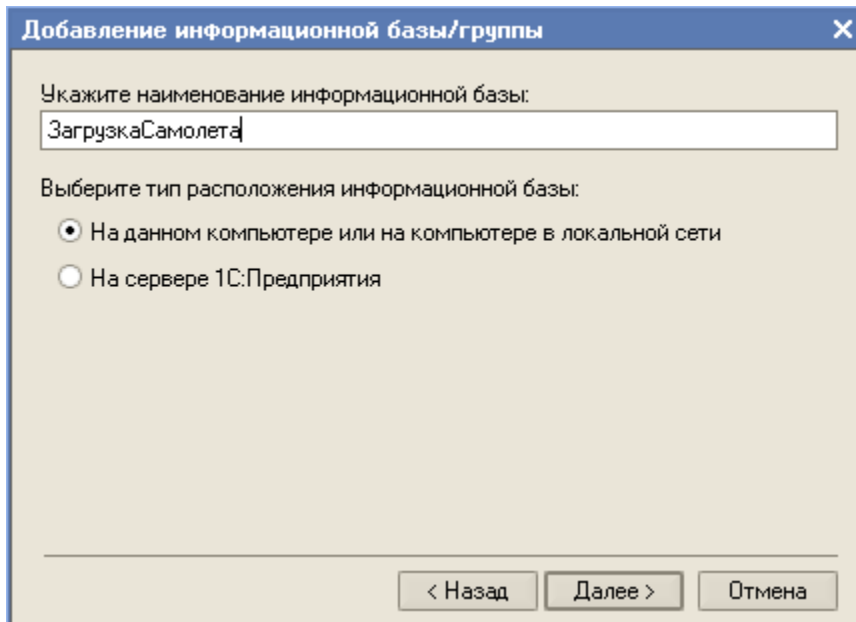


Рис.5. Создание новой информационной базы без конфигурации

6. Затем задаются такие параметры ИБ как каталог для хранения и язык (страна). На этом процесс создания новой ИБ завершен, что подтверждается нажатием кнопки «Готово» (рис.6):

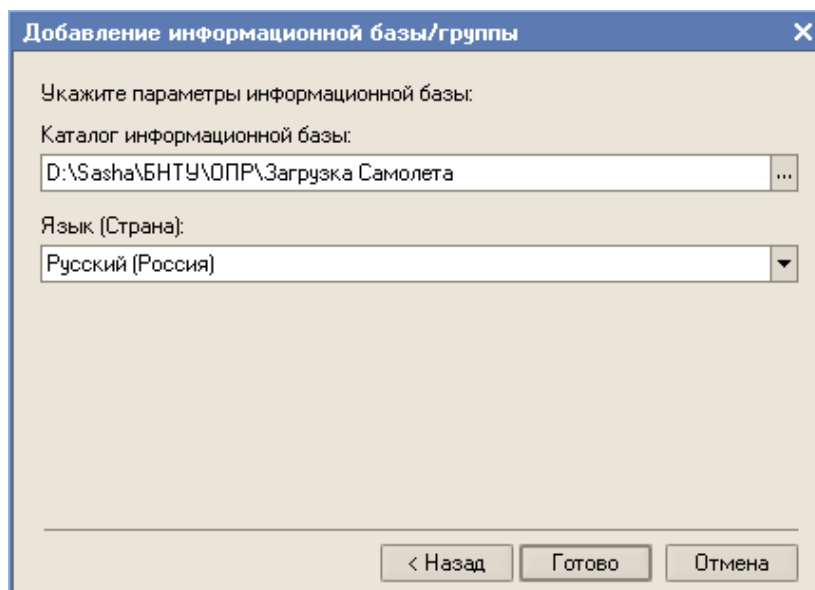


Рис.6. Завершающий этап создания новой информационной базы

7. Созданную ИБ «ЗагрузкаСамолета» необходимо запустить в режиме «Конфигуратор» (рис.7):

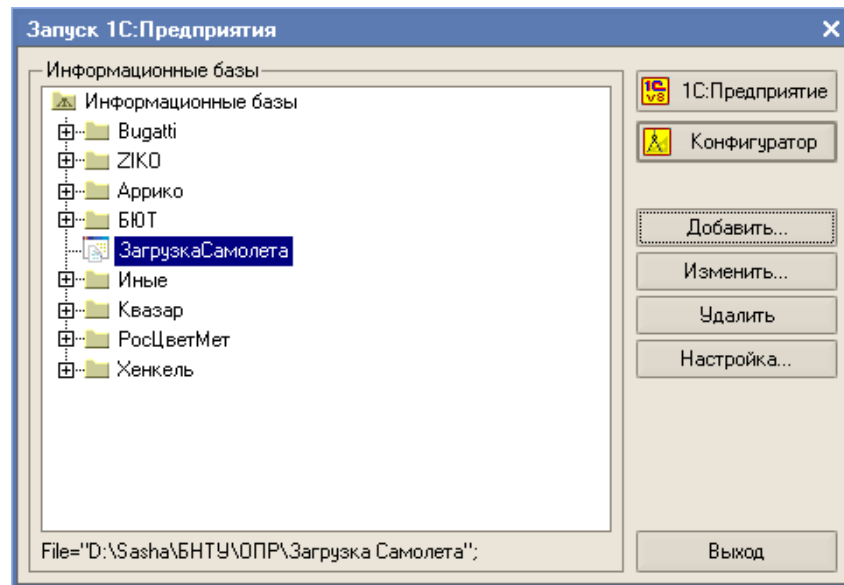


Рис.7. Выбор режима работы системы с новой ИБ

8. Для запуска ИБ в главном меню «Конфигуратора» выбираем закладку «Конфигурация» и затем выполняем действие, показанное на рис.8: *Конфигурация → Открыть конфигурацию:*

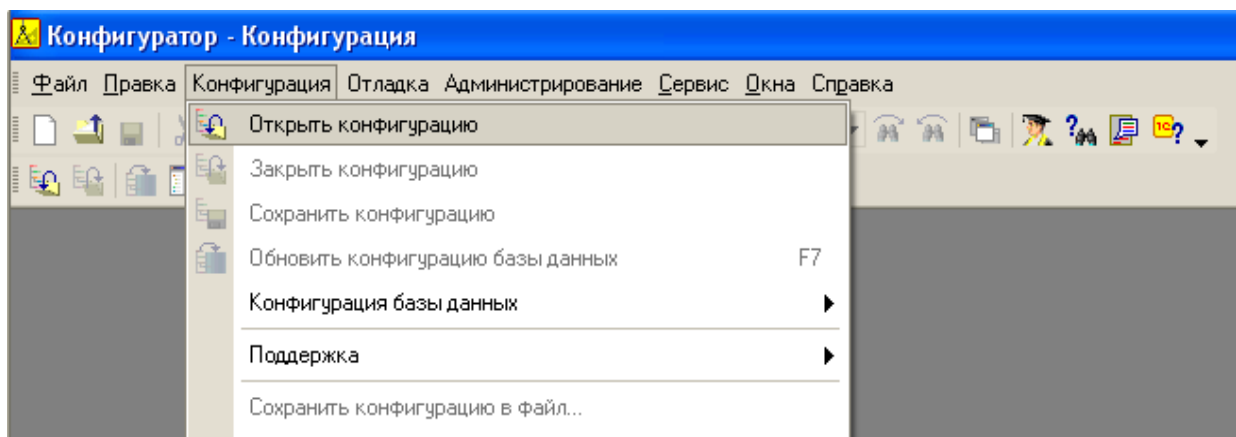


Рис.8. Открытие новой ИБ в режиме «Конфигуратора»

9. Для решения задачи загрузки, т.е. обработки исходных данных можно использовать не сохраняемый в базе данных объект системы *1С: Предприятие 8. «Обработка»*. Выбрать его можно в открывшемся дереве конфигурации (рис.9). Дерево конфигурации позволяет создавать и

удалять объекты конфигурации, менять их порядок, копировать и переносить объекты по подсистемам.

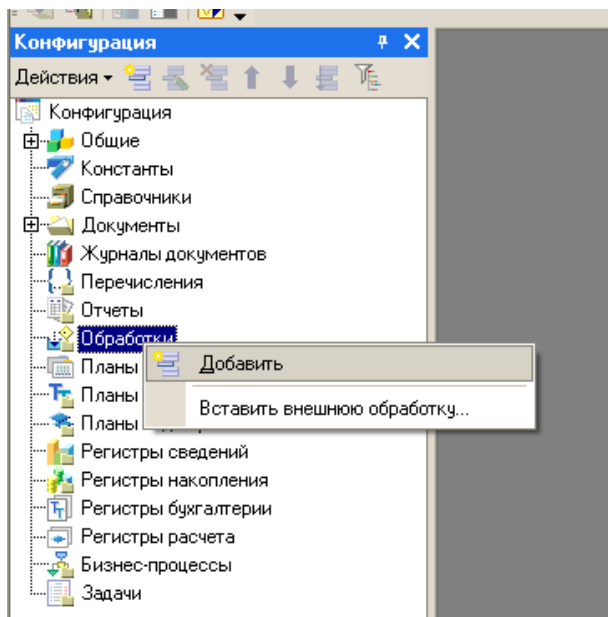


Рис.9. Выбор объектов в системе 1С: Предприятие 8.0

10. Для того чтобы добавить новый объект, необходимо на закладке «Основные» задать его имя и синоним (рис.10).

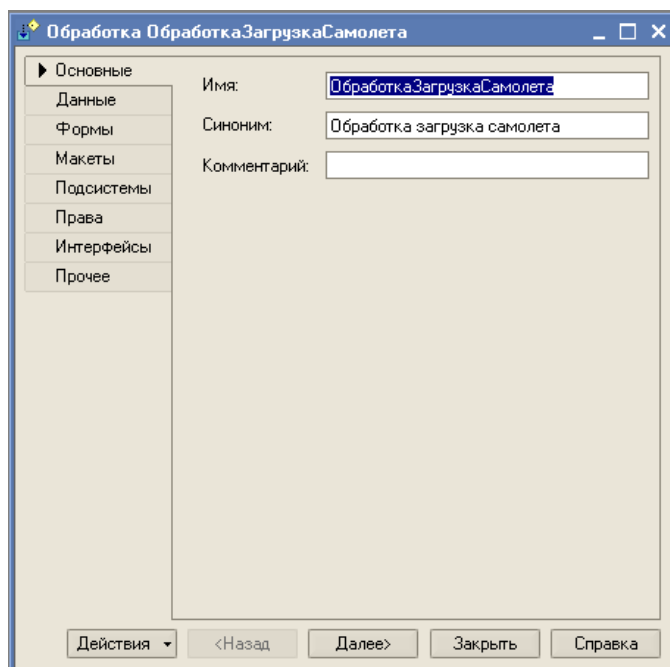


Рис.10. Задание имени для объекта «Обработка»

11. Для вновь созданного объекта «ОбработкаЗагрузкаСамолета» можно сформировать подчиненный объект – форму. Для этого на закладке «Формы» выбираем «Добавить» и создаем новую форму (рис.11).

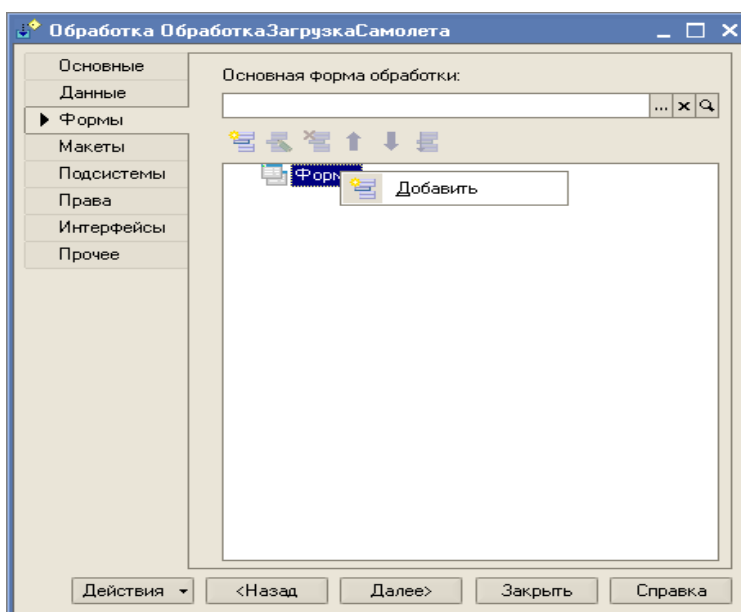


Рис.11. Создание формы для объекта «ОбработкаЗагрузкаСамолета»

12. Тип формы: произвольная, снимаем указатели «Командная панель формы сверху» и «Командная панель формы снизу» (они не нужны), остальное оставляем по умолчанию и нажимаем «Готово» (рис.12).

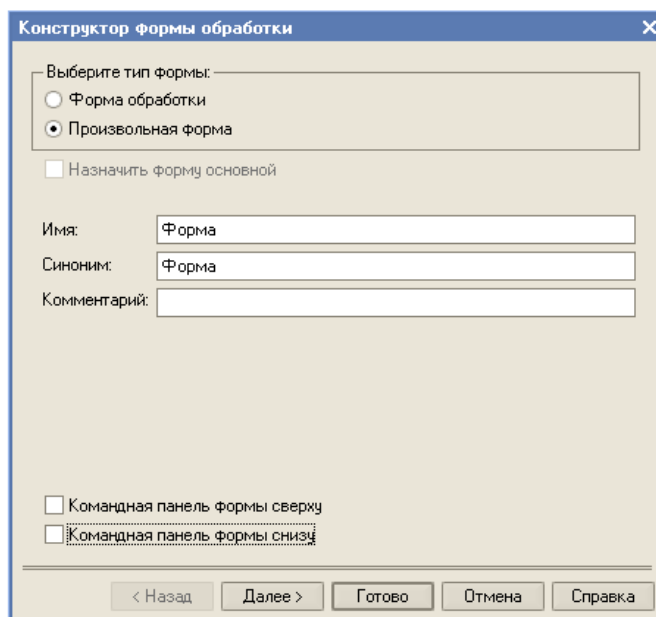


Рис.12. Создание формы для объекта «ОбработкаЗагрузкаСамолета»

В результате описанных выше действий создалась новая форма (рис.13).

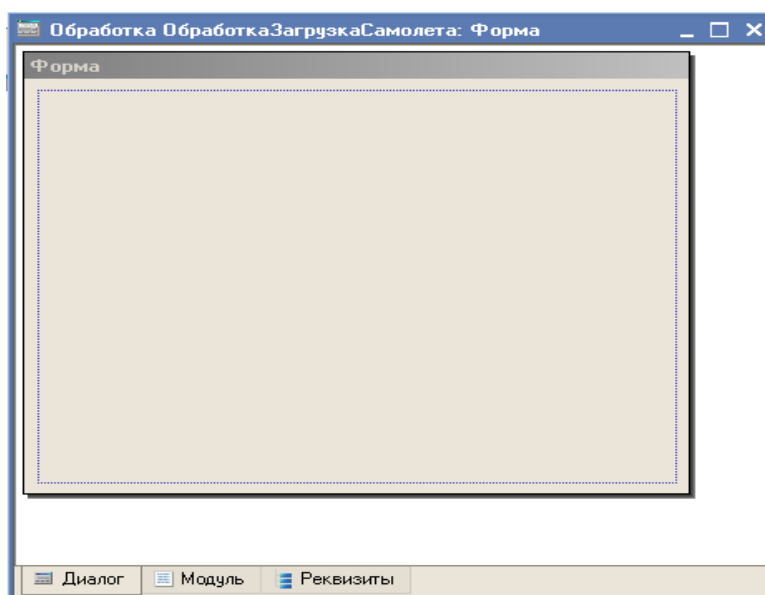


Рис.13. Исходная форма для объекта «ОбработкаЗагрузкаСамолета»

13. В форме могут размещаться различные элементы управления, например, *Кнопка*, *Надпись*, *Флажок* и т.д. Используем для наполнения формы для объекта «ОбработкаЗагрузкаСамолета» элементы управления *Поле ввода*, *Кнопка*, *Поле табличного документа*. Они создаются в ходе выполнения следующей последовательности действий: Главное меню – Форма – Вставить элемент управления (рис.14).

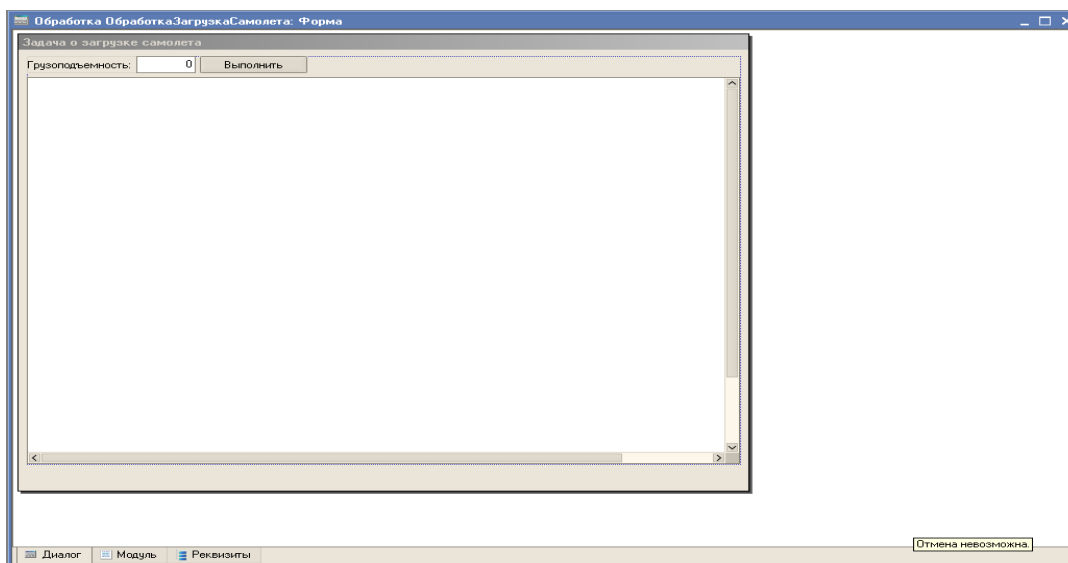


Рис.14. Окончательный вид формы для объекта «ОбработкаЗагрузкаСамолета»

Все атрибуты для каждого вводимого элемента управления задаются в его свойствах, либо изначально при создании (рис.15).

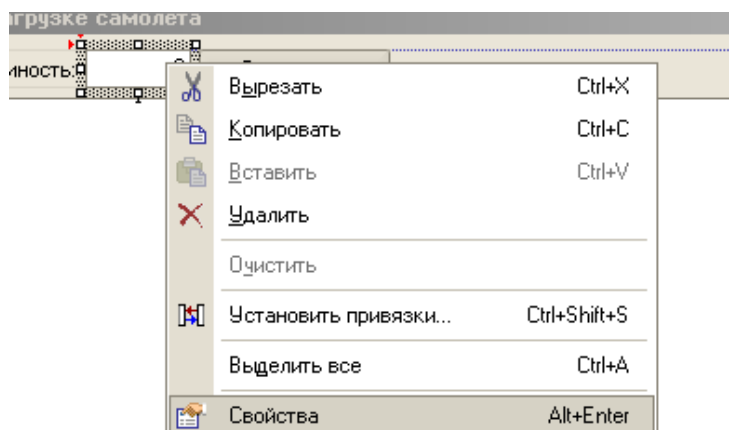


Рис.15. Задание атрибутов элемента управления

Для решаемой задачи зададим следующие атрибуты элементов управления:

Поле ввода:

Имя: Грузоподъемность

Данные: Грузоподъемность

Тип значения: Число (10,0)

Кнопка:

Имя: Выполнить

Данные: Выполнить

События, нажатие: КнопкаВыполнитьНажатие

Поле табличного документа:

Имя: ДокументРезультат

14. Для задания исходных данных можно использовать объект «Справочник». Добавим его в дерево конфигурации (рис.16).

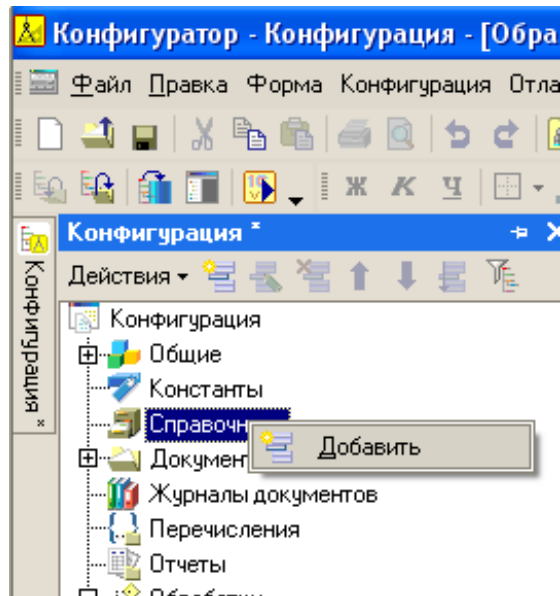


Рис.16. Добавление объекта «Справочник» в дерево конфигурации

15. Справочники являются основным механизмом хранения условно-постоянной информации. Для описания предметов загрузки, веса, прибыли создаем справочник «ОбъектыДляЗагрузкиСамолета» (рис.17).

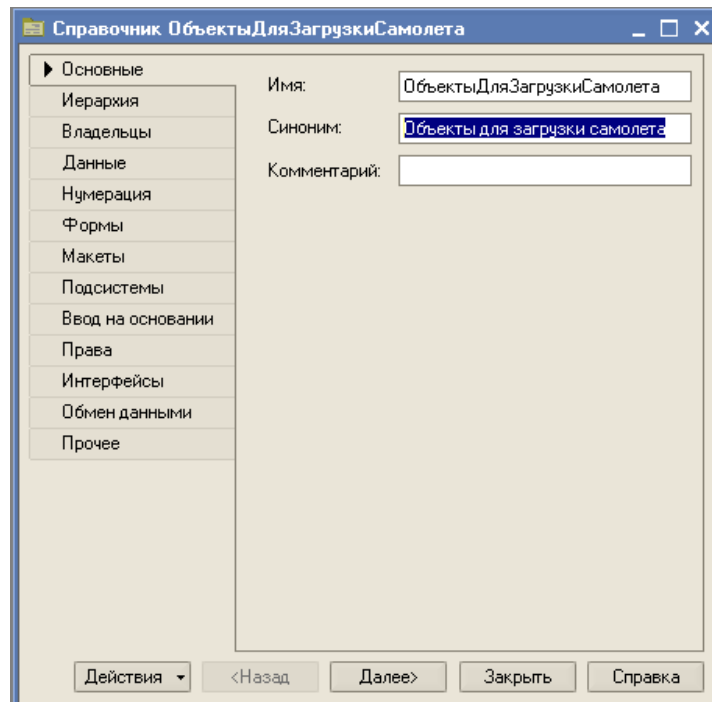


Рис.17. Создание справочника «ОбъектыДляЗагрузкиСамолета»

16. Структура справочника определяет состав хранимой информации. Под структурой в первую очередь понимаются состав реквизитов и табличных частей. Для нашей задачи на закладке «Данные» добавляем два реквизита:

«ВесОбъекта» (тип данных Строка(10,0)) и «Прибыль» (тип данных Строка(10,0)) (рис.18).

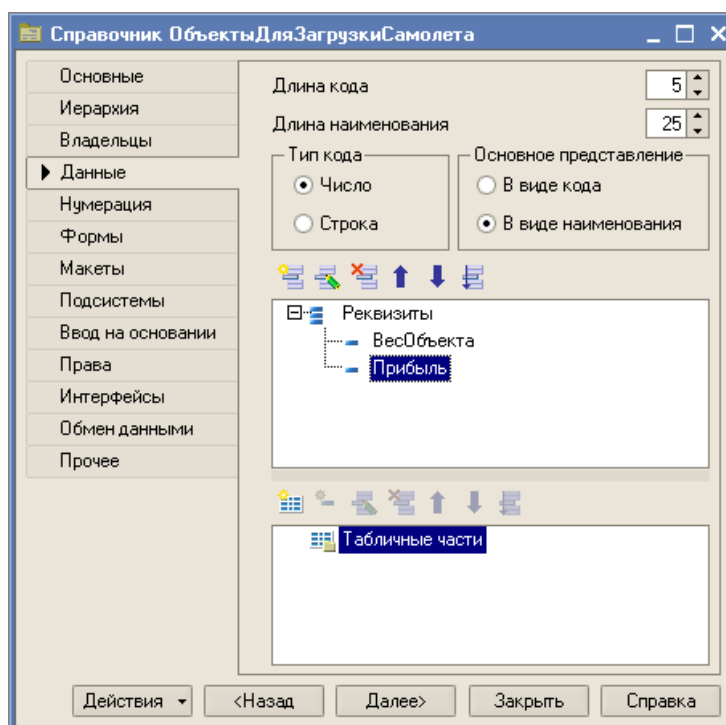


Рис.18. Структура справочника «ОбъектыДляЗагрузкиСамолета»

17. Для формирования любой печатной формы используется макет. Для отображения работы программы по расчету загрузки самолета нам понадобится три макета. Для этого на форме для объекта «ОбработкаЗагрузкаСамолета» выберем закладку «Макеты» и создадим три макета: «МакетИсходныеДанные», «МакетЭтапы Решения» и «МакетРешение» (рис.19).

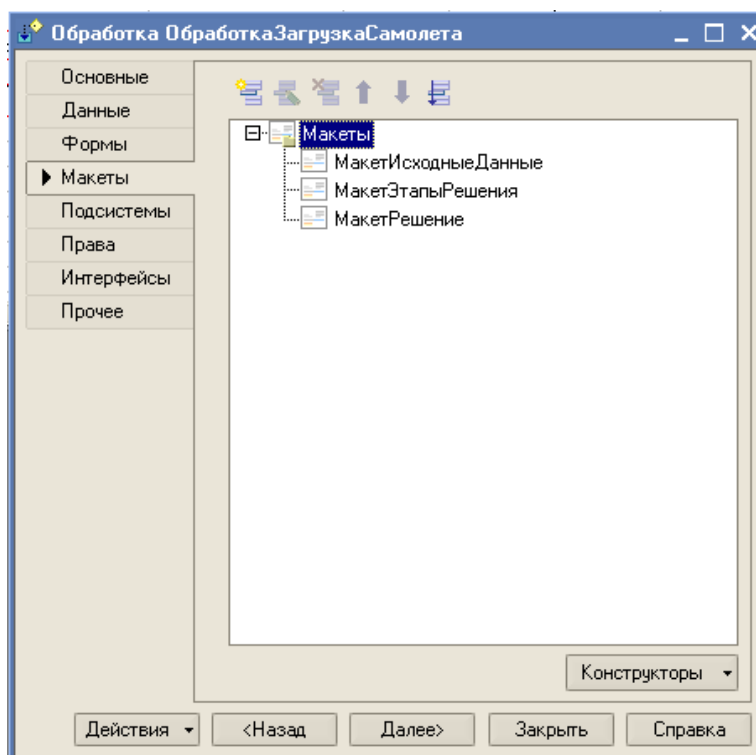


Рис.19. Создание макетов для формирования печатных форм

Макет является табличным элементом. Итоговая таблица строится на основе макета, как из строительных блоков. Для этого в макете создаются области, предназначенные для однократного или многократного вывода.

Сначала в табличный документ выводятся горизонтальные области Заголовок и Шапка. Далее несколько раз выводится область Строка, т.е. табличный документ наращивается снизу, хотя в *1С: Предприятия 8.0* можно присоединять области и справа. Затем один раз выводится область Подвал и формирование печатной формы документа завершено (рис.20).

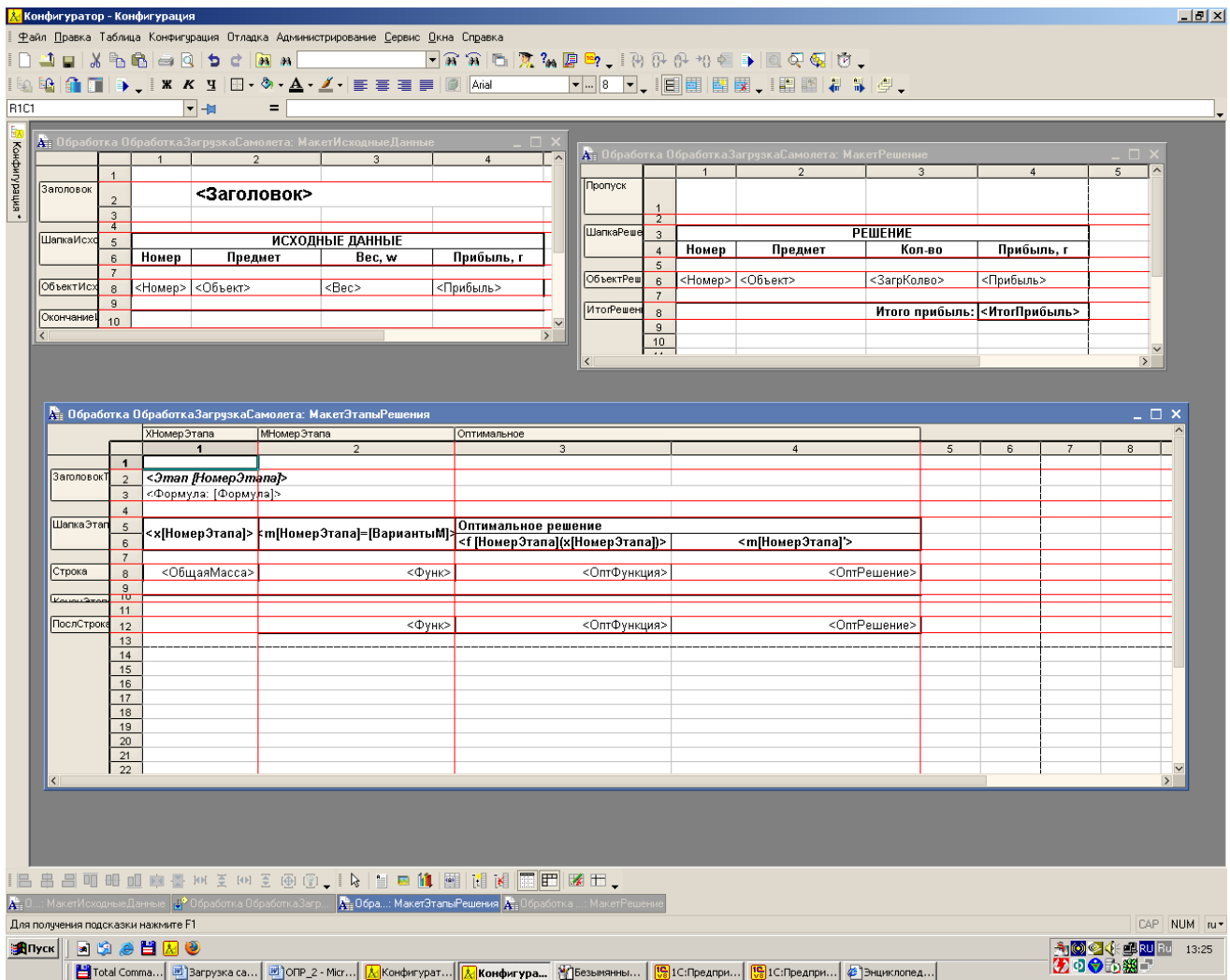


Рис.20. Вид макетов для формирования печатных форм

Макеты создаются и редактируются наподобие Excel документов. Работа с областями (слева и сверху от табличного поля их наименование) производится посредством кнопок 1, работа с ячейками – 2, а также с помощью контекстного меню (рис. 21).

Свойства ячеек (значения, оформление, положение текста и др.) задаются с помощью окна «Свойства».

Ячейка может быть трех типов (Заполнение):

- Текст
- Параметр
- Шаблон (объединение параметров и текста в одной ячейке)

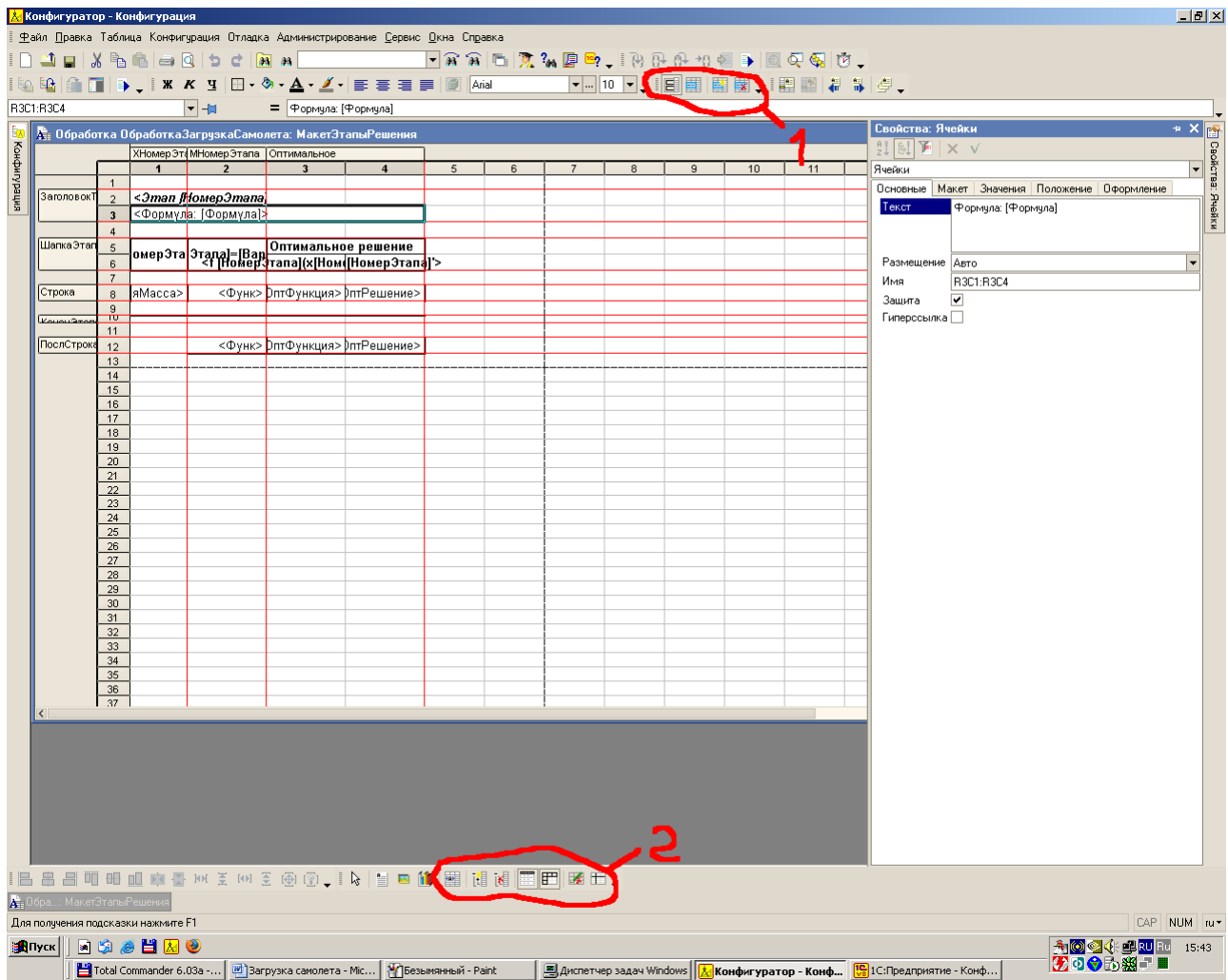


Рис.21. Редактирование макета

Если ячейка имеет заполнение текст, то при выводе области макета ячейка будет отображаться текст, написанный в ячейке.

Если ячейка имеет заполнение параметр, то при выводе области макета ячейка будет отображаться значение параметра, которое мы передаем в модуле.

Если ячейка имеет заполнение шаблон, то при выводе области макета ячейка будет отображаться следующим образом:

То, что стоит в скобках «< >», передается из модуля как параметр, все остальное - как простой текст.

18 . Чтобы ввести исходные данные, запустим конфигурацию на отладку. Для чего в главном меню выберем *Отладка* → *Начать отладку* (рис.22).

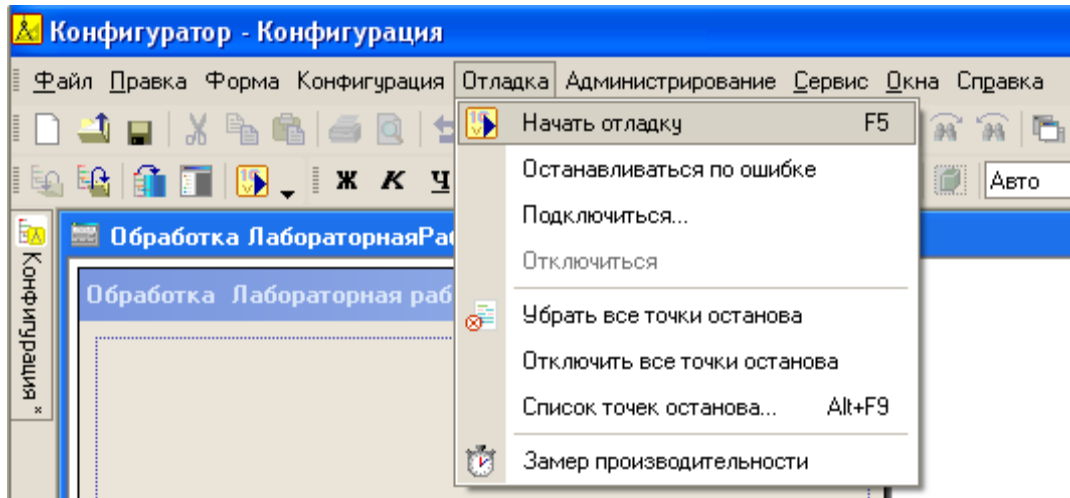


Рис.22. Запуск конфигурации на отладку

В появившемся окне нажимаем кнопку «Да» (рис.23).

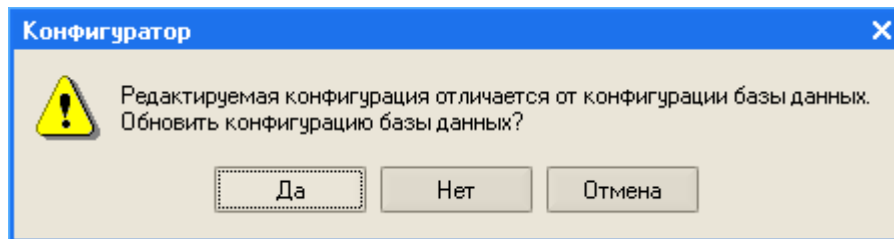


Рис.23. Обновление конфигурации

И принимаем изменения в структуре информации конфигурации (рис.24):

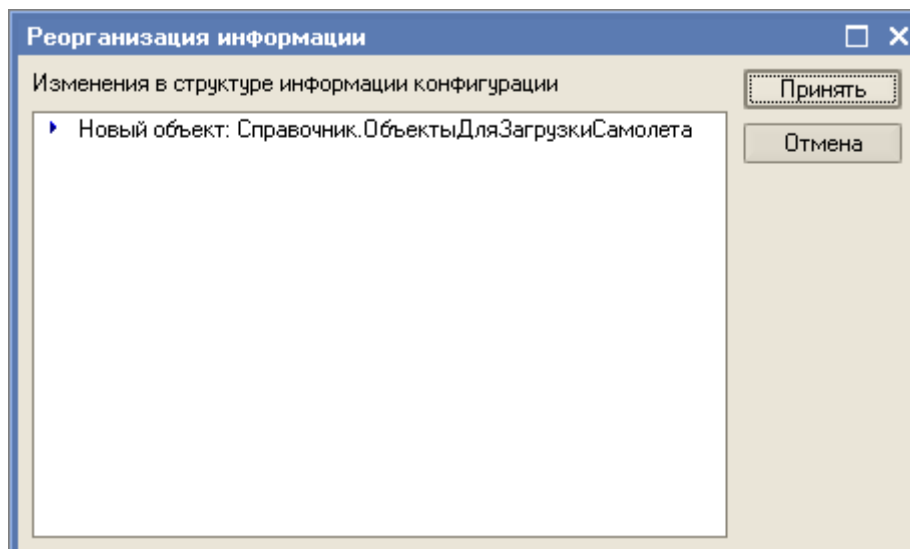


Рис.24. Изменения в структуре информации конфигурации

19. Для открытия справочника «ОбъектыДляЗагрузкиСамолета» (синоним «Объекты для загрузки самолета») в главном меню выбираем *Операции* → *Справочник...* (рис.25).

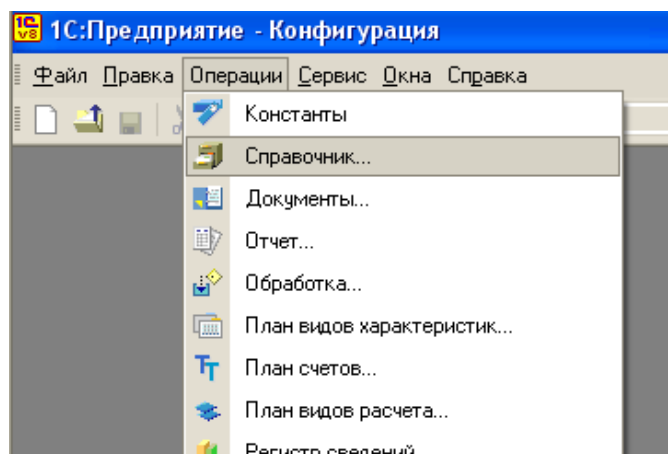


Рис.25. Открытие созданного справочника

Выбираем справочник «Объекты для загрузки самолета» (рис.26):

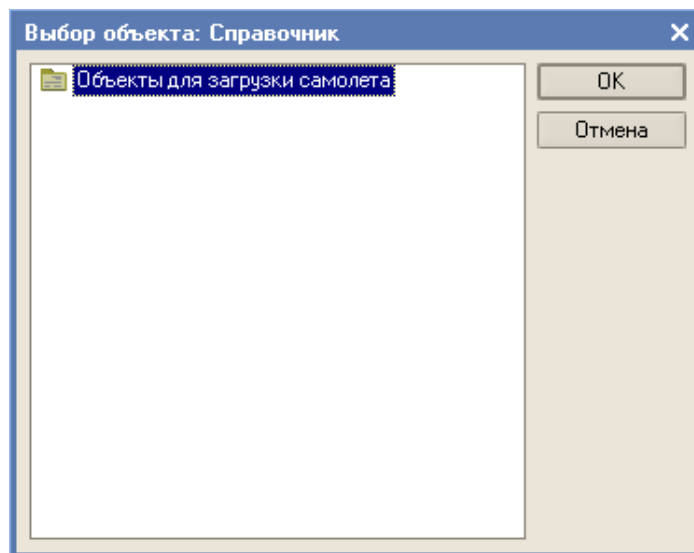
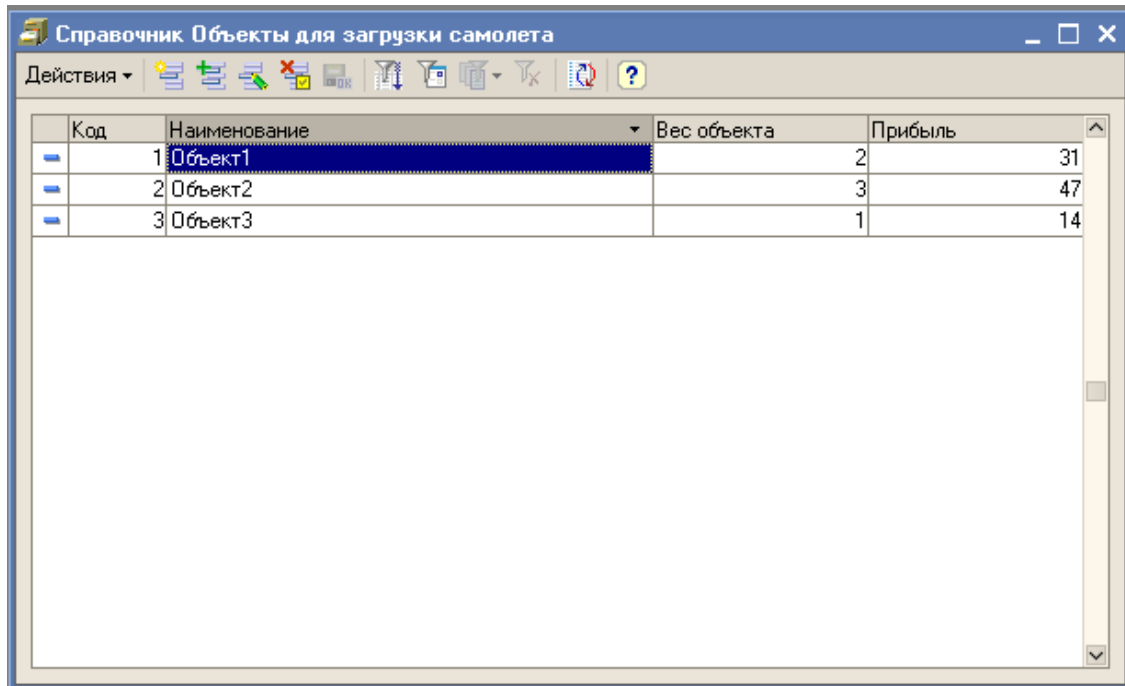


Рис.26. Выбор созданного справочника «ОбъектыДляЗагрузкиСамолета» (синоним «Объекты для загрузки самолета»)

20. Добавляем в справочник «ОбъектыДляЗагрузкиСамолета» (синоним «Объекты для загрузки самолета») наименование объектов (загружаемых предметов), их веса и прибыль по каждому объекту (предмету) соответственно (рис.27).



Код	Наименование	Вес объекта	Прибыль
1	Объект1	2	31
2	Объект2	3	47
3	Объект3	1	14

Рис.27. Заполнение справочника «ОбъектыДляЗагрузкиСамолета»
(синоним «Объекты для загрузки самолета»)

21. Откроем форму обработки «ОбработкаЗагрузкаСамолета» и перейдем на закладку «Модуль» (рис.28)

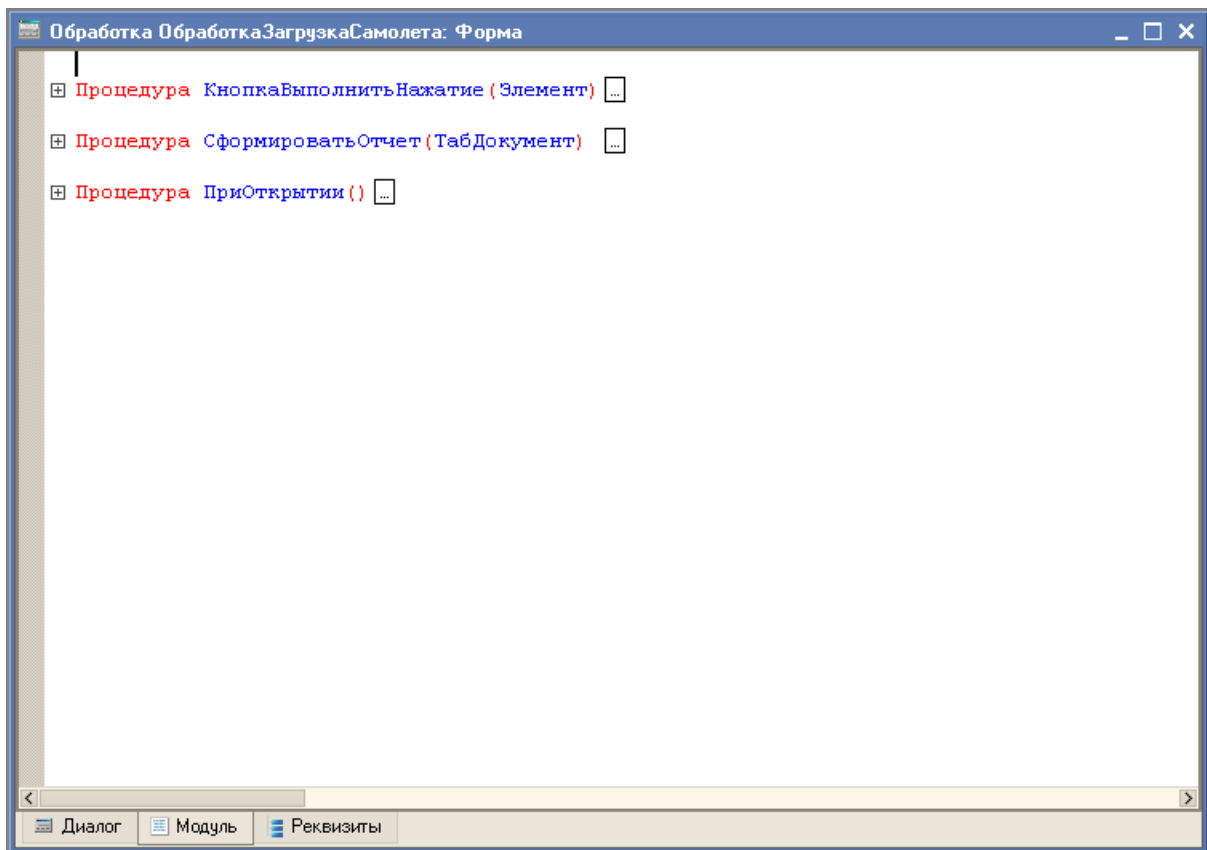


Рис.28. Вид формы «ОбработкаЗагрузкаСамолета» на закладке «Модуль»

22. Осталось добавить недостающие процедуры:

- В процедуре «ПриОткрытии» пишем следующую строку:

Процедура ПриОткрытии()

Грузоподъемность = 4;

КонецПроцедуры

- В процедуре «КнопкаВыполнитьНажатие» пишем следующую строку:

Процедура КнопкаВыполнитьНажатие(Элемент)

СформироватьОтчет(ЭлементыФормы.ДокументРезультат);

КонецПроцедуры

- Процедура «СформироватьОтчет» обрабатывает наши данные и выводит их в табличное поле на форме:

Процедура СформироватьОтчет(ТабДокумент)

ТабДокумент.Очистить();

//Создаем новый табличный документ, получаем макет и его области

Макет = ПолучитьМакет("МакетИсходныеДанные");

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

ОбластьШапкаИсходныеДанные = Макет.ПолучитьОбласть("ШапкаИсходныеДанные");

ОбластьОбъектИсходныеДанные = Макет.ПолучитьОбласть("ОбъектИсходныеДанные");

```

ОбластьОкончаниеИсходныхДанных = Макет.ПолучитьОбласть("ОкончаниеИсходныхДанных");
//Задаем значение параметра в ОбластьЗаголовков, выводим ОбластьЗаголовков и
//ОбластьШапкаИсходныеДанные
ОбластьЗаголовков.Параметры.Заголовков = "Задача о загрузке самолета";
ТабДокумент.Вывести(ОбластьЗаголовков);
ТабДокумент.Вывести(ОбластьШапкаИсходныеДанные);

//Достаем наши значения для каждого из предметов загрузки из справочника
ОбъектыДляЗагрузкиСамолета
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
|      ОбъектыДляЗагрузкиСамолета.Наименование КАК Объект,
|      ОбъектыДляЗагрузкиСамолета.ВесОбъекта КАК Вес,
|      ОбъектыДляЗагрузкиСамолета.Прибыль
|ИЗ
|      Справочник.ОбъектыДляЗагрузкиСамолета КАК ОбъектыДляЗагрузкиСамолета
|
|УПОРЯДОЧИТЬ ПО
|      Вес УБЫВ.";

КоличествоОбъектов = 0;
РезультатИсходныеДанные = Запрос.Выполнить().Выгрузить();
Для каждого СтрокаИсходныеДанные Из РезультатИсходныеДанные Цикл
    КоличествоОбъектов = КоличествоОбъектов + 1;
    ОбластьОбъектИсходныеДанные.Параметры.Номер = КоличествоОбъектов;
    ОбластьОбъектИсходныеДанные.Параметры.Объект = СтрокаИсходныеДанные.Объект;
    ОбластьОбъектИсходныеДанные.Параметры.Вес = СтрокаИсходныеДанные.Вес;
    ОбластьОбъектИсходныеДанные.Параметры.Прибыль = СтрокаИсходныеДанные.Прибыль;
    ТабДокумент.Вывести(ОбластьОбъектИсходныеДанные);
КонецЦикла;
ТабДокумент.Вывести(ОбластьОкончаниеИсходныхДанных);
НомерПослОбъекта = КоличествоОбъектов;
ТабДокумент.ФиксацияСверху = НомерПослОбъекта + 5;

МассивФункций = Новый Массив;
МассивМаксФункций = Новый Массив;
МассивТаблиц = Новый Массив;

Макет = ПолучитьМакет("МакетЭтапыРешения");
ОбластьЗаголовковТаблицы      = Макет.ПолучитьОбласть("ЗаголовковТаблицы");
ОбластьШапкаЭтапаХНомерЭтапа  = Макет.ПолучитьОбласть("ШапкаЭтапа|ХНомерЭтапа");
ОбластьШапкаЭтапаМНомерЭтапа  = Макет.ПолучитьОбласть("ШапкаЭтапа|МНомерЭтапа");
ОбластьШапкаЭтапаОптимальное  = Макет.ПолучитьОбласть("ШапкаЭтапа|Оптимальное");

ОбластьСтрокаЭтапаХНомерЭтапа  = Макет.ПолучитьОбласть("Строка|ХНомерЭтапа");
ОбластьСтрокаЭтапаМНомерЭтапа  = Макет.ПолучитьОбласть("Строка|МНомерЭтапа");
ОбластьСтрокаЭтапаОптимальное  = Макет.ПолучитьОбласть("Строка|Оптимальное");

ОбластьКонецЭтапаХНомерЭтапа  = Макет.ПолучитьОбласть("КонецЭтапа|ХНомерЭтапа");
ОбластьПослСтрокаЭтапаМНомерЭтапа  = Макет.ПолучитьОбласть("ПослСтрока|МНомерЭтапа");
ОбластьПослСтрокаЭтапаОптимальное  = Макет.ПолучитьОбласть("ПослСтрока|Оптимальное");

Пока КоличествоОбъектов <> 0 Цикл
    Вес = РезультатИсходныеДанные.Получить(КоличествоОбъектов-1).Получить(1);
    Прибыль = РезультатИсходныеДанные.Получить(КоличествоОбъектов-1).Получить(2);
    //Вывод заголовка этапа
    ОптимРешение = 0;
    НомерЭтапа = КоличествоОбъектов;
    Если КоличествоОбъектов <> НомерПослОбъекта тогда
        ДобавкаНаПечать = "+"f+(КоличествоОбъектов+1)+"(x"+КоличествоОбъектов+"-
"+Вес+"*m"+КоличествоОбъектов+)";
    Иначе
        ДобавкаНаПечать = "";
    КонецЕсли;
    ФункцияНаПечать =
    "f"+КоличествоОбъектов+"(x"+КоличествоОбъектов+)="+Прибыль+"*m"+КоличествоОбъектов+""+Добавка
НаПечать;
    ОбластьЗаголовковТаблицы.Параметры.НомерЭтапа = НомерЭтапа;
    ОбластьЗаголовковТаблицы.Параметры.Формула = ФункцияНаПечать;

```

```

ТабДокумент.Вывести(ОбластьЗаголовокТаблицы);

//Выводим ОбластьШапкаЭтапаХНомерЭтапа
ОбластьШапкаЭтапаХНомерЭтапа.Параметры.НомерЭтапа = НомерЭтапа;
ТабДокумент.Вывести(ОбластьШапкаЭтапаХНомерЭтапа);

ТаблицаЭтапа = Новый ТаблицаЗначений;
ТаблицаЭтапа.Колонки.Добавить("Икс");
ТаблицаЭтапа.Колонки.Добавить("Функ");
ТаблицаЭтапа.Колонки.Добавить("ОптимРешение");
МаксКоличество = Цел(Грузоподъемность/Вес);
ОбщаяМасса = 0;
Пока ОбщаяМасса <= Грузоподъемность Цикл
    Ммм = 0;
    МаксФунк = 0;
    Если НомерПослОбъекта = КоличествоОбъектов и ОбщаяМасса = 0 тогда
        Пока МММ <= МаксКоличество*100 Цикл
            МассивФункций.Добавить(0);
            МассивМаксФункций.Добавить(0);
            МММ = МММ+1;
        КонецЦикла;
    КонецЕсли;
    //Выводим ОбластьШапкаЭтапаМНомерЭтапа сколько у нас МММ
    МММ = 0;
    Если ОбщаяМасса = 0 тогда
        Пока МММ<= МаксКоличество Цикл
            ОбластьШапкаЭтапаМНомерЭтапа.Параметры.НомерЭтапа =
НомерЭтапа;
                ОбластьШапкаЭтапаМНомерЭтапа.Параметры.ВариантыМ = МММ;
                ТабДокумент.Присоединить(ОбластьШапкаЭтапаМНомерЭтапа);
                МММ = МММ+1;
            КонецЦикла;
        КонецЕсли;
        Если ОбщаяМасса = 0 тогда
            ОбластьШапкаЭтапаОптимальное.Параметры.НомерЭтапа = НомерЭтапа;
            ТабДокумент.Присоединить(ОбластьШапкаЭтапаОптимальное);
        КонецЕсли;
        //Выводим ОбластьСтрокаЭтапаХНомерЭтапа сколько у нас ОбщаяМасса
        ОбластьСтрокаЭтапаХНомерЭтапа.Параметры.ОбщаяМасса = ОбщаяМасса;
        ТабДокумент.Вывести(ОбластьСтрокаЭтапаХНомерЭтапа);
        Ммм = 0;
        Пока МММ <= МаксКоличество Цикл
            Если ОбщаяМасса >= Вес*МММ тогда

                Функ = Прибыль*МММ + МассивМаксФункций[ОбщаяМасса-Вес*МММ];

                Если ОбщаяМасса = Грузоподъемность тогда
                    ОбластьПослСтрокаЭтапаМНомерЭтапа.Параметры.Функ =
Функ;
            ТабДокумент.Присоединить(ОбластьПослСтрокаЭтапаМНомерЭтапа);
            Иначе
                ОбластьСтрокаЭтапаМНомерЭтапа.Параметры.Функ = Функ;

            ТабДокумент.Присоединить(ОбластьСтрокаЭтапаМНомерЭтапа);
            КонецЕсли;
            Иначе
                Функ = "-";
                Если ОбщаяМасса = Грузоподъемность тогда
                    ОбластьПослСтрокаЭтапаМНомерЭтапа.Параметры.Функ =
Функ;
            ТабДокумент.Присоединить(ОбластьПослСтрокаЭтапаМНомерЭтапа);
            Иначе
                ОбластьСтрокаЭтапаМНомерЭтапа.Параметры.Функ = Функ;

            ТабДокумент.Присоединить(ОбластьСтрокаЭтапаМНомерЭтапа);
            КонецЕсли;
        КонецЕсли;

```

```

        Если Функ <> "-" тогда
            Если Функ > МаксФунк Тогда
                МаксФунк = Функ;
                ОптимРешение = МММ;
            КонецЕсли;
        КонецЕсли;
        МММ = МММ+1;
    КонецЦикла;
    МассивФункций.Установить(ОбщаяМасса,МаксФунк);
    НоваяСтрока = ТаблицаЭтапа.Добавить();
    НоваяСтрока.Икс = ОбщаяМасса;
    НоваяСтрока.Функ = МаксФунк;
    НоваяСтрока.ОптимРешение = ОптимРешение;
    //Выводим оптимальные значения
    Если ОбщаяМасса = Грузоподъемность тогда
        ОбластьПослСтрокаЭтапаОптимальное.Параметры.ОптФункция=МаксФунк;

ОбластьПослСтрокаЭтапаОптимальное.Параметры.ОптРешение=ОптимРешение;
    ТабДокумент.Присоединить(ОбластьПослСтрокаЭтапаОптимальное);
    Иначе
        ОбластьСтрокаЭтапаОптимальное.Параметры.ОптФункция=МаксФунк;
        ОбластьСтрокаЭтапаОптимальное.Параметры.ОптРешение=ОптимРешение;
        ТабДокумент.Присоединить(ОбластьСтрокаЭтапаОптимальное);
    КонецЕсли;

    ОбщаяМасса = ОбщаяМасса+1;
    КонецЦикла;
    Индекс = 0;
    Для каждого ЭлементМассива из МассивФункций Цикл
        МассивМаксФункций.Установить(Индекс,МассивФункций.Получить(Индекс));
        Индекс = Индекс+1;
    КонецЦикла;
    МассивТаблиц.Добавить(ТаблицаЭтапа);
    КоличествоОбъектов = КоличествоОбъектов - 1;
    ТабДокумент.Вывести(ОбластьКонецЭтапаХНомерЭтапа);
КонецЦикла;

//поиск и вывод оптимального решения по массиву МассивТаблиц
Макет = ПолучитьМакет("МакетРешение");
ОбластьПропуск = Макет.ПолучитьОбласть("Пропуск");
ОбластьШапкаРешение = Макет.ПолучитьОбласть("ШапкаРешение");
ОбластьСтрокаРешение = Макет.ПолучитьОбласть("ОбъектРешение");
ОбластьИтогРешение = Макет.ПолучитьОбласть("ИтогРешение");

ТабДокумент.Вывести(ОбластьПропуск);
ТабДокумент.Вывести(ОбластьШапкаРешение);

ТаблицаРешение = Новый ТаблицаЗначений;
ТаблицаРешение.Колонки.Добавить("Предмет");
ТаблицаРешение.Колонки.Добавить("Количество");
ТаблицаРешение.Колонки.Добавить("Выручка");

// Получаем оптимальное значение функции на первом этапе
МаксФ = 0;
Для каждого СтрокаТаблицыПервогоЭтапа из МассивТаблиц[НомерПослОбъекта-1] Цикл
    Загр = СтрокаТаблицыПервогоЭтапа[0];
    Ф = СтрокаТаблицыПервогоЭтапа[1];
    Опт = СтрокаТаблицыПервогоЭтапа[2];
    Если Ф > МаксФ Тогда
        МаксФ = Ф;
        Оптимум = Опт;
        Загрузили = Загр;
    КонецЕсли;
КонецЦикла;
Строка = 0;
НоваяСтрока = ТаблицаРешение.Добавить();
НоваяСтрока.Предмет = РезультатИсходныеДанные.Получить(НомерПослОбъекта-1).Получить(0);
НоваяСтрока.Количество = Оптимум;
НоваяСтрока.Выручка = Оптимум*РезультатИсходныеДанные.Получить(0).Получить(2);

```

```

ИндексПоМассиву = НомерПослОбъекта-2;
ИндексПоДанным = 0;
    Прибыль = 0;
Пока ИндексПоМассиву >= 0 Цикл
    Вес= РезультатИсходныеДанные.Получить(ИндексПоДанным).Получить(1);
    Прибыль = РезультатИсходныеДанные.Получить(ИндексПоДанным).Получить(2);
    Таблица = МассивТаблиц[ИндексПоМассиву];
    Загр =Таблица[Загрузили-Вес*Оптимум][0];
    Ф = Таблица[Загрузили-Вес*Оптимум][1];
    Опт = Таблица[Загрузили-Вес*Оптимум][2];
    МаксФ = Ф;
    Оптимум = Опт;
    Загрузили = Загр;
    НоваяСтрока = ТаблицаРешение.Добавить();
    НоваяСтрока.Предмет =
РезультатИсходныеДанные.Получить(ИндексПоМассиву).Получить(0);
    НоваяСтрока.Количество = Оптимум;
    НоваяСтрока.Выручка = МаксФ;
    ИндексПоМассиву = ИндексПоМассиву - 1;
    ИндексПоДанным = ИндексПоДанным + 1;
КонецЦикла;

Индекс = НомерПослОбъекта - 1;
Номер = 1;
ИтогПрибыль = 0;
Пока Индекс >= 0 Цикл
    ОбластьСтрокаРешение.Параметры.Номер = Номер;
    ОбластьСтрокаРешение.Параметры.Объект = ТаблицаРешение[Индекс][0];
    ОбластьСтрокаРешение.Параметры.ЗагрКолво = ТаблицаРешение[Индекс][1];
Прибыль =
ТаблицаРешение[Индекс][1]*РезультатИсходныеДанные.Получить(ИндексПоДанным).Получить(2);
    ОбластьСтрокаРешение.Параметры.Прибыль = Прибыль;
    ТабДокумент.Вывести(ОбластьСтрокаРешение);
    ИтогПрибыль = ИтогПрибыль + Прибыль;
    Индекс = Индекс - 1;
    Номер= Номер+1;
    ИндексПоДанным = ИндексПоДанным – 1;
КонецЦикла;

ОбластьИтогРешение.Параметры.ИтогПрибыль = ИтогПрибыль;
ТабДокумент.Вывести(ОбластьИтогРешение);

ТабДокумент.ОтображатьСетку = Ложь;
ТабДокумент.АвтоМасштаб = Истина;
ТабДокумент.ОриентацияСтраницы = ОриентацияСтраницы.Ландшафт;
ТабДокумент.Показать("Загрузка самолета");

КонецПроцедуры

```

23. Для проверки работы программы запускаем конфигурацию на отладку и открываем обработку (в главном меню *Операции* → *Обработки...*). Результаты работы программы приведены на рис.29.

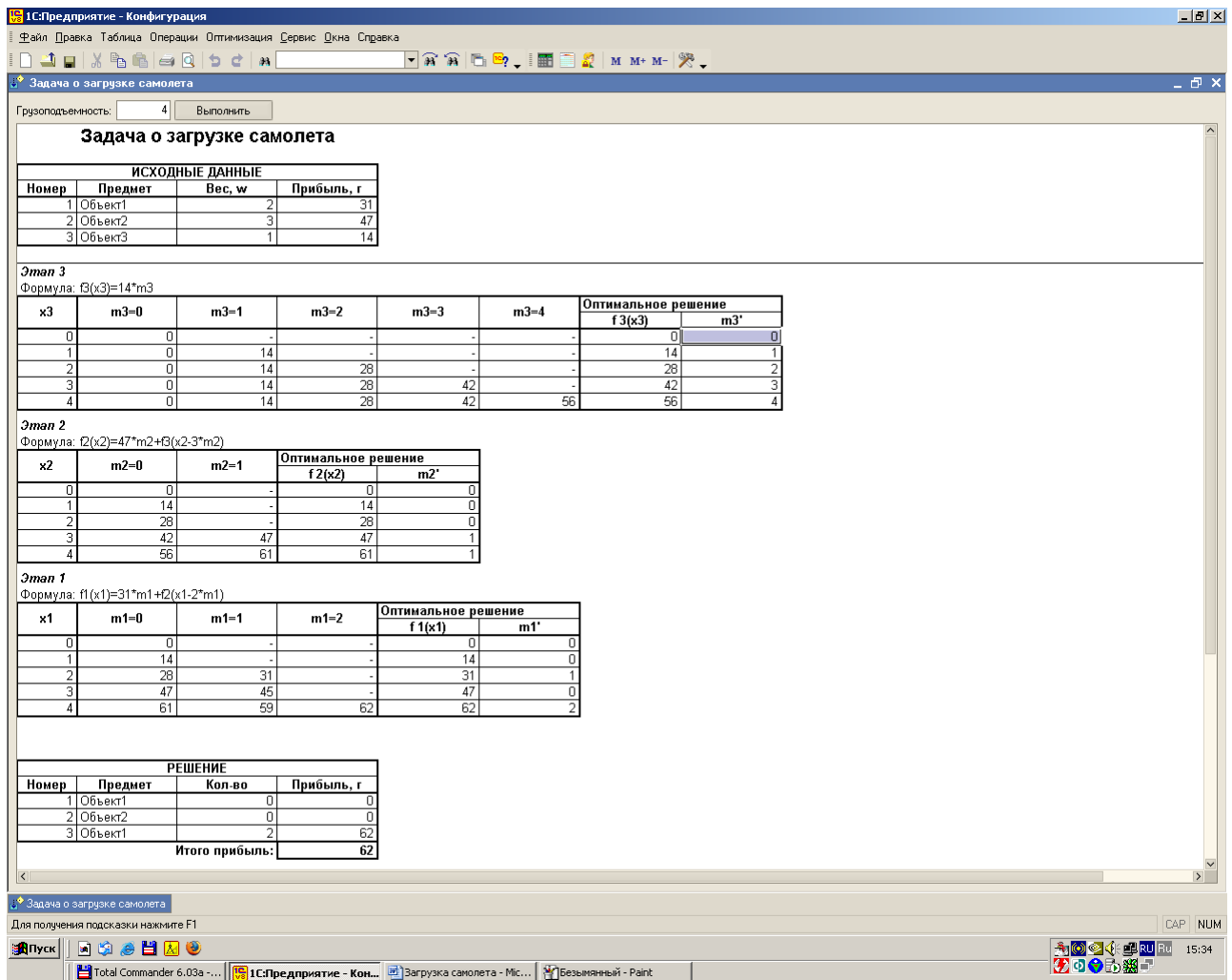


Рис.29. Результаты работы программы

Задание

Разработать конфигурацию для решения одной из нижеперечисленных задач.

1. Для разобранный выше примера определить оптимальное решение, предполагая, что максимальная грузоподъемность самолета составляет 3 тонны.

2. Решите задачу о загрузке для каждого из следующих случаев:

a) $w_1 = 4, r_1 = 70, w_2 = 1, r_2 = 20, w_3 = 2, r_3 = 40, W = 6.$

b) $w_1 = 1, r_1 = 30, w_2 = 2, r_2 = 60, w_3 = 3, r_3 = 80, W = 4.$

3. Турист собирается в путешествие по дикой местности и должен упаковать в рюкзак предметы трех наименований: пищу, средства первой

помощи и одежду. Объем рюкзака составляет 3 кубических фута. Каждая единица пищи занимает 1 кубический фут, упаковка средств первой помощи — четверть кубического фута, а отдельный предмет одежды — примерно половину кубического фута. Турист определил свои предпочтения весовыми коэффициентами 3, 4 и 5 — для пищи, средств первой помощи и одежды соответственно. Это означает, что одежда является самым ценным предметом среди остальных. Опыт подсказывает туристу, что он должен взять не менее одного предмета каждого наименования и не более двух комплектов средств первой помощи. Сколько единиц каждого наименования возьмет турист в поход?

4. Студент должен выбрать 10 факультативных курсов на четырех различных факультетах, причем на каждом факультете должен быть выбран по меньшей мере один курс. Эти курсы распределяются между факультетами таким образом, чтобы максимизировать объем "знаний". Студент оценивает знания по шкале в сто баллов и приходит к выводам, представленным в следующей таблице.

Какие курсы следует выбрать студенту?

Факультет	Номер курса						
	1	2	3	4	5	6	≥ 7
1	25	50	60	80	100	100	
2	100						
3	20	70	90	100	100	100	
4	100						
	40	60	80	100	100	100	
	100						
	10	20	30	40	50	60	
	70						

5. У меня во дворе имеется небольшой огород 10 x 20 футов. Этой весной я собираюсь посадить овощи трех видов: помидоры, зеленые бобы и кукурузу. Огород разбит на ряды, длина которых равна 20 футам. Кукуруза и помидоры занимают ряды шириной 2 фута, а зеленые бобы — 3 фута. Помидоры мне нравятся больше, а бобы меньше, По 10-балльной

шкале предпочтений я бы присвоил помидорам 10 баллов, кукурузе— 7 баллов и зеленым бобам - 3 балла. Независимо от моих предпочтений, жена настаивает, чтобы я посадил не менее одного ряда зеленых бобов и не более двух рядов помидоров. Сколько рядов каждого вида овощей следует мне посадить?

6. "Жилище для Человечества" — прекрасная благотворительная организация, которая строит дома для бедствующих семей силами добровольцев: Такая семья может выбрать себе дом из трех типоразмеров: 1000, 1100 и 1200 квадратных футов. Дом каждого типоразмера требует выполнения определенного объема работ силами добровольцев. Филиал организации в городе Файтвилл получил пять заявок на предстоящие шесть месяцев. Комитет по надзору дает каждой заявке в численном виде, принимая во внимание различные факторы. Более высокая оценка означает более острую потребность в жилье. В течение предстоящих шести месяцев филиал организации в этом городе может привлечь к работе максимум 23 добровольца. Следующая таблица содержит оценку каждой заявки и необходимое число добровольцев для ее выполнения. Какие заявки следует утвердить комитету?

Заявка	Размер дома (футов ²)	Оценка	Необходимое число добровольцев
1	1200	78	7
2	1000	64	4
3	1100	68	6
4	1000	62	5
5	1200	85	8

7. Шериф округа Вашингтон принимает участие в переизбрании на следующий срок. Денежные средства на предвыборную кампанию составляют примерно 10000 долларов. Хотя комитет по переизбранию хотел бы провести кампанию во всех пяти избирательных участках округа, ограниченность денежных средств предписывает действовать по-другому. Приведенная ниже таблица содержит данные о числе избирателей и

денежных средствах, необходимых для проведения успешной кампании по каждому избирательному участку. Каждый участок может либо использовать все предназначенные деньги, либо вовсе их не использовать. Как следует распределить денежные средства?

Участок средства	Число изобретателей	Необходимые
1	3100	3500
2	2600	2500
3	3500	4000
4	2800	3000
5	2400	2000

8. Конструируется электронный прибор, состоящий из трех основных компонентов. Все компоненты соединены последовательно, поэтому выход из строя одного из них влечет за собой отказ всего прибора. Надежность (вероятность безаварийной работы) прибора можно повысить путем дублирования каждого компонента. Конструкция прибора допускает использование одного или двух резервных (параллельных) блоков, т.е. каждый компонент прибора может содержать до трех блоков, соединенных параллельно. Следующая таблица содержит данные о надежности r и стоимости компонентов прибора.

Число параллельных блоков	Компонент1		Компонент2		
	r_1	$c_1(\$)$	r_2	$c_2(\$)$	r_3
1	0,6	1000	0,7	3000	0,5
2	0,8	2000	0,8	5000	0,7
3	0,9	3000	0,9	6000	0,9
	0,6	4000			
	0,9	5000			

Общая сумма, выделенная на конструирование прибора, равна 10000 долларов. Как следует сконструировать прибор? (Совет. Наша задача состоит в максимизации надежности $r_1 r_2 r_3$ прибора. Это значит, что целевая функция, является мультипликативной, а не аддитивной).