

Министерство образования и науки Российской Федерации

Владивостокский государственный университет  
экономики и сервиса

---

# **ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ**

Руководство по выполнению курсовых работ

Владивосток  
Издательство ВГУЭС  
2005

ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ: Руководство по выполнению курсовых работ / Сост. С.Л. Бедрина, О.Б. Богданова. – Владивосток: Изд-во ВГУЭС, 2005. – 64 с.

Руководство по выполнению курсовой работы по дисциплине «Информатика и программирование» составлено в соответствии с требованиями государственного стандарта России и предусмотрено учебным планом специальности. Содержит организационно-методические указания и рекомендации для выполнения курсовой работы.

Предназначено студентам специальности 351400 «Прикладная информатика в экономике», изучающих ее в качестве дисциплины естественнонаучного цикла.

Печатается по решению РИСО ВГУЭС

ББК 32.973/2-018

© Издательство Владивостокского государственного университета экономики и сервиса, 2005

## ВВЕДЕНИЕ

Visual Basic for Applications (VBA) – это сочетание одного из самых простых языков программирования и разнообразных возможностей пакета MS Office. С помощью VBA возможно легко и быстро создавать разнообразные приложения, используя единые для всех офисных программ среду разработки и язык. VBA содержит графическую среду, позволяющую наглядно конструировать экранные формы и управляющие элементы. Visual Basic for Applications позволяет с легкостью решать многие задачи, выполнение которых обычными средствами невозможно.

Для специалистов в области экономики и финансов особый интерес представляют разработки в среде MS Excel. MS Excel является современным компьютерным средством для оперативного анализа экономической ситуации и выбора оптимального решения из возможных альтернатив.

Предлагаемое пособие предназначено для изучения возможностей языка VBA по созданию конкретных приложений в Excel. В работе рассматриваются основные понятия и приёмы программирования на VBA и принципы разработки приложений с помощью методов объектно-ориентированного программирования.

В предлагаемой работе содержатся варианты индивидуальных заданий для студентов и рассматривается пример создания приложения на языке VBA, который поможет лучше овладеть основами программирования и разобраться с приемами обработки информации в прикладных программах.

# 1. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ

## 1.1. Цель и задачи курсовой работы

Целью выполнения курсовой работы является осмысление и углубление знаний по данной дисциплине, развитие навыков самостоятельной работы по анализу и обработке информации, использованию средств Microsoft Office для создания приложений пользователя на примере конкретной предметной области.

В ходе выполнения курсовой работы у студента должно формироваться представление о принципах и способах обработки информации в прикладных системах.

В ходе достижения цели решаются следующие задачи:

- развитие логического и алгоритмического мышления;
- изучение принципов работы прикладного программного обеспечения;
- выработка умения самостоятельного решения задач по анализу информации и выбору метода ее обработки;
- получение навыков в разработке прикладных программ; программировании и отладке программ, а также тестировании создаваемых программных модулей;
- изучение перспектив развития технологий создания ПО.

## 1.2. Требования к выполнению курсовой работы

Для выполнения курсовой работы следует ознакомиться с предметной областью для которой решается поставленная задача. Следует исследовать технологические процессы предприятия и информационные потоки, которые задействованы при решении задачи. В соответствии с вариантом задания студент должен собрать и обработать необходимую информацию, проверить ее достоверность и согласованность. На основании собранной информации студент должен продумать алгоритм решения задачи, разработать формы входных и выходных документов. Используя язык программирования Visual Basic for Application, студент должен составить программу, которая реализует алгоритм решения задачи.

## 1.3. Объем и содержание курсовой работы

Курсовая работа выполняется в форме пользовательского приложения, выполненного в среде MS Excel и VBA, и пояснительной записки к работе, которая должна иметь следующую структуру.

Аннотация

Содержание

1. Описание предметной области
  - 1.1. Постановка задачи
  - 1.2. Описание функций предметной области (описание задач, которые решаются на рабочем месте)
  - 1.3. Технологии решения задач
2. Разработка информационного обеспечения задачи
  - 2.1. Описание входных документов
  - 2.2. Описание выходных документов
  - 2.3. Дерево диалога (описание меню и связей между формами)
  - 2.4. Руководство пользователя
    - 2.4.1. Функции программы
    - 2.4.2. Установка программы
    - 2.4.3. Описание технологии работы (по операциям: ввод, корректировка, поиск и т.д.)
    - 2.4.4. Сообщение об ошибках.

#### Приложения

В **аннотации** в краткой и сжатой форме излагается содержание курсовой работы.

В **содержании** представляется структура работы в соответствии с выбранной темой. Указывается страница, с которой начинается каждый пункт.

В **первом разделе** в соответствии с выбранной темой индивидуального задания дается краткая характеристика предметной области, описываются основные задачи решаемые на рабочем месте и существующие технологии решения этих задач.

Содержание **второго раздела** пояснительной записки посвящено описанию информационного обеспечения поставленной задачи. В нем приводятся формы входных и выходных документов, разработанной системы, описывается созданное меню, пользовательские формы и связи между ними. Частью второго раздела является разработанное руководство пользователя, которое должно включать в себя описание функционального назначения программы, процесса ее установки, основных технологических операций и возможных ошибок в разработанной системе.

В качестве **Приложения** к пояснительной записке должны быть представлены код программы, формы, отчеты, входные документы.

#### **1.4. Техническое и программное обеспечение для выполнения курсовой работы**

Для выполнения курсовой работы необходимо наличие персонального компьютера не менее Pentium III-500МГц с оперативной памятью не менее 96 Мбайт и памятью на жестком диске 8 Гбайт и выше. На персональных компьютерах должно быть установлено следующее программное обеспечение: операционная система Windows 2000 и выше, а также интегрированный пакет MS Office 2000 и выше.

#### **1.5. Виды контроля знаний студентов и их отчетности по курсовой работе**

Написанная курсовая работа должна быть проверена преподавателем и защищена студентом с учетом высказанных замечаний. Проверка курсовой работы осуществляется поэтапно, согласно промежуточных аттестаций, проводимых в университете. К первой аттестации студент должен представить разработанные формы входных и выходных документов, структуры таблиц и меню программы. Ко второй аттестации студент представляет завершенную программу с набором тестовых данных. К защите студент представляет готовую программу и пояснительную записку. Студенты не прошедшие очередной этап аттестации в установленный срок к следующему этапу не допускаются.

## **2. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ. РАБОТА В РЕДАКТОРЕ VBA**

Разработка проекта на языке Visual Basic для приложений выполняется практически полностью в редакторе VBA. VBA имеет специальную интегрированную среду разработки.

Чтобы создать проект VBA, используется один из следующих методов:

- Автоматическая запись макроса, а затем его редактирование;
- Создание макроса в редакторе VBA;
- Создание проекта в редакторе VBA.

### **2.1. Автоматическая запись макроса и его редактирование**

В настоящее время почти каждое приложение, предназначенное для ведения деловой документации, имеет макроязык и средство записи макросов. Используя такие приложения, можно записать выполняемые пользователем действия, а затем в любое время воспроизвести нажатия клавиш и операции мышью, и таким образом автоматизировать работу.

При записи макроса основное приложение создает код, который можно просмотреть и отредактировать. Этот код также можно написать самостоятельно от начала до конца. Однако самый простой способ разработать макрос – это записать его, а затем изменить созданный код. В доказательство этого утверждения приведем два аргумента:

Для записи макроса и редактирования его в редакторе VBA:

1. Запустите в основном приложении средство автоматической записи макросов, выполнив для этого следующие действия:

- 1.1. выберите пункт меню Сервис ~ Макрос ~ Начать запись.
- 1.2. в окне диалога Запись макроса введите имя создаваемого макроса
- 1.3. выполните требуемую последовательность действий, например очистите диапазон ячеек A1:C1, и остановите запись, используя панель инструментов Остановить запись.

2. Выберите записанный макрос в списке макросов основного приложения, а затем нажмите кнопку изменить (Edit). Запускается редактор VBA, и в окне модуля выводится текст созданного макроса (рис. 1).

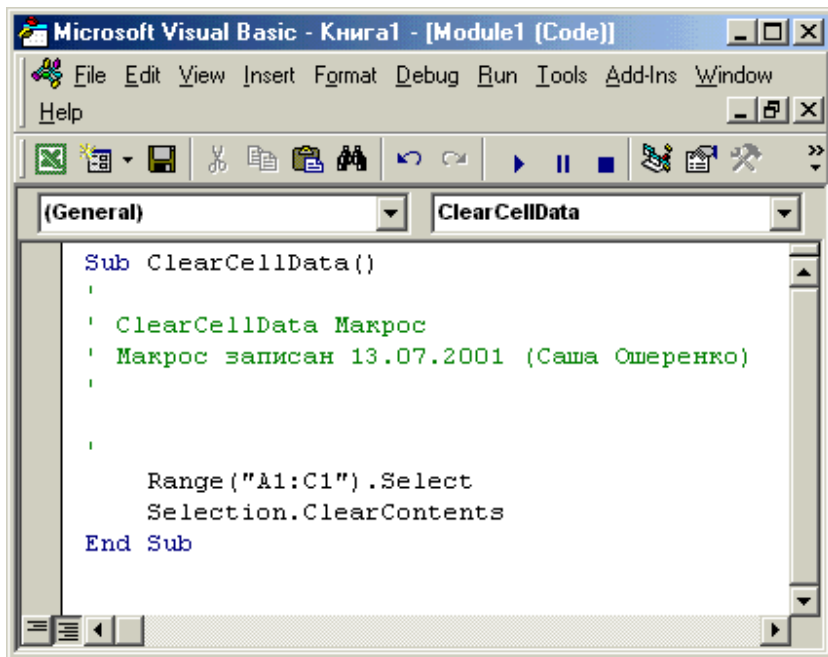


Рис. 1. Текст созданного макроса в окне модуля

## 2.2. Создание макроса в редакторе VBA

Чтобы создать проект VBA, можно написать макрос самостоятельно от начала до конца, не выполняя запись и последующее редактирование. Для создания макроса достаточно задать только его имя. С макросом пока еще не связан код и его необходимо ввести. Данный метод используется, если создаваемый макрос весьма сложен, либо использует формы или другие объекты. Кроме того, этот метод применяется для больших проектов.

После создания макроса можно добавить в него код, используя редактор VBA. Выберите макрос в списке макросов основного приложения, а затем нажмите кнопку *Изменить* для запуска редактора VBA и открытия окна модуля (рис. 2).

## 2.3. Создание проекта в редакторе VBA

Последний и наиболее быстрый метод создания проекта заключается в непосредственном запуске редактора VBA. Данный способ позволяет начать работу над проектом без предварительной разработки макроса. Рекомендуется использовать этот метод для больших проектов.



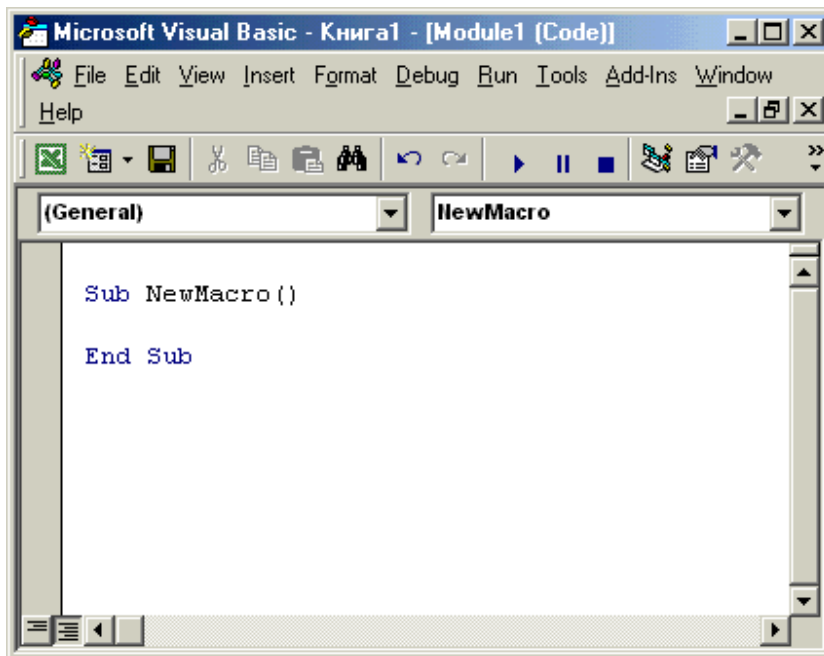


Рис. 2. Текст нового макроса

Чтобы запустить редактор VBA, выберите в основном приложении команду *Сервис\Макрос\Редактор Visual Basic* (Tools\Macro\Visual Basic Editor).

## 2.4. Описание редактора VBA

В редакторе VBA имеются различные инструменты, облегчающие разработку и управление приложениями. На рис. 3 показаны часто используемые средства. Ниже описывается их использование.

**Окно проекта** – это специальное окно редактора VBA, в котором выводятся все элементы проекта VBA. Элементы отображаются в виде иерархической структуры, включающей все формы, модули кода, и объекты основного приложения, например, рабочие листы, шаблоны, презентации, документы, базы данных и т. д.

В окне проекта можно с легкостью выбрать требуемый объект. Например, если требуется добавить кнопку в некоторую форму, выделите форму в окне проекта. После выбора выводятся относящиеся к объекту инструменты. Например, если выделить форму, то на экране отображается набор инструментов формы.

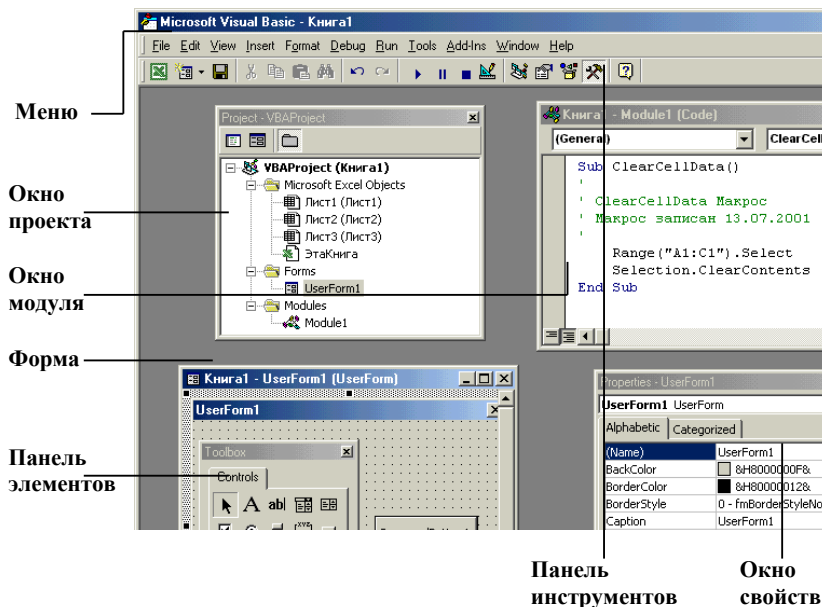


Рис. 3. Редактор VBA

Для выбора и редактирования объекта, выведенного в окне проекта:

- Дважды щелкните по имени объекта.
- Выберите объект, щелкните по нему правой кнопкой мыши, а затем либо воспользуйтесь командой *Объект* (View Object) или *Программа* (View Code). В контекстном меню доступна только та команда, которую можно выполнить. Например, команда *Объект* недоступна для модулей.

Для некоторых объектов доступны обе команды. Такими объектами обычно являются документы приложений, например документ Word, рабочая книга Excel и т. п., которые имеют как код, так и интерфейсные элементы.

**Окно свойств** используется для просмотра и задания свойств объектов проекта. Например, можно указать в окне свойств фон для формы, или изменить имя листа рабочей книги Excel. Окно предназначено для установки свойств и просмотра значений всех компонентов проекта, создаваемых форм и объектов основного приложения (документов или шаблонов Word, презентаций PowerPoint и т. д.).

Окно свойств разделено на два столбца название свойства выводится в левом столбце, а значение выбранного свойства – в правом.

Чтобы изменить значение свойства, выберите требуемое свойство в левом столбце, а затем задайте нужную величину в правом столбце. Для

некоторых свойств имеется ограниченный набор значений. Если при выборе свойства у правого края столбца значений отображается кнопка со стрелкой, то можно задать другую величину, нажав эту кнопку и выбрав из раскрывающегося списка требуемый элемент.

## 2.5. Добавление модуля в проект

В зависимости от масштаба и способа реализации проекта может понадобиться разработать ряд вспомогательных процедур, используемых во многих частях создаваемого приложения. Эти подпрограммы можно сохранить отдельно от остальной части приложения. Кроме того, имеется возможность сгруппировать взаимосвязанные процедуры в одном модуле. Для этих целей в VBA используются *модули*.

Чтобы добавить модуль в проект VBA:

1. Выберите проект, в который требуется добавить модуль, в окне проекта.
2. Выберите в меню редактора VBA команду *Вставка\Модуль* (Insert\Module). Выводится пустое окно модуля.
3. Выведите окно свойств, нажав клавишу <F4>, а затем задайте имя модуля.

### 2.5.1. Использование окна модуля

*Окно модуля* используется при написании любой программы VBA, будь это код макроса, запуск которого осуществляется при нажатии кнопки в созданной пользователем форме, или подпрограмма, которая входит в библиотеку служебных процедур разрабатываемого проекта (рис. 4).

Обратите внимание на два списка, расположенные под строкой заголовка окна. В первом списке выводятся все объекты модуля, а во втором – список процедур, связанных с выбранным объектом.

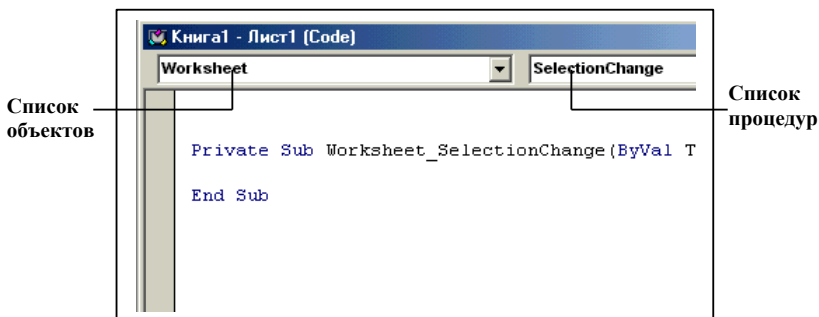


Рис. 4. Окно модуля

## 2.5.2. Редактирование кода в окне модуля

Количеством одновременно отображаемых в окне модуля процедур можно управлять. Обычно модуль включает несколько функций или подпрограмм. В его окне можно либо вывести все процедуры с отделением текста одной от текста другой горизонтальной линией (рис. 5), либо отобразить только одну процедуру модуля, нажав расположенную в нижнем левом углу кнопку Представление полного модуля (Full Module View) или кнопку Представление процедуры (Procedure View) соответственно.

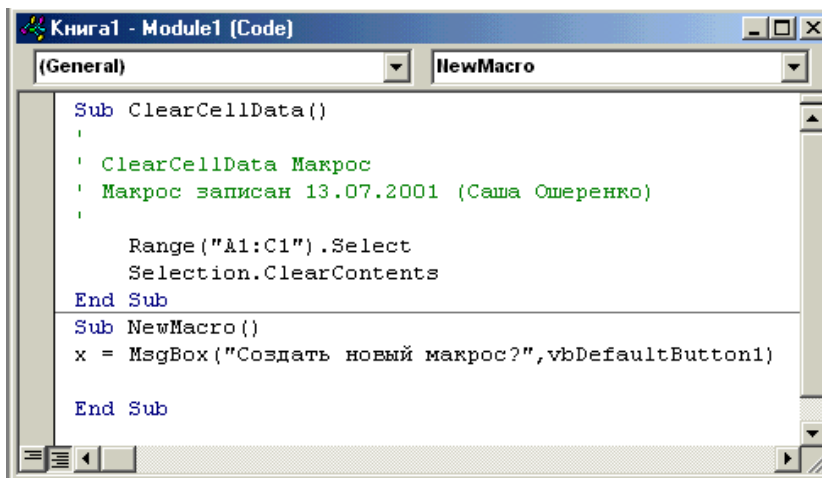


Рис. 5. Представление полного модуля

Код вводится непосредственно в окно модуля, так же как текст в любом текстовом редакторе.

## 2.6. Описание переменных и объектов

В редакторе VBA можно описать объекты и переменные, используя инструкцию *Dim*. Например:

```
Dim UserName as String
```

Выражение *Dim* позволяет задать использование объектов в разрабатываемом проекте:

```
Dim wbMyBudget as Workbook
```

В данном примере переменная *wbMyBudget* задает использование объекта *Workbook*, являющегося объектом Excel.

Если, VBA определяет, что описывается переменная или объект, то отображается специальное окно, в котором можно выбрать допустимый объект или тип. Например, после ввода:

```
Dim something As
```

редактор отображает список всех известных объектов и типов данных (рис. 6).

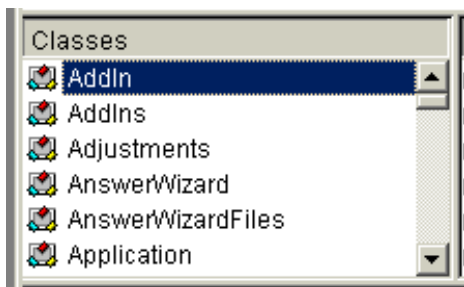


Рис. 6. Список допустимых типов

После вывода списка выберите требуемую переменную или объект и нажмите клавишу <Enter>. VBA подставляет выделенный элемент в выражение *Dim*.

Приведенный список отображается только тогда, когда установлен флажок *Список данных* (Auto List Members) в диалоговом окне *Параметры* (Options).

## 2.7. Использование Процедур

Редактор VBA упрощает использование процедур в программе, показывая их синтаксис. При вводе имени стандартной функции или подпрограммы он отображает в нужном порядке все требуемые аргументы, а также их тип.

На рис. 7 показан вывод параметров функции MsgBox() в редакторе VBA.

При отображении синтаксиса процедуры имена некоторых параметров выводятся жирным шрифтом, а остальные – заключаются в квадратные скобки. Параметры, выделенные жирным шрифтом, являются обязательными, например, параметр Prompt (текст, отображаемый в окне сообщения), а другие аргументы, например, параметр Title (текст заголовка окна сообщения) – дополнительными.

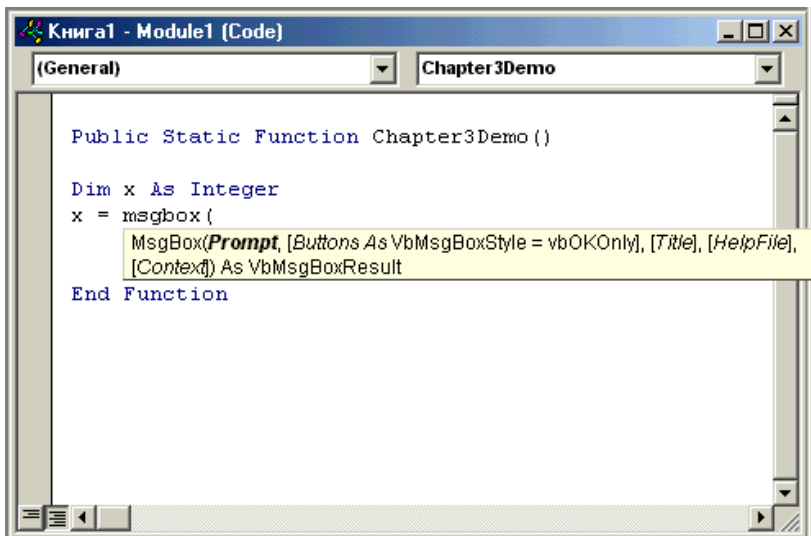


Рис. 7. Вывод параметров процедуры или функции в редакторе VBA

## 2.8. Описание свойств и методов

В VBA можно использовать объекты, включая меню, кнопки в форме, данные, а также компоненты основного приложения, например, рабочие листы. В программе необходимо указать имя объекта, а затем после точки указать операцию (метод) или некоторый атрибут этого объекта (свойство) (табл. 1):

Таблица 1

**Свойства и методы объектов**

Свойство	Методы
MyWorksheet.Name	MyWorksheet. Open
MyList.ItemCount	MyList.AddItem
ThisReport.Header	ThisReport. Print

Кроме того, можно задать автоматическое отображение списка свойств и методов объекта при вводе его имени и точки.

## 2.9. Создание форм

В VBA формы можно создать самостоятельно. Форма – это то же самое, что и любое диалоговое окно в операционных системах семейства Windows. Единственное различие заключается в том, что имеется возможность задать внешний вид такого окна, а также указать, для чего оно предназначено.

### 2.9.1. Добавление форм в проект

В проект можно добавить произвольное число форм. Если в документе существует несколько проектов, то имеется возможность использовать форму для них всех.

Чтобы добавить форму в проект нужно:

1. Если одновременно открыто несколько проектов, выберите имя проекта, в который требуется добавить форму, в окне проекта.
2. Выберите команду *Вставка\Форма* (Insert\User Form). Выводится пустая форма и набор инструментов.
3. В меню редактора VBA выберите команду *Вид\Окно свойств* (View\Properties Window). Выводится окно свойств.
4. Задайте свойства новой формы, например, имя.

### 2.9.2. Инструменты панели элементов

На рис. 8 показаны имена элементов управления, а в табл. 2 содержится их описание.

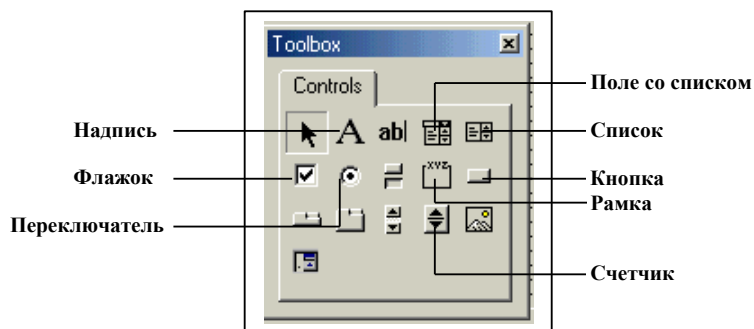


Рис. 8. Панель элементов

### Элементы управления набора инструментов

Элементы управления	Использование/Описание
Поле (Edit box)	Отображает данные и используется для ввода информации
Надпись (Label)	Отображает неизменяемый текст в формах, например, заголовки для других элементов управления
Флажок (CheckBox)	Позволяет задать значение «Истина» или «Ложь»
Поле со списком (ComboBox)	Используется либо для выбора нужного элемента из списка, либо для ввода значений в поле
Список (ListBox)	Позволяет создать список, содержащий требуемые значения
Кнопка (CommandButton)	Создает кнопку, при нажатии которой выполняется команда
Переключатель (OptionButton)	Используется для предоставления выбора одного варианта из многих
Рамка (Frame)	Позволяет установить графическую или функциональную группировку элементов управления
Счетчик (Spinner)	Используется для увеличения или уменьшения числовых значений
Рисунок (Image)	Отображает в форме импортированный рисунок
Выключатель (Toggle)	Создает кнопку, имеющую два состояния: включено и выключено

#### 2.9.3. Размещение элементов управления в форме

Разместив на панели элементов все необходимые инструменты, можно приступить к размещению элементов управления в форме. Для этого:

1. Выберите форму, в которую требуется добавить элементы управления. При необходимости выведите набор инструментов.
2. Нажмите на панели элементов кнопку требуемого элемента, а затем щелкните по любому месту формы. В форме выводится элемент управления.
3. Перетащите элемент управления в требуемую позицию.
4. Измените размеры элемента управления, перетаскив его рамку. То же самое можно сделать и с формой.

#### 2.9.4. Просмотр формы

Часто при разработке формы требуется посмотреть на ее внешний вид. По мере разработки формы можно просматривать ее. Для этого необходимо открыть форму и либо нажать клавишу <F5>, либо воспользоваться командой Запуск\Запуск подпрограммы/формы (Run\Run Sub/User Form).



## 2.9.5. Написание процедур для элементов управления в формах

Рассмотрим примеры ситуаций, в которых обязательно требуется написать код для формы:

- Запуск выполнения определенной задачи при нажатии кнопки ОК.
- Закрытие диалогового окна (и проекта VBA) при нажатии кнопки Отмена (Cancel).
- Запоминание выделенного элемента в списке.

Отмена доступа к элементу управления, например, если выбран переключатель, который запрещает изменение значения в этом элементе.

В VBA очень просто связать объект в форме с кодом. Для написания кода, необходимо открыть окно модуля, в котором задаются процедуры обработки событий этого элемента. Для выполнения данной операции:

- Дважды щелкните по элементу управления в форме. Появляется окно модуля для выбранного объекта. Выберите событие, для которого требуется создать процедуру обработки, в списке, расположенном в верхнем правом углу окна модуля. Введите текст процедуры (рис. 9). Для перехода к другому объекту выберите этот элемент управления в списке, расположенном в верхнем левом углу окна модуля.

- Щелкните по элементу управления и выберите команду Вид\Программа (View\Code) в меню редактора VBA.

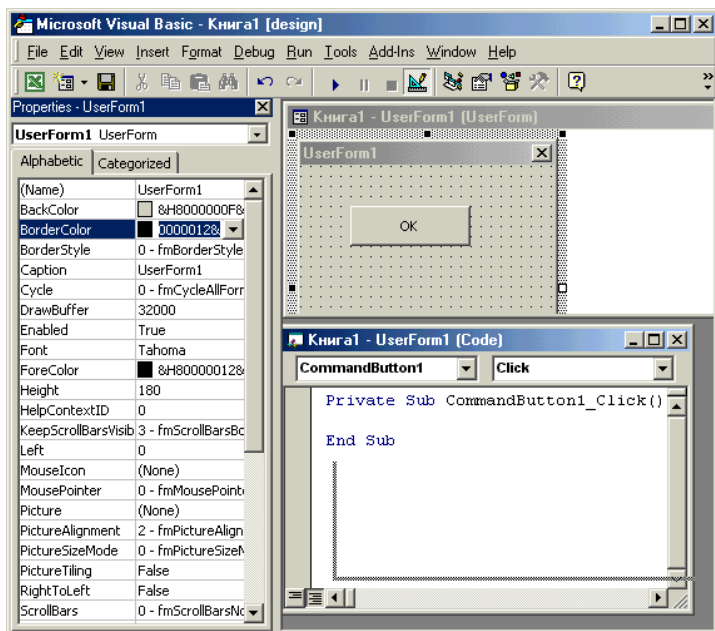


Рис. 9. Окно модуля для выбранного элемента управления

## 2.9.6. Запуск форм, процедур и макросов

До получения окончательной версии приложения потребуется неоднократно запускать формы, процедуры и макросы. Для запуска:

- *Процедуры.* Установите курсор в любом месте процедуры, а затем выберите команду *Запуск\Запуск подпрограммы/формы* или нажмите клавишу <F5>.

- *Макросы.* Для запуска макроса из окна редактора VBA выберите команду *Сервис\Макросы* (Tools\Macros), укажите имя требуемого макроса, а затем нажмите кнопку *Выполнить* (Run).

- *Формы.* Активизируйте форму или любой ее объект, а затем выберите команду *Запуск\Запуск подпрограммы/формы* или нажмите клавишу <F5>.

## 2.10. Объекты, свойства и методы VBA

Одним из основных понятий VBA является объект. Объект – это то, чем вы управляете с помощью программы на языке VBA, например, диалог, рабочий лист, диапазон ячеек. Каждый объект обладает некоторыми характеристиками или свойствами. Например, диалог может быть видимым или невидимым в данный момент на экране. Можно узнать текущее состояние диалога с помощью свойства *Visible*. Другим примером свойства объекта может быть шрифт, используемый для отображения информации в ячейке (объекте) рабочего листа. Изменяя свойства, вы можете менять характеристики объекта.

Объект содержит также список методов, которые к нему применимы. Методы – это то, что вы можете делать с объектом. Например, показать диалог на экране или убрать его с помощью методов *Show* и *Hide*, соответственно.

Таким образом, объект – это программный элемент, который имеет свое отображение на экране, содержит некоторые переменные, определяющие его свойства, и некоторые методы для управления объектом. Объект является «кирпичиком» построения программ VBA. В VBA имеется много встроенных объектов, например:

Range	Диапазон ячеек (можете включать только одну ячейку)
Cells	Ячейка
Sheet	Лист
Worksheet	Рабочий лист
DialogSheet	Диалоговое окно

Большинство объектов принадлежит к группе подобных объектов. Эти группы называются наборами. Например, все рабочие листы рабочей книги образуют набор, называемый *Worksheets*. Наборы используются одним из двух способов: либо какое-либо действие совершается

над всеми объектами набора, например, удалить, либо со ссылкой на набор выбирается конкретный объект для работы с ним. Второй способ реализован в следующей конструкции:

Worksheets ("Первый")

выбирающей рабочий лист Первый из рабочей книги. Другими примерами наборов являются:

Sheets	Листы
DialogSheets	Диалоговые окна
DrawingObjects	Графические объекты

Изменяя свойства, можно изменять характеристики объекта или набора объектов. Установка значений свойств – это один из способов управления объектами. Для установки свойства необходимо ввести имя объекта, затем поставить точку и за ней – имя свойства. Далее должен следовать знак равенства и значение свойства. Синтаксис установки значения свойства объекта выглядит следующим образом:

Объект.Свойство = Выражение

В приведенном ниже примере для свойства Value диапазона ячеек Начальные\_данные устанавливается значение 0,1 (т.е. в ячейках этого диапазона будет записано число 0,1):

Range("начальные\_данные").Value = 0,1

В следующем примере в ячейку A2 вставляется формула путем изменения свойства Formula (формула):

Range("A2").Formula = "=СУММ(A1:C1)"

Некоторые свойства являются неизменяемыми, т.е. допустимыми только для чтения. Имеется в виду, что значение свойства можно узнать, но нельзя изменить. Например, для диапазона, состоящего из одной ячейки, свойства Row (строка) и Column (столбец) являются неизменяемыми. Другими свойствами, можно узнать, в какой строке и в каком столбце находится ячейка, но изменить ее положение путем изменения ее свойств нельзя. Синтаксис чтения свойств объекта выглядит следующим образом:

Переменная = Объект.Свойство

В следующем примере переменной Процентная\_ставка присваивается значение из ячейки A1 текущего рабочего листа:

Процентная\_ставка = Range("A1").Value

или другим способом:

Процентная\_ставка = Cells(1,1).Value

Кроме свойств, как уже отмечалось выше, у объектов есть ряд методов, т.е. команд, применяемых к объекту. Например, У объекта – диапазон ячеек – имеется метод `Clear`, позволяющий очистить содержимое диапазона. Приводимый ниже пример показывает, как можно очистить диапазон Начальные\_данные :

```
Range("Начальные_данные").Clear
```

А в примере

```
Range("Начальные_данные").Clear
```

```
Range("A10:B12").Select
```

выбирается диапазон ячеек A10:B12. Таким образом, синтаксис вызова объекта имеет следующий вид:

Объект.Метод

В Excel имеется много объектов, причем некоторые из них содержат другие объекты. Например, рабочая книга содержит рабочие листы, рабочий лист содержит диапазон ячеек и т.д. Объектом самого высокого уровня является `Application` (приложение). Если вы изменяете его свойства или вызываете его методы, то результат применяется к текущей работе Excel. Например, можно завершить работу с Excel, применив метод `Quit` (выход) к объекту `Application`:

```
Application.Quit
```

Как было отмечено, точка после имени объекта указывает на то, что далее следует имя свойства или метода. Но после точки можно указать и имя объекта для перехода от одного объекта к другому. Например, следующее выражение очищает вторую строку рабочего листа Май в рабочей книге Отчет :

```
Application.WorkBooks("Отчет").Worksheets("Май").Row(2).Delete
```

Строка может содержать отдельные ячейки, свойства которых тоже можно устанавливать:

```
Application.WorkBooks("Отчет").Sheets("Май").Row(2)._  
Cells(1). Value="Да"
```

## **2.10.1. Элементы управления в приложениях VBA**

### **2.10.1.1. Добавление элемента управления в форму**

Чтобы добавить элемент управления в форму, нажмите на панели элементов соответствующий элемент, например Поле (`TextBox`) (рис. 10), а затем щелкните по форме. Элемент появится на форме (рис. 11). После создания элемента управления лучше сразу присвоить ему новое имя.

Если не указать название элемента, то используется имя, заданное по умолчанию, а при последующем изменении имени элемента управления, потребуется отредактировать все процедуры, в которых имеется старое имя объекта (значение по умолчанию).

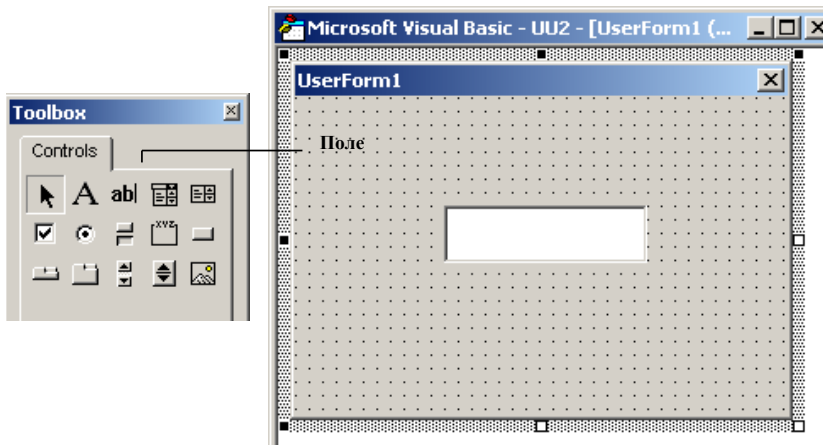


Рис. 10. Кнопка Поле

Рис. 11. Поле в форме

### 2.10.1.2. Использование надписей

Элемент управления Label (Надпись) используется для вывода текста в форме. Например, заголовка, не имеющих собственного свойства Caption объектов формы, таких как поле или рисунок. В этом случае надпись находится около элемента управления, указывая на назначение элемента. В других случаях объект Label применяется для отображения другой информации.

Наиболее важным свойством элемента управления Label является свойство Caption, содержащее текст надписи. Свойство Caption можно задать в окне свойств, так же как и имя элемента управления Label. Подробную информацию о работе с окном свойств смотрите в главе 3. Покажем, как свойство Caption используется в программе:

```
MyLabelControl.Caption = "Адрес"
```

### 2.10.1.3. Использование кнопок

Элемент управления CommandButton (Кнопка) инициирует выполнение некоторого действия, например, запуск, остановку или прерывание процесса.

Если не указать название кнопки, то используется имя, заданное по умолчанию, а при последующем изменении имени элемента управле-

ния, потребуется отредактировать все процедуры, в которых имеется старое имя объекта (значение по умолчанию).

Чтобы указать имя объекта `CommandButton`, нажмите клавишу `<F4>` сразу после создания кнопки, а затем введите название объекта в поле Имя окна свойств. При необходимости установите значения других свойств.

Внешний вид элемента управления `CommandButton` можно изменить. Например, имеется возможность задать текст, который выводится на кнопке, с помощью свойства `Caption`. Рассмотрим фрагмент программы, в котором текст кнопки меняется на строку "Нажми меня!":

```
CbMyControl.Caption = "Нажми меня!"
```

Если весь текст не уместится на поверхности кнопки, то он будет усекаться. Однако можно задать автоматическое изменение размеров объекта `CommandButton` с помощью свойства `AutoSize`:

```
CbMyControl.AutoSize = True
```

С нажатием кнопки можно связать выполнение некоторого действия, если назначить ее событию Нажатие кнопки (`Click`) процедуру обработки. Процедура обработки события Нажатие кнопки не имеет параметров. В программе 2 фраза "кнопка нажата" выводится в окне отладки, когда нажимается кнопка `CommandButton1`.

```
Private Sub cbMyButton_Click ()  
    Debug.Print "кнопка нажата"  
End Sub
```

#### 2.10.1.4. Использование списков

Элемент управления `ListBox` (Список) применяется для хранения списка значений. В списке пользователь выбирает один или несколько элементов, которые затем используются в программе.

В VBA имеется возможность задать число элементов в списке, которые можно выбрать одновременно. Если допускается выделение нескольких пунктов, то имеется возможность указать также, что выбранные элементы списка расположены либо последовательно, либо их порядок является произвольным.

Способ выбора элементов определяется значением свойства `MultiSelect`. Его можно задать как в окне свойств, так и в программе. В табл. 3 показаны значения, которые соответствуют описанным вариантам выделения.

Приведем пример задания варианта выбора:

```
ListBox1.MultiSelect = fmMultiSelectExtended
```

**Значения вариантов выделения**

Вариант	Значение	Константа
Один элемент	0	FmMultiSelectSingle
Последовательные элементы	2	FmMultiSelectExtended
Произвольно расположенные элементы	1	RnMultiSelectMulti

Для добавления элементов списка используется метод `AddItem`. Для данного метода требуется задать параметр, который определяет строку с названием пункта:

```
ListBox.AddItem строка_элемента
```

В приведенной ниже программе показано, как добавить в список дни недели.

```
Public Sub PopulateList()
    ListBox1.AddItem "Понедельник"
    ListBox1.AddItem "Вторник"
    ListBox1.AddItem "Среда"
    ListBox1.AddItem "Четверг"
    ListBox1.AddItem "Пятница"
    ListBox1.AddItem "Суббота"
    ListBox1.AddItem "Воскресенье"
End Sub
```

Чтобы заполнить список числами, обычно используется цикл `For...Next`.

```
Public Sub PopulateList()
    For I = 1 to 100
        ListBox1.AddItem "Номер элемента: " + str$(I)
    Next
End Sub
```

Если в программе потребуется определить выбранные элементы списка, то необходимо использовать свойство `Text`, которое содержит выделенный в объекте пункт, если в списке задан выбор только одного элемента. Если же в элементе `ListBox` допускается выбор нескольких элементов, то свойство `Text` равно пустой строке. Покажем, как вывести выбранный элемент списка в окне отладки:

```
Debug.Print ListBox1.Text
```

Кроме того, для определения выбранного в списке элемента используется свойство `ListIndex`. Оно содержит номер выделенного пункта.

### 2.10.1.5. Использование полей со списком

Объект `ComboBox` (Поле со списком) позволяет выбрать элемент из списка, либо ввести данные вручную. Текущее значение в поле со списком отображается в поле, а список возможных значений выводится при нажатии кнопки со стрелкой. Отличие объекта `ComboBox` от элемента управления `ListBox` заключается в том, что в поле со списком можно явно выделить требуемое значение. Кроме того, объект `ComboBox` занимает в форме меньше места.

Для заполнения элемента управления `ComboBox` используется метод `AddItem`, как и для элемента управления `ListBox`. Для данного метода требуется задать параметр, который определяет требуемый пункт:

```
ComboBox.AddItem строка_элемента
```

Чтобы получить или установить значение, хранящееся в текстовой части поля со списком, используется свойство `Text` объекта `ComboBox`. При присваивании значения данному свойству автоматически выполняется следующее:

- Заданный текст выводится в поле. Однако если заданное значение не является элементом списка, то отображается сообщение об ошибке.
- Свойству `ListIndex` объекта `ComboBox` присваивается индекс элемента списка, соответствующего заданному значению.

Программа, приведенная ниже выводит слово "красный" в окне отладки.

```
Private Sub UserForm_Click()  
    ComboBox1.AddItem "зеленый"  
    ComboBox1.AddItem "синий"  
    ComboBox1.AddItem "красный"  
    ComboBox1.AddItem "белый"  
    ComboBox1.Text = "красный"  
    I = ComboBox1.ListIndex  
    Debug.Print ComboBox1.List (I)  
End Sub
```

Чтобы получить значение, содержащееся в поле со списком, используются свойства `Text` или `Value`.

Элементы списка могут формироваться из данных электронной таблицы. В свою очередь выбранные данные могут помещаться в любую ячейку электронной таблицы. В программе показано как заполнить список днями недели, расположенными в ячейках A1:A7 и вывести результат в ячейку B1, нажав соответствующую кнопку на форме.

```
Private Sub UserForm_Initialize()  
    With ComboBox1
```



```

For i = 1 To 7
.AddItem (Cells(i, 1))
Next
End With
End Sub
Private Sub CommandButton1_Click()
Range("b1").Value = ComboBox1.Value
End Sub

```

### 2.10.1.6. Использование флажков

Элемент управления CheckBox (Флажок) создает ячейку, которая может быть помечена пользователем, как имеющая значение *истина* или *ложь*. Флажок представляет собой маленький квадрат, с которым связан заголовок. Если квадрат пуст, то при щелчке по нему появляется галочка, и наоборот, если квадрат помечен галочкой, то при щелчке по нему галочка исчезает. Кроме того, элемент управления CheckBox может иметь три состояния. Это означает, что пользователь имеет возможность установить неопределенное значение для объекта CheckBox: *не ложь* и не *истина*.

Иногда необходимо определить текущее состояние флажка. Для этого имеется свойство Value, значение которого показывает, установлен ли флажок или нет. Если да, то свойство Value равно True. Состояние флажка обычно используется в процедуре обработки события флажка Нажатие кнопки, либо при принятии решения о выполнении некоторого действия.

### 2.10.1.7. Использование переключателей

Элемент управления OptionButton (Переключатель) используется для предоставления выбора одного варианта из многих (рис. 22). На заданный момент в группе может быть выбран один и только один переключатель. Отмена выбора одного объекта OptionButton при выделении другого осуществляется автоматически, поэтому в программе это не требуется специально предусматривать. Объекты OptionButton иногда группируются в рамке.

Иногда требуется определить выбранный объект OptionButton. Свойству Value выбранного пользователем переключателя автоматически присваивается значение True. Поскольку известно, какие переключатели входят в группу, необходимо просто проверить, свойство Value какого из них имеет значение True, как показано в программе.

```

Public Sub CheckOptionButtons()
If OptionButton1.Value Then
'инструкции
Elseif OptionButton2.Value Then
'инструкции
End If
End Sub

```

## 3. ПРИМЕР СОЗДАНИЯ ПРИЛОЖЕНИЯ

### 3.1. Постановка задачи

Ежедневно в несколько булочных с хлебозавода поступают различные сорта хлеба по разной цене. Поставка может осуществляться несколько раз в день. По мере поступления хлеба оператор вводит в таблицы данные о количестве хлеба, поставленного в каждую булочную. Производится расчет стоимости хлеба по сортам и общей стоимости хлеба. В конце работы формируются итоговые таблицы. Результаты расчета распечатываются.

Разрабатываемая система должна предоставлять возможность пользователю ввода, обновления информации, расчета итогов, формирования выходных документов и просмотра инструкций.

### 3.2. Алгоритм решения задачи

Разрабатываемая система должна позволять выполнять следующие функции:

- ввод информации;
- корректировка ранее введенных данных, включая исходные данные и созданные документы по булочным;
- формирование итоговых документов, а именно отчет по работе хлебокомбината за день, отчет по поставке хлеба по сортам и итоговый отчет по поставке хлеба в булочные;
- поиск информации;
- просмотр инструкций;
- возможность обновления всей системы.

Для выполнения этих функций в системе должно быть разработано меню в виде набора кнопок на экранной форме. Работа системы предполагает предварительное создание таблиц-шаблонов, каждый из которых хранится в отдельном файле. Базовой из которых является таблица Shablon, хранящаяся в файле Shablon.xls (рис. 12).

Кроме описанной выше таблицы в системе используются следующие таблицы-шаблоны (табл. 4, табл. 5, табл. 6):

Таблица 4

#### Шаблон «Поставка хлеба по булочной» (файл Shablon.xls)

Сорт хлеба	Выпечка в штуках	Цена (в руб.)	Стоимость (в руб.)
...	...	...	...

Таблица 5

#### Шаблон «Сводная информация по булочным» (файл itBul.xls)

№ булочной	Выручка (в руб.)
...	...

Работа хлебокомбината за день				
Всего видов хлеба				
Изменить выпечку	Вернуться в главное меню	Поиск информации	Удалить данные по текущему документу	Печать
булочная №				
Дата				
Сорт хлеба	Выпечка в штуках	Цена (в руб.)	Стоимость (в руб.)	

Рис. 12. Таблица Shablon

Таблица 6

**Шаблон «Сводная информация о поставках хлеба по сортам в булочные» (файл itSort.xls)**

№ булочной	Количество хлеба (в шт.)	Выручка (в руб.)
...		...

Перед созданием программы необходимо разработать интерфейс системы, в основу которого положена структура дерева диалога, представленная на рис. 13.

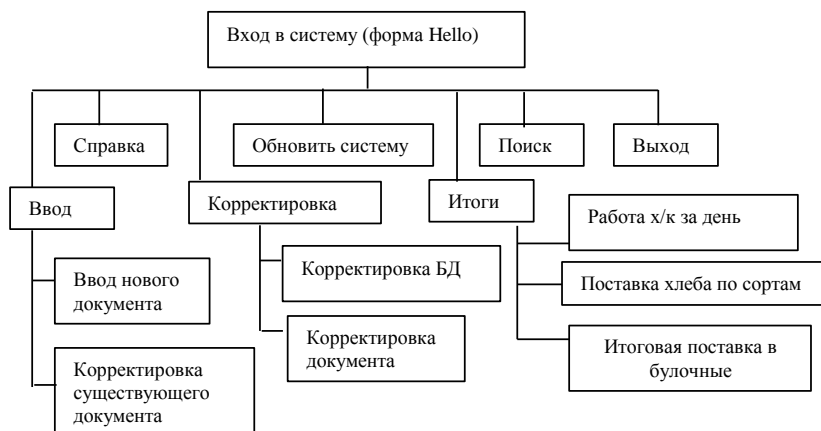


Рис. 13. Дерево диалога

### 3.3. Описание программных модулей системы

#### 3.3.1. Структура основного макроса

Основной макрос представляет собой модуль, состоящий из отдельных процедур – макросов, которые и позволяют управлять системой.

Модуль начинается с описания переменных, необходимых для работы программы. Присвоения типа переменным Global необходимо для того, чтобы эти переменные были доступны на уровне всего проекта

```
Public Const MaxBul = 30 ' кол-во булочных
Public Const Bread = 100 ' кол-во сортов хлеба
Public NumBread As Integer ' номер булочной
Global DifSort(Bread) As String ' виды хлеба
Global PaySort(Bread) As Currency ' цена хлеба
Global KolAll(Bread) As Long
Public pay As String
Global HowMuch As Integer ' сколько всего видов хлеба
Public VidBread As Integer
Public myNameBread As String
Public Payment As String
Global SortChange As Integer
Global FlagShow As Boolean
Public MyX As Boolean
Public FlagDobavBD As Boolean
Global MyPath As String ' пароль для доступа к рабочим файлам
Dim MyR As Byte
Public FlagSorts As Boolean
```

Для автоматического запуска системы при открытии файла в основном макросе находится макрос Auto\_Open(), который проверяет наличие основного файла системы (MainBread.xls) в директории и открывает его. Путь к нему может указать сам пользователь, по умолчанию путь MyPath = "c:\Учет поставок хлеба\". Если система не находит этот файл, то выдается сообщение об ошибке и работа прекращается. Текст макроса приведен ниже.

```
Sub Auto_Open()
    msg = "Укажите путь для доступа к файлам системы." & Chr(13) &
    Chr(10) & " ВНИМАНИЕ! При неверном указании пути система рабо-
    тать не будет!"
    q = InputBox(msg, "Breads", "c:\Учет поставок хлеба\")
    If (Dir(q, vbDirectory) = "") Or (q = "") Then
        a = MsgBox("Неверно указан путь для доступа.", vbCritical,
        "Breads")
    Else
```

```

MyPath = Trim(q) ' удаляю пробелы
'MyPath = " c:\Учет поставок хлеба \"
If Dir(MyPath & "MainBread.xls") <> "" Then
Application.ScreenUpdating = False
OldDisplay = Application.DisplayFullScreen
Application.DisplayFullScreen = True
Toolbars(13).Visible = False
Workbooks.Open MyPath & "MainBread.xls"
Worksheets("Лист1").Activate
ActiveWindow.ScrollRow = 1
ActiveSheet.Protect Password:="I&R", DrawingObjects:=True, Con-
tents:=True, Scenarios:=True
ActiveWorkbook.Protect Password:="I&R", Structure:=True, Win-
dows:=True
ActiveWorkbook.Save
Call activBD ' активизирую базу данных
Hello.Show ' показываю заставку – приглашение
MyX = False
Else
a = MsgBox("Не найден файл MainBread.xls. Работа прекращается.",
vbCritical, "Breads")
End If
End If
End Sub

```

Загрузка меню происходит из файла MainBread.xls. Перед непосредственной загрузкой меню выдается сообщение о начале работы и на экране появляется форма «Hello» (рис. 14).

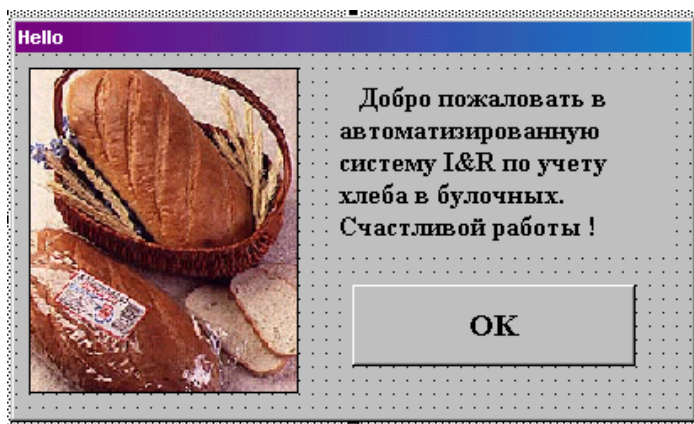


Рис. 14. Заставка программы

```
Private Sub CommandButton1_Click()  
Hello.Hide  
End Sub
```

Далее пользователь имеет возможность выбрать вид осуществляемой работы, нажимая на ту или иную кнопку на главном меню (рис. 15).



Рис. 15. Основное меню системы

### 3.3.2. Структура макроса для первичного ввода данных

Первичный ввод информации осуществляется при нажатии кнопки меню «Ввод данных» и выбора номера булочной в окне диалога «Номер булочной» (форма NameBread) (рис. 16). Текст макроса приводится ниже.

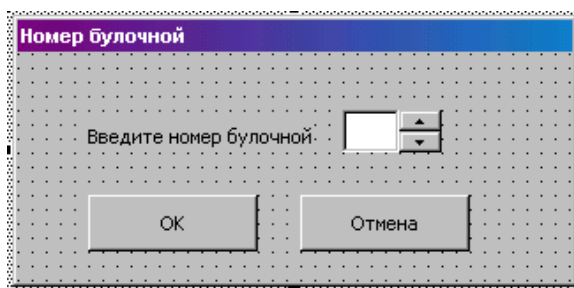


Рис. 16. Формой «Номер булочной»

```

Public Sub makenew()
FlagShow = False
NameBread.Show
If FlagShow = True Then
If Dir(MyPath & "Bread" & Str(NumBread) & ".xls") <> "" Then
msg = "Данные по булочной с номером " & Str(NumBread) & " уже
существуют. Заменить?"
a = MsgBox(msg, vbInformation + vbDefaultButton2 + vbYesNoCancel, "Breads")
Select Case a
Case vbYes
Call new2 'Вызов подпрограммы создания нового документа
Case vbNo
MyX = True
Call makeold
End Select
Else
Call new2
End If
End If
End Sub

```

При работе с формой «Номер булочной» необходимо описать процедуру для каждой кнопки формы и счетчика. Ниже приводится описание процедуры. Логическая переменная FlagShow вводится для определения наличия файла с нужным номером булочной и принимает значения False или True.

**Private Sub CommandButton1\_Click()**

```

Unload NameBread
FlagShow = False
End Sub

```

**Private Sub much\_SpinDown()**

```

Num.Value = Much.Value
NumBread = Val(Much.Value)
End Sub

```

**Private Sub much\_SpinUp()**

```

Num.Value = Much.Value
NumBread = Val(Much.Value)
End Sub

```

**Private Sub OKButton\_Click()**

```

Unload NameBread
FlagShow = True

```

End Sub

**Private Sub UserForm\_Initialize()**

Much.max = MaxBul

Num.Value = 1

Much.Value = 1

NumBread = Val(Num.Value)

End Sub

Если система находит данные по булочной с выбранным номером, то выдается сообщение и пользователь может выбрать, либо заменить имеющиеся данные, либо продолжить работу с ними. В случае, если булочная с выбранным номером не обнаружена создается новый документ. Для этого используется подпрограмма new2().

**Sub new2()**

Application.ScreenUpdating = False

ActiveWorkbook.Close savechanges:=False

If Dir(MyPath & "Shablon.xls") <> "" Then

Workbooks.Open MyPath & "Shablon.xls"

Sheets("Лист1").Select

Application.DisplayFullScreen = True

ActiveWorkbook.Unprotect Password:="I&R"

ActiveSheet.Unprotect Password:="I&R"

ActiveSheet.Cells(10, 4).Value = Str(NumBread)

For i = 1 To HowMuch

ActiveSheet.Cells(i + 14, 3).Value = DifSort(i)

ActiveSheet.Cells(i + 14, 5).Value = Str(PaySort(i))

Call MyLines(i + 14, 3, 6)

Next i

ActiveSheet.Cells(HowMuch + 15, 6).Value = ActiveSheet.Cells(51,

6).Value

For i = HowMuch + 16 To 51

ActiveSheet.Cells(i, 6).Value = ""

Next i

‘Занесение номера булочной в документ

ActiveSheet.Cells(4, 5).Value = Str(HowMuch)

ActiveSheet.Cells(HowMuch + 15, 3).Value = "Итого стоимость"

‘Занесение даты документ

Range("D11").Value = Date

Sheets("Лист1").Protect Password:="I&R", DrawingObjects:=True,

Contents:=True, Scenarios:=True

ActiveSheet.Protect Password:="I&R", Structure:=True, Windows:=True

‘Задание имени файла



```

fname = MyPath & "Bread" & Str(NumBread) & ".xls"
If Dir(fname) <> "" Then Kill fname
Workbooks("Shablon").SaveAs FileName:=fname
ActiveWindow.ScrollRow = 1
Else
a = MsgBox("Не найден файл для формирования отчета!", vbCritical, "Breads.")
Workbooks.Open MyPath & "MainBread.xls"
Worksheets("Лист1").Activate
ActiveWindow.ScrollRow = 1
End If
End Sub

```

При сохранении нового документа ему присваивается имя «Bread n.xls», где n – номер булочки. Документ защищен от изменений с помощью пароля.

Процедура MyLines – вспомогательная, служит для вычерчивания рамки вокруг вновь появившихся данных.

```

Public Sub MyLines(r1, c1, c2 As Integer)
ActiveSheet.Range(Cells(r1, c1), Cells(r1, c2)).Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
.LineStyle = xlContinuous
.Weight = xlThin
.ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeTop)
.LineStyle = xlContinuous
.Weight = xlThin
.ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeBottom)
.LineStyle = xlContinuous
.Weight = xlThin
.ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeRight)
.LineStyle = xlContinuous
.Weight = xlThin
.ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideVertical)

```

```

.LineStyle = xlContinuous
.Weight = xlThin
.ColorIndex = xlAutomatic
End With
End Sub

```

### 3.3.3. Корректировка документа

При нажатии на кнопку «Корректировка» основного меню на экране появляется форма, позволяющая пользователю выбрать, какой именно документ надо корректировать: исходные данные (база данных) или документ по булочной (рис. 17).

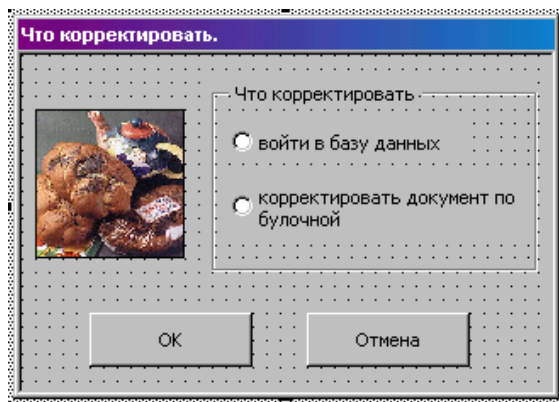


Рис. 17. Форма для выбора варианта для корректировки

Описание процедур для элементов формы «Что корректировать».

**Private Sub CommandButton1\_Click()**

```
If OptionButton1.Value = True Then
```

```
Unload ChCor
```

```
Call showBD
```

```
Else
```

```
Unload ChCor
```

```
Call makeold
```

```
End If
```

```
End Sub
```

**Private Sub CommandButton2\_Click()**

```
Unload ChCor
```

```
End Sub
```

**Private Sub UserForm\_Activate()**

```
OptionButton1.Value = True
```

```
End Sub
```

### **Private Sub UserForm\_Initialize()**

```
OptionButton1.Value = True  
End Sub
```

Процедура корректировки документа по булочной повторяет процедуру ввода нового документа, с той разницей, что загружается уже существующий файл. Номер булочной, данные по которой надо загрузить, получены с помощью уже упоминавшейся формы NameBread.

### **Public Sub makeold()**

```
If MyX = False Then  
    FlagShow = False  
    NameBread.Show  
End If  
If FlagShow = True Then  
    MyX = False  
    'Проверка на присутствие файла в директории  
    If Dir(MyPath & "Bread" & Str(NumBread) & ".xls") = "" Then  
        msg = "Извините, но данных по булочной с номером " &  
Str(NumBread) & " нет."  
        answer = MsgBox(msg, vbCritical + vbDefaultButton1 + vbOKOnly, "Breads")  
    Else  
        Application.ScreenUpdating = False  
        ActiveWorkbook.Close savechanges:=False  
        Workbooks.Open MyPath & "Bread" & Str(NumBread) & ".xls"  
        Sheets("Лист1").Select  
        ActiveWindow.ScrollRow = 1  
        ActiveSheet.Protect Password:="I&R", DrawingObjects:=True,  
Contents:=True, Scenarios:=True  
        ActiveWorkbook.Protect Password:="I&R", Structure:=True, Windows:=True  
    End If  
End If  
End Sub
```

При выборе кнопки «войти в базу данных» загружается база данных. Если данные по этой булочной не обнаружены, то выдается сообщение об ошибке и вновь высвечивается главное меню.

### **3.3.4. Формирование итоговых документов**

При нажатии на кнопку «Итоги» основного меню на экране появляется окно «Итоги.Breds» (форма choice), вызов которой осуществляется процедурой, показанной ниже.

### **Public Sub ItogAll()**

```
    chose.Show  
End Sub
```

Форма «Итоги.Breds» позволяет пользователю выбрать, какой именно документ надо сформировать (рис. 18).

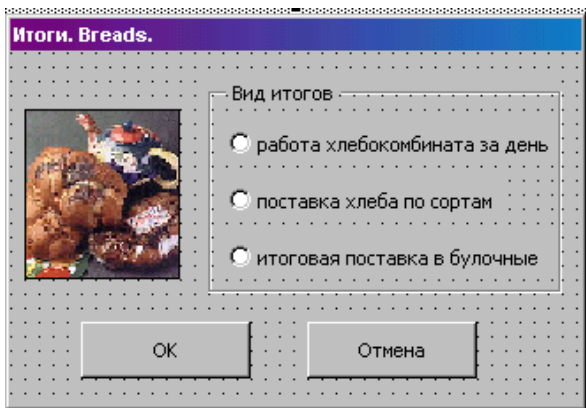


Рис. 18. Форма выбора вида итогового документа

Процедура выбора формы отчета (кнопка ОК) приведена ниже.

### **Private Sub CommandButton1\_Click()**

```
If OptionButton1.Value = True Then  
    Unload chose  
    Call dairybak  
ElseIf OptionButton2.Value = True Then  
    Unload chose  
    Call mysorts  
ElseIf OptionButton3.Value = True Then  
    Unload chose  
    Call bakery  
End If  
End Sub
```

### **Private Sub CommandButton2\_Click()**

```
    Unload chose  
End Sub
```

#### **3.3.4.1. Формирование документа «Работа хлебокомбината за день»**

Во всех таблицах содержатся данные о количестве поставленного в булочные хлеба, его цене, стоимости и итоговой стоимости полученного булочной хлеба. Требуется рассчитать суммарную стоимость хлеба и

представить результаты расчета в виде таблицы Shablon1. Таблица находится в файле Shablon1.xls (рис. 19) и, в целом, совпадает с таблицей Shablon.

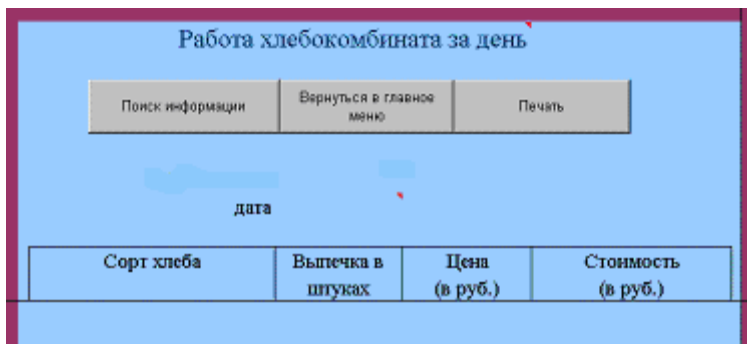


Рис. 19. Таблица Shablon1

Для формирования документа необходимо выполнить следующие действия.

Загружаем в новое окно таблицу Shablon1.

Формируем новый документ путем суммирования данных в столбце «Выпечка в штуках». Эта процедура организована в цикле. Происходит обращение к каждой из существующих таблиц для суммирования их по колонке «Выпечка в штуках». Имя загружаемого файла формируется в цикле с помощью операции соединения (&).

Сохраняем полученную таблицу с именем DairyWork.xls. Если такой файл уже существовал, то он удаляется с диска и записывается вновь. Текст макроса имеет следующий вид.

```
Public Sub dairybak()
    If Dir(MyPath & "Shablon1.xls") <> "" Then
        Application.ScreenUpdating = False
        ActiveWorkbook.Close savechanges:=False
        For i = 1 To Bread
            KolAll(i) = 0
        Next i 'обнуляем общее кол-во
        For i = 1 To MaxBul 'цикл по всем булочным
            'Проверка на присутствие файла в директории
            If Dir(MyPath & "Bread" & Str(i) & ".xls") <> "" Then Work-
                books.Open MyPath & "Bread" & Str(i) & ".xls"
                Sheets("Лист1").Select
                For j = 1 To HowMuch 'цикл по всем видам хлеба
```

```

    KolAll(j) = KolAll(j) + Val(ActiveSheet.Cells(j + 14, 4).Value)
Next j
Workbooks("Bread" & Str(i) & ".xls").Close
    End If
Next i
Workbooks.Open MyPath & "Shablon1.xls"
Sheets("Лист1").Select
ActiveWorkbook.Unprotect Password:="I&R"
ActiveSheet.Unprotect Password:="I&R"
Range("D11").Value = Date
it = 0
For i = 1 To HowMuch
    ActiveSheet.Cells(i + 14, 3).Value = DifSort(i)
    ActiveSheet.Cells(i + 14, 4).Value = Str(KolAll(i))
    ActiveSheet.Cells(i + 14, 5).Value = Str(PaySort(i))
    ActiveSheet.Cells(i + 14, 6).Value = Str(PaySort(i) * KolAll(i))
    it = it + PaySort(i) * KolAll(i)
    Call MyLines(i + 14, 3, 6)
Next i
ActiveSheet.Cells(10, 4).Value = "Все булочные"
ActiveSheet.Cells(HowMuch + 15, 6).Value = it
ActiveSheet.Cells(HowMuch + 15, 3).Value = "Итого стоимость"
fname = MyPath & "DairyWork.xls"
Range("C15").Select
If Dir(fname) <> "" Then Kill fname
Workbooks("Shablon1").SaveAs FileName:=fname
ActiveWindow.ScrollRow = 1
Sheets("Лист1").Protect Password:="I&R", DrawingObjects:=True,
Contents:=True, Scenarios:=True
Workbooks("DairyWork.xls").Protect Password:="I&R", Structure:=True,
Windows:=True
Else
    a = MsgBox("Не найден файл для формирования отчета!",
vbCritical, "Breads.")
End If
End Sub

```

### **3.3.4.2. Формирование документа «Итоговая поставка в булочные»**

Для формирования документа необходимо рассчитать суммарную стоимость хлеба, и сформировать сводную таблицу (рис. 20).

Для формирования документа необходимо выполнить следующие действия:

Загружаем в новое окно шаблон сводной таблицы, который хранится в файле ItBul.xls и имеет следующий вид.

Вернуться в главное меню		Печать	
<b>Итоговая поставка в булочные</b>			
<b>Всего известных</b>			
<b>булочных</b>			
<b>Дата</b>			
<b>Номер булочной</b>		<b>Выручка (в руб.)</b>	

Рис. 20. Сводная таблица ItBul.xls

Формируем новый документ путем копирования итоговых сумм из существующих таблиц в таблицу – шаблон. Эта процедура организована в виде цикла.

Сформированная таблица сохраняется с именем DairyBul.xls. Если такая таблица уже существовала ранее, то она удаляется и записывается вновь. Макрос для выполнения этих действий имеет следующий вид.

**Public Sub bakery()**

```

If Dir(MyPath & "ItBul.xls") <> "" Then
Application.ScreenUpdating = False
ActiveWorkbook.Close savechanges:=False
Workbooks.Open MyPath & "ItBul.xls"
Sheets("Лист1").Select
ActiveWorkbook.Unprotect Password:="I&R"
ActiveSheet.Unprotect Password:="I&R"
kolb = 0
it = 0
For i = 1 To MaxBul 'цикл по всем булочным
'Проверка на присутствие файла в директории
If Dir(MyPath & "Bread" & Str(i) & ".xls") <> "" Then
kolb = kolb + 1
Workbooks.Open MyPath & "Bread" & Str(i) & ".xls"
Sheets("Лист1").Select
pay = ActiveSheet.Cells(15 + HowMuch, 6).Value
Workbooks("ItBul.xls").Activate
Sheets("Лист1").Select
ActiveSheet.Cells(kolb + 8, 2).Value = Str(i)
ActiveSheet.Cells(kolb + 8, 3).Value = pay
it = it + CCur(pay)
Call MyLines(kolb + 8, 2, 3)
Workbooks("Bread" & Str(i) & ".xls").Close
End If

```

```

Next i
Workbooks("ItBul.xls").Activate
Sheets("Лист1").Select
ActiveSheet.Cells(kolb + 9, 3).Value = it
ActiveSheet.Cells(6, 3).Value = Str(kolb)
Range("C7").Value = Date
ActiveSheet.Cells(kolb + 9, 2).Value = "Итого стоимость"
Range("B7").Select
fname = MyPath & "DairyBul.xls"
If Dir(fname) <> "" Then Kill fname
  Workbooks("ItBul.xls").SaveAs FileName:=fname
  ActiveWindow.ScrollRow = 1
  Sheets("Лист1").Protect Password:"I&R", DrawingObjects:=True,
Contents:=True, Scenarios:=True
  ActiveWorkbook.Protect Password:"I&R", Structure:=True, Win-
dows:=True
  Else
    a = MsgBox("Не найден файл ItBul.xls для формирования отчета!",
vbCritical, "Breads.")
  End If
End Sub

```

### 3.3.4.3. Формирование документа «Поставка хлеба по сортам»

Для формирования документа необходимо рассчитать суммарное количество и суммарную стоимость выбранного пользователем сорта хлеба и сформировать сводную выходную таблицу.

Для формирования документа необходимо выполнить следующие действия:

Загружаем форму, с помощью которой пользователь выбирает нужный вид хлеба (рис. 21).

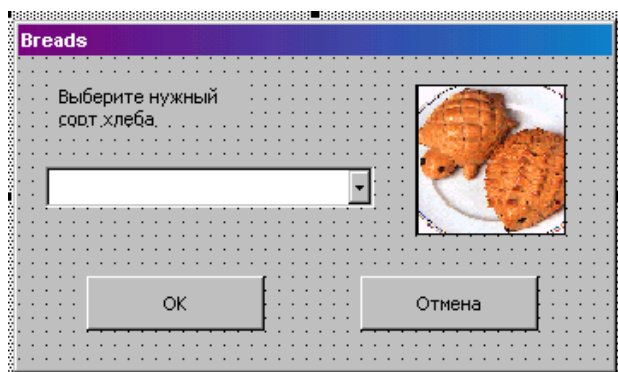


Рис. 21. Форма для выбора сорта хлеба



**Private Sub CommandButton1\_Click()**

```
VidBread = dsort.ListIndex + 1
```

```
Unload sorts
```

```
End Sub
```

**Private Sub CommandButton2\_Click()**

```
FlagSorts = False
```

```
Unload sorts
```

```
End Sub
```

**Sub BegInAct()**

```
dsort.Clear
```

```
For i = 1 To HowMuch
```

```
    dsort.AddItem DifSort(i)
```

```
Next i
```

```
dsort.ListIndex = 0
```

```
End Sub
```

**Private Sub UserForm\_Initialize()**

```
Call BegInAct
```

```
End Sub
```

Загружаем в новое окно файл ItSort.xls (если он существует) со сводной таблицей (рис. 22).

Поставка в булочные сорта хлеба				
Сорт хлеба	Дата	Номер булочной	Кол-во хлеба (в шт.)	Выручка за этот вид (в руб.)

Рис. 22. Сводная таблица ItSort.xls

Формируем новый документ путем обращения к строке с номером `howmuch+14`, где `howmuch` – номер сорта хлеба в массиве, загружаемом из базы данных.

Сохраняем сформированную таблицу с именем `DairySort.xls`. Если такой файл уже существовал, то он удаляется и сохраняется вновь. Макрос имеет следующий вид.

**Public Sub mysorts()**

```
If Dir(MyPath & "ItSort.xls") <> "" Then
```

```
FlagSorts = True
```

```
sorts.Show
```

```

Application.ScreenUpdating = False
If FlagSorts = True Then
ActiveWorkbook.Close savechanges:=False
Workbooks.Open MyPath & "ItSort.xls"
Sheets("Лист1").Select
ActiveWorkbook.Unprotect Password:="I&R"
ActiveSheet.Unprotect Password:="I&R"
Range("C7") = DifSort(VidBread)
Range("C8") = Date
kolb = 0
itp = 0
itk = 0
For i = 1 To MaxBul 'цикл по всем булочным
'Проверка на присутствие файла в директории
If Dir(MyPath & "Bread" & Str(i) & ".xls") <> "" Then
kolb = kolb + 1
Workbooks.Open MyPath & "Bread" & Str(i) & ".xls"
Sheets("Лист1").Select
pay = ActiveSheet.Cells(VidBread + 14, 6).Value
Kol = ActiveSheet.Cells(VidBread + 14, 4).Value
Workbooks("ItSort.xls").Activate
Sheets("Лист1").Select
ActiveSheet.Cells(kolb + 9, 2).Value = Str(i)
ActiveSheet.Cells(kolb + 9, 3).Value = Kol
ActiveSheet.Cells(kolb + 9, 4).Value = pay
itp = itp + CCur(pay)
itk = itk + Val(Kol)
Call MyLines(kolb + 9, 2, 4)
Workbooks("Bread" & Str(i) & ".xls").Close
End If
Next i
Workbooks("ItSort.xls").Activate
Sheets("Лист1").Select
ActiveSheet.Cells(kolb + 10, 3).Value = Str(itk)
ActiveSheet.Cells(kolb + 10, 4).Value = Str(itp)
ActiveSheet.Cells(kolb + 10, 2).Value = "Итого"
Range("C7").Select
fname = MyPath & "DairySort.xls"
If Dir(fname) <> "" Then Kill fname
Workbooks("ItSort.xls").SaveAs FileName:=fname
ActiveWindow.ScrollRow = 1
Sheets("Лист1").Protect Password:="I&R", DrawingObjects:=True,
Contents:=True, Scenarios:=True

```

```

ActiveWorkbook.Protect Password:="I&R", Structure:=True, Win-
dows:=True
End If
Else
a = MsgBox("Не найден файл ItSort.xls для формирования отчета!",
vbCritical, "Breads.")
End If
End Sub

```

### 3.3.5. Печать выходных документов

На каждом документе находится кнопка «Печать», которая вызывает процедуру myprint(), позволяющая печатать текущий документ. Перед печатью предусмотрен предварительный просмотр.

```

Public Sub myprint()
ActiveSheet.PrintPreview
msg = "Будете выводить на принтер?"
a = MsgBox(msg, vbDefaultButton2 + vbQuestion + vbYesNo,
"Breads")
If a = vbYes Then
ActiveWindow.SelectedSheets.PrintOut Copies:=1
ActiveSheet.PageSetup.PrintArea = ""
End If
End Sub

```

### 3.3.6. Корректировка первичной информации

При запуске системы автоматически происходит активизация базы данных. Кроме того, повторная активизация происходит при корректировке базы данных: добавления нового вида хлеба, корректировка существующего и удаление.

```

Sub activBD()
Application.ScreenUpdating = False
If Dir(MyPath & "BreadBD.xls") <> "" Then
Workbooks.Open MyPath & "BreadBD.xls"
Sheets("Лист1").Select
ActiveWorkbook.Unprotect Password:="I&R"
ActiveSheet.Unprotect Password:="I&R"
For i = 1 To 100
If ActiveSheet.Cells(i + 14, 4).Value <> "" Then
difsort(i) = ActiveSheet.Cells(i + 14, 4).Value
paysort(i) = Val(ActiveSheet.Cells(i + 14, 5).Value)
Else
Exit For

```

```

End If
Next i
howmuch = i - 1
Range("E14").Value = Str(howmuch) ' вношу кол-во
Range("D15").Select
Sheets("Лист1").Protect Password:="I&R", DrawingObjects:=True,
Contents:=True, Scenarios:=True
ActiveWorkbook.Protect Password:="I&R", Structure:=True, Win-
dows:=True
ActiveWorkbook.Save
Workbooks("BreadBD.xls").Close savechanges:=False
Else
a = MsgBox("Не найден файл базы данных!", vbCritical, "Breads.")
Workbooks.Open MyPath & "MainBread.xls"
Worksheets("Лист1").Activate
ActiveWindow.ScrollRow = 1
End If
End Sub

```

### 3.3.6.1. Добавление информации

В системе должна быть учтена возможность добавления информации (процедура `dobav()`) в таблицу исходных данных (базу данных). Для этого разработана форма (рис. 23) В форму вводятся данные о новом виде продукции и его цене, которые после нажатия кнопки ОК (`CommandButton1_Click()`) вносятся последней строкой в базу данных.

The image shows a dialog box with a blue title bar containing the word "Добавление" in white. The main area has a light gray background with a dotted pattern. It contains two text input fields. The first field is preceded by the text "Введите название хлебобулочного изделия" (Enter the name of the bakery product). The second field is preceded by "Введите цену (в руб.)" (Enter the price (in rubles)). To the right of the input fields is a small rectangular image showing several golden-brown, braided bread rolls. At the bottom of the dialog box are two buttons: "ОК" (OK) on the left and "Отмена" (Cancel) on the right.

Рис. 23. Форма для добавления данных

**Sub dobav()**

```
FlagDobavBD = False
dobavform.Show
If FlagDobavBD = True Then
Application.ScreenUpdating = False
ActiveWorkbook.Close
Call activBD
Workbooks.Open MyPath & "BreadBD.xls"
Sheets("Лист1").Select
End If
End Sub
```

**Private Sub CommandButton1\_Click()**

```
myNameBread = nbread.Text
Payment = paybread.Text
If myNameBread = "" Or Payment = "" Then
a = MsgBox("Неверный ввод!", vbOKOnly + vbDefaultButton1 +
vbCritical, "Breads")
Else
flag1 = True
For i = 1 To HowMuch
If DifSort(i) = myNameBread Then
a = MsgBox("Повторное название!", vbOKOnly + vbDefaultBut-
ton1 + vbCritical, "Breads")
flag1 = False
Exit For
End If
Next i
If flag1 <> False Then
For i = 1 To Len(Payment)
flag2 = False
For j = 1 To 10
If Mid(Payment, i, 1) = Chr(47 + j) Or Mid(Payment, i, 1) = "."
Or Mid(Payment, i, 1) = "," Then
flag2 = True
End If
Next j
If flag2 = False Then
a = MsgBox("Неверно введена цена!", vbOKOnly + vbDe-
faultButton1 + vbCritical, "Breads")
Exit For 'i
End If
Next i
If flag2 <> False Then 'все верно
```

```

FlagDobavBD = True
Application.ScreenUpdating = False
Sheets("Лист1").Select
Workbooks("BreadBD").Unprotect Password:="I&R"
Worksheets("Лист1").Unprotect Password:="I&R"
ActiveSheet.Cells(HowMuch + 15, 4).Value = myName-
Bread
DifSort(HowMuch + 1) = myNameBread
ActiveSheet.Cells(HowMuch + 15, 5).Value = Payment
PaySort(HowMuch + 1) = CCur(Payment)
HowMuch = HowMuch + 1
ActiveWindow.ScrollRow = 1
Sheets("Лист1").Protect Password:="I&R", DrawingOb-
jects:=True, Contents:=True, Scenarios:=True
Workbooks("BreadBD.xls").Protect Password:="I&R",
Structure:=True, Windows:=True
Workbooks("BreadBD").Save
For i = 1 To MaxBul 'цикл по всем булочным
' Проверка на присутствие файла в директории
If Dir(MyPath & "Bread" & Str(i) & ".xls") <> "" Then
Workbooks.Open MyPath & "Bread" & Str(i) & ".xls"
Sheets("Лист1").Select
ActiveWorkbook.Unprotect Password:="I&R"
ActiveSheet.Unprotect Password:="I&R"
' Надо вставить кол-во видов
Range("E4").Value = Str(HowMuch)
' А теперь еще одну строчку
Rows(HowMuch + 14).Select
Selection.Insert Shift:=xlDown
ActiveSheet.Cells(HowMuch + 14, 3).Value = Dif-
Sort(HowMuch)
ActiveSheet.Cells(HowMuch + 14, 5).Value = Pay-
Sort(HowMuch)
ActiveSheet.Cells(HowMuch + 14, 6).Value = 0
Call MyLines(HowMuch + 14, 3, 6)
Range("C15").Select
Sheets("Лист1").Protect Password:="I&R", DrawingOb-
jects:=True, Contents:=True, Scenarios:=True
ActiveWorkbook.Protect Password:="I&R", Structure:=True,
Windows:=True
ActiveWorkbook.Save
Workbooks("Bread" & Str(i) & ".xls").Close
End If

```

```

Next i
Unload dobavform
End If
End If
End If
End Sub

```

**Private Sub CommandButton2\_Click()**

```

Unload dobavform
End Sub

```

**Private Sub UserForm\_Initialize()**

```

nbread.Text = ""
paybread.Text = ""
End Sub

```

### 3.3.6.2. Корректировка информации

Система должна предусматривать возможность корректировки существующих данных, т.е. изменение цены и/или названия продукции. Для выполнения этой операции целесообразно разработать форму следующего вида (рис. 24).

Рис. 24. Форма для корректировки базы данных

Чтобы откорректировать нужный вид хлеба необходимо обратиться к таблице с исходными данными, установить на строку с корректируемой информацией и запустить процедуру вызова формы «Корректировка базы данных». Вызванная форма в полях «Корректируемый вид» и «Его цена» будет содержать данные для изменения. При нажатии на кнопку ОК происходит обновление данных в таблице.

### 3.3.6.3. Удаление информации

Аналогично операции «Корректировать» может быть реализована операция удаления информации из базы данных.

### 3.3.7. Выход из системы

Чтобы выйти из системы необходимо нажать на кнопку «Выход» в главном меню. Текст макроса.

#### Public Sub goout()

```
If MsgBox("Вы на самом деле хотите выйти из системы Bakery Info I&R v.1.0?", vbOKCancel + vbDefaultButton1 + vbQuestion, "Breads") = 1 Then
```

```
Application.ScreenUpdating = True  
Application.DisplayFullScreen = OldDisplay  
Application.DisplayFullScreen = False  
Workbooks("MainBread.xls").Close savechanges:=False  
End If  
End Sub
```

### 3.3.8. Корректировка отчетных документов

Для того, чтобы изменить данные в готовых отчетных документах по булочным, необходимо использовать кнопку «Изменить кол-во выпечки» на листе отчетного документа. При этом можно либо подвести курсор к тому виду, количество которого надо изменить, и тогда вызовется форма «Корректировка» (рис. 25) для этого конкретного вида.

Рис. 25. Форма для корректировки выпечки по сорту хлеба



Если же не подводить курсор ни к какому виду, то будет вызвана обобщенная форма «Изменение количества» (рис. 26).

Наименование	Ведите количество (в шт.)
хлебобулочного изделия	<input type="text"/>
	<input type="text"/>
	<input type="text"/>
	<input type="text"/>
	<input type="text"/>
	<input type="text"/>
	<input type="text"/>

Занести в документ

Очистить все

Отмена




Рис. 26. Форма для изменения выпечки по всем сортам хлеба

### 3.3.9. Операция поиска данных

В тех документах, где используются названия сортов хлеба, предусмотрена функция поиска. Для этого необходимо нажать кнопку «Поиск». Поиск можно осуществлять либо в текущем документе, либо выбрать номер булочной для открытия соответствующего документа. Также можно осуществлять поиск либо по названию хлеба, либо по его цене (рис. 27). В конце поиска курсор укажет необходимую ячейку.

Для вызова формы поиска (форма findform) необходимо нажать кнопку «Поиск» в основном меню системы. Процедура вызова формы приведена ниже.

```
Public Sub myfind()
```

```
findform.Show
```

```
End Sub
```

Для выполнения операций поиска разрабатывается набор процедур, которые связываются с соответствующими процедурами формы.

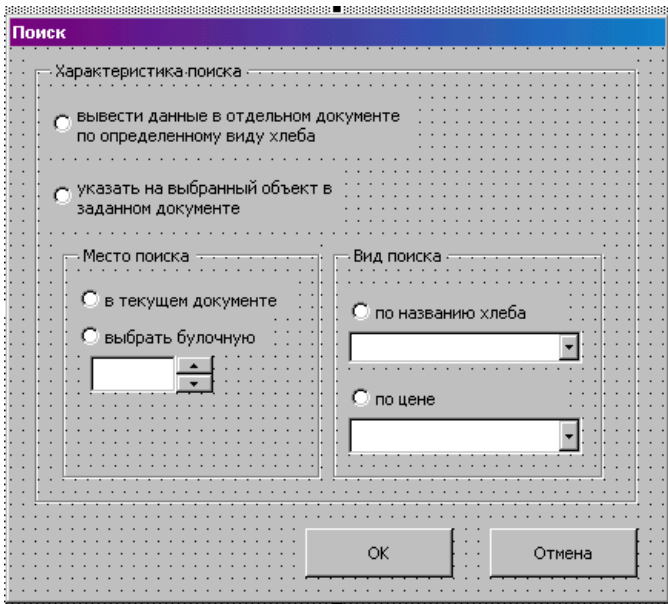


Рис. 27. Форма для выбора критериев поиска

### Private Sub CommandButton1\_Click()

```

If OptionButton2.Value = True Then
    If Dir(MyPath & "Bread" & Str(numbread) & ".xls") = "" Then
        msg = "Извините, но данных по булочной с номером " &
Str(numbread) & " нет."
        answer = MsgBox(msg, vbCritical + vbDefaultButton1 + vbOKOnly, "Breads")
    Else
        Application.ScreenUpdating = False
        ActiveWorkbook.Save
        ActiveWorkbook.Close
        Workbooks.Open MyPath & "Bread" & Str(numbread) & ".xls"
        Sheets("Лист1").Select
        If OptionButton3.Value = True Then
            ActiveSheet.Cells(15 + ComboBox1.ListIndex, 3).Select
        Else
            ActiveSheet.Cells(15 + ComboBox2.ListIndex, 5).Select
        End If
        findform.Hide
    End If
Else

```

```

    If OptionButton3.Value = True Then
        ActiveSheet.Cells(15 + ComboBox1.ListIndex, 3).Select
    Else
        ActiveSheet.Cells(15 + ComboBox2.ListIndex, 5).Select
    End If
    findform.Hide
End If
End Sub

Private Sub SpinButton1_SpinDown()
    TextBox1.Value = SpinButton1.Value
    numbread = Val(SpinButton1.Value)
End Sub

Private Sub SpinButton1_SpinUp()
    TextBox1.Value = SpinButton1.Value
    numbread = Val(SpinButton1.Value)
End Sub

Private Sub CommandButton2_Click()
    findform.Hide
End Sub

Private Sub OptionButton1_Click()
    SpinButton1.Enabled = False
End Sub

Private Sub OptionButton2_Click()
    SpinButton1.Enabled = True
End Sub

Private Sub OptionButton3_Click()
    ComboBox1.Enabled = True
    ComboBox2.Enabled = False
End Sub

Private Sub OptionButton4_Click()
    ComboBox1.Enabled = False
    ComboBox2.Enabled = True
End Sub

Private Sub UserForm_Activate()
    OptionButton1.Value = True
    OptionButton3.Value = True
    For i = 1 To howmuch
        ComboBox1.AddItem difsort(i)
        ComboBox2.AddItem Str(paysort(i))
    Next i

```

```

ComboBox1.ListIndex = 0
ComboBox2.ListIndex = 0
SpinButton1.Enabled = False
ComboBox2.Enabled = False
TextBox1.Text = 1
numbread = 1
End Sub

```

#### **Private Sub UserForm\_Initialize()**

```

OptionButton1.Value = True
OptionButton3.Value = True
For i = 1 To howmuch
    ComboBox1.AddItem difsort(i)
    ComboBox2.AddItem Str(paysort(i))
Next i
ComboBox1.ListIndex = 0
ComboBox2.ListIndex = 0
SpinButton1.Enabled = False
ComboBox2.Enabled = False
TextBox1.Text = 1
numbread = 1
End Sub

```

### **3.3.10. Работа со справкой системы**

При нажатии кнопки «Справка» в главном меню выполняются следующие действия.

#### **Public Sub showhelp()**

```

Application.ScreenUpdating = False
If Dir(MyPath & "BreadHelp.xls") <> "" Then
    ActiveWorkbook.Close savechanges:=False
    Workbooks.Open MyPath & "BreadHelp.xls"
    Sheets("Лист1").Select
Else
    a = MsgBox("Не найден файл подсказки BreadHelp.xls!", vbCritical,
"Breads.")
End If
End Sub

```

### **3.3.11. Обновление системы**

При нажатии кнопки «Обновить систему» происходит удаление всех рабочих файлов, относящихся к системе (если они существуют). Это DairyWork.xls, DairyBul.xls, DairySort.xls и Bread n.xls, где n – номера булочных.

### **Public Sub DelAll()**

```
msg = "Последнее обновление было " & ActiveSheet.Range("Z1").  
Value & ".Вы уверены, что хотите вновь обновить всю систему? Сего-  
дня " & Date  
a = MsgBox(msg, vbDefaultButton2 + vbQuestion + vbYesNo,  
"Breads")  
If a = vbYes Then  
Application.ScreenUpdating = False  
For i = 1 To max 'цикл по всем булочным  
fname = MyPath & "Bread" & Str(i) & ".xls"  
If Dir(fname) <> "" Then Kill fname  
Next i  
If Dir(MyPath & "DairyWork.xls") <> "" Then Kill MyPath & "Dairy-  
Work.xls"  
If Dir(MyPath & "DairyBul.xls") <> "" Then Kill MyPath & "Dairy-  
Bul.xls"  
If Dir(MyPath & "DairySort.xls") <> "" Then Kill MyPath & "Dairy-  
Sort.xls"  
Workbooks("MainBread.xls").Activate  
Sheets("Лист1").Select  
ActiveWorkbook.Unprotect Password:="I&R"  
ActiveSheet.Unprotect Password:="I&R"  
ActiveSheet.Range("Z1").Value = Date  
ActiveSheet.Protect Password:="I&R", DrawingObjects:=True, Con-  
tents:=True, Scenarios:=True  
ActiveWorkbook.Protect Password:="I&R", Structure:=True, Win-  
dows:=True  
ActiveWorkbook.Save  
ActiveWindow.ScrollRow = 1  
End If  
End Sub
```

## **4. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ К КУРСОВОЙ РАБОТЕ**

Для выполнения курсовой работы студенту необходимо выбрать один из вариантов задания. Каждый вариант задания включает в себя краткую постановку задачи и требования к получаемым результатам.

Каждая разрабатываемая система должна предоставлять пользователю возможность ввода и обновления информации, расчета итогов, формирование выходных документов и просмотр инструкций. Предусмотреть в программе наличие меню.

Данные обрабатываемые системой, должны быть проанализированы с использованием графических возможностей используемого пакета

Ниже представлен список вариантов индивидуальных заданий:

### **1. Расчет выручки автомобильного завода от реализации продукции**

Ежедневно автомобильный завод выпускает до 100 автомобилей которые поступают в дилерские конторы и в магазины. Оператор вводит в таблицу данные о количестве поставленных автомобилей в каждую контору или магазин. Производится расчет стоимости автомобилей по моделям и общей стоимости автомобилей. В конце работы формируется итоговая таблица, результаты расчета распечатываются.

### **2. Расчет выручки цветочного салона от реализации продукции за 1 месяц**

Салон цветов периодически закупает цветочные изделия в государственных и частных теплицах, а также подручный материал и искусственную цветочную продукцию для аранжировок. Клиент может сделать заказ непосредственно в салоне и в агентствах салона в других городах. Оператор вводит в таблицу данные о заказах. Разрабатываемая система должна рассчитывать возможный доход, получаемый данным предприятием за 1 месяц работы без учета расходов за содержание помещения, оборудования и заработную плату работников. Расчет производится раз в месяц.

### **3. Расчет выручки телефонной станции за 1 месяц работы**

Ежедневно телефонная станция обрабатывает заказы на переговоры частных лиц и предприятия. Оператор вносит в таблицу данные о заказах. Разрабатываемая система должна рассчитывать ежедневную выручку телефонной станции, на основе имеющихся тарифов, а также на основе ежедневных таблиц должна формироваться итоговая таблица о доходах за месяц.

#### **4. Система учета посещений и оплаты в детском саду**

Каждый месяц в детском саду формируются итоговые ведомости оплат в каждой группе, а также сводная ведомость платежей. Оператор вводит данные о посещаемости детей в каждой группе. Разрабатываемая система должна формировать ведомости оплат по каждой группе и сводную ведомость и производить расчет по оплате для каждого ребенка.

#### **5. Расчет выручки ЗАГСа от услуг, оказываемых клиентам за неделю**

Каждый день в отделение ЗАГСА обращаются посетители. Ежедневно данные об оказанных услугах, оператор заносит в таблицу. Разрабатываемая система должна рассчитывать итоговый доход за оказанные населению услуги в течении недели и выручку за отдельные виды услуг.

#### **6. Расчет выручки теплицы от реализации продукции**

В теплицу ежедневно поступают предложения и заявки на товары (овощи), а так же информация о выручке, поступившей от продажи товара с места и ларьков за день. По мере поступления информации, оператор вводит в таблицу данные о количестве проданных овощей. Производится расчет выручки от торговли. К концу работы формируется итоговая таблица. Результаты работы распечатываются.

#### **7. Расчет выручки КБО от услуг, оказываемых клиентам**

Ежедневно в Комбинате бытового обслуживания оказываются услуги в парикмахерском, маникюрном и косметологическом салонах. По мере появления нового клиента и выполнения для него услуги оператор вводит в таблицу данные о клиенте и услуге.

Разрабатываемая система должна рассчитывать итоговую стоимость услуг по подразделениям, общий доход КБО и формировать расчетную квитанцию каждому клиенту. Разработанная система должна предоставлять пользователю возможности ввода и обновления информации, расчета и формирования выходных документов, просмотр инструкций.

#### **8. Расчет выручки агентства по продаже недвижимости**

Ежедневно в различных районах города агентство по продаже недвижимости продает квартиры с различной стоимостью одного квадратного метра. Стоимость квадратного метра зависит от района города. Ежедневно оператор вводит в таблицу данные о проданных и купленных агентством квартирах. Производится расчет ежедневной общей выручки агентства от операций купли-продажи, отдельно производится

расчет затрат по покупке квартир и дохода от продажи. Результаты расчета распечатываются.

### **9. Расчет выручки мебельной фабрики от реализации продукции**

Ежедневно на мебельной фабрике производится отправка в магазин и продажа на месте различных видов мебели по разной цене. По мере отправки в магазин или продажи на месте оператор вводит данные о количестве реализованных изделий. Разрабатываемая система должна производить расчет стоимости проданной продукции по видам и общей выручки предприятия. В конце работы формируются итоговые таблицы, результаты расчета распечатываются.

### **10. Система учета денежных вкладов клиентов банка**

Ежедневно банк осуществляет работу с клиентами. Клиент может положить или снять со счета деньги. По мере поступления денег в банк, оператор вводит в таблицу данные о сумме вклада на каждый счет или о количестве изымаемых средств. Ежедневно происходит пересчет денежных сумм по обновленным вкладам с учетом начисляемых процентов. В конце рабочего дня формируется итоговая таблица по всем вкладам граждан. Результаты расчетов распечатываются.

### **11. Расчет выручки диагностического центра от услуг, оказываемых населению**

Ежедневно в диагностическом центре людям оказывают различные услуги. Ежедневно оператор вводит в таблицу данные об оказанных услугах. Производится расчет стоимости услуг для каждого клиента на основе имеющихся тарифов и общей выручки центра за оказанные услуги. Разрабатываемая система формирует документ по расчету с клиентом, таблицы выручки по каждому виду услуг и итоговую таблицу полученной выручки за день.

### **12. Расчет выручки птицефабрики от реализации продукции**

Ежедневно продукция птицефабрики, выпущенная до обеда, поступает в магазины, а выпущенная после обеда на склад готовой продукции. Со склада она поступает в розничную продажу с машин этой птицефабрики, нереализованный товар возвращается на склад. Система должна рассчитывать реальный доход за 1 день от продажи продукции и часть возможного дохода за следующий день с учетом того, что вся нереализованная продукция будет реализована завтра.



### **13. Система расчета калорийности блюд в столовой**

Ежедневно столовая предоставляет клиентам меню из различных блюд. Меню составляется с учетом калорийности каждого блюда и не должно превышать общепринятых норм. Оператор вводит в таблицу меню на один день. Система должна рассчитывать калорийность каждого блюда, используя таблицы калорийности продуктов и их массу, а также количество калорий во всех блюдах.

### **14. Расчет выручки станции технического обслуживания автомобилей**

Ежедневно в автоцентр поступают на ремонт различные марки автомобилей. Автомобили поступают в течение дня. Оператор вводит в таблицу данные об автомобиле и видах необходимых услуг. Система должна рассчитывать выручку по каждому виду услуг с учетом имеющихся тарифов и общую выручку автоцентра за один день.

### **15. Расчет выручки молочного комбината от реализации продукции**

Ежедневно молочный комбинат выпускает продукцию различного вида. Оператор вводит в таблицу данные о количестве реализованной за день продукции каждого вида. Система должна учитывать, что продажа продукции молококомбината осуществляется через магазины и с машин предприятия. В конце работы формируется итоговая таблица общей выручки предприятия за день и таблицы дохода по каждому виду продукции.

### **16. Расчет ежедневной выручки КПП таможи**

Ежедневно граждане, въезжающие из-за границы, на КПП заполняют таможенные декларации. Оператор вводит эти данные в систему, которая на основе таблицы о пошлинах по каждому виду товара формирует таблицы о суммарной пошлине, уплаченной каждым клиентом, и сводную таблицу о суммарном доходе, получаемом за день работы КПП.

### **17. Расчет выручки объединения "Дальпушнина" от реализации продукции**

Ежедневно в объединение "Дальпушнина" поступают заявки, предложения на товары (изделия из меха), а также информация о выручке с продаж в магазинах за день. По мере поступления информации, оператор вводит в таблицу данные о количестве проданных товаров. Производится расчет выручки торговли. К концу работы формируются итоговые таблицы общей выручки и по отдельным видам товаров. Результаты работы распечатываются.

## **18. Расчет выручки частной автозаправочной станции**

Ежедневно автозаправочная станция продает различные виды топлива государственным предприятиям и частным лицам. Ежедневно оператор вводит данные о количестве проданного топлива. Разрабатываемая система должна рассчитывать общую выручку предприятия на день, учитывая затраты на заработную плату сотрудникам и содержание оборудования, а также доход по каждому виду топлива.

## **19. Расчет выручки газеты "Дальпресс" от рекламы**

Выручка газеты "Дальпресс" формируется из стоимости рекламы коммерческих фирм. Ежедневно оператор формирует таблицу о напечатанной рекламе за текущий день. В конце месяца производится расчет общей выручки газеты от рекламы и выручка, полученная газетой, от каждой коммерческой фирмы.

## **20. Система: "АРМ кассира ДВ пароходства"**

Пароходство осуществляет перевозку пассажиров. Разрабатываемая система должна рассчитывать прибыль от перевозки пассажиров и формировать проездной билет пассажира на основе полученного заказа, расписания движения судов, наличия свободных мест и имеющихся тарифов на билеты. Перевозку на каждом направлении могут осуществлять несколько пароходов.

## **21. Расчет выручки отдела по продаже видеотехники универмага**

Ежемесячно в отдел универмага поступает видеотехника от нескольких поставщиков в ассортименте по различной цене. Розничная цена товара формируется в зависимости от спроса. Оплата поставок осуществляется по мере реализации. Ежедневно оператором формируются таблицы о проданном товаре. Разрабатываемая система должна рассчитать общую выручку отдела за месяц, определить спрос по каждому виду товара и выдать предложения по изменению цен на продукцию.

## **22. Система расчета выручки магазина от продажи товаров**

Ежедневно со склада магазина в отделы поступают товары в ассортименте по различным ценам. В разрабатываемой системе хранится информация о наличии товара на складе и в отделах. Ежедневно оператор вводит в систему данные о поступивших в отделы товарах. Производится расчет выручки от продажи товара по каждому отделу и общей выручки магазина, кроме того, корректируются таблицы о наличии товара в отделах и формируется таблица заявок на следующий день.

### **23. Система расчета выручки юридической консультации**

В юридической консультации работает несколько юристов, дающих консультации как населению, так и юридическим лицам. Ежедневно оператор вводит данные об услугах, оказанных юристами. Разрабатываемая система должна рассчитывать на основе, имеющихся тарифов, общую выручку юридической консультации, доход каждого юриста и отчисления в бюджет.

### **24. Расчет выручки кондитерской фабрики от реализации продукции**

На кондитерскую фабрику ежедневно поступают заявки на продукцию от торговых фирм и собственных фирменных магазинов, а так же информация о реализации этой продукции за день. По мере поступления информации, оператор вводит в таблицу данные о количестве проданной продукции. Система должна производить расчет выручки от торговли (учитывая разницу в оптовых и розничных ценах). К концу работы формируется итоговая таблица. Результаты работы распечатываются.

### **25. Расчет выручки салона красоты от услуг, оказываемых клиентам**

Ежедневно в салоне красоты оказывают услуги в парикмахерском, маникюрном и косметологическом отделениях. По мере появления нового клиента и выполнения для него услуги оператор вводит в таблицу данные о клиенте и услуге. Разрабатываемая система должна рассчитывать итоговую стоимость услуг по подразделениям, общий доход КБО и формировать расчетную квитанцию каждому клиенту. В системе должно быть предусмотрено, что салон может обслужить за день только определенное число клиентов, и то, что заявки на обслуживание могут поступать заранее.

### **26. Расчет выручки телерадиокомпании от рекламной деятельности**

Выручка телерадиокомпании формируется из стоимости одной минуты рекламы коммерческих фирм. Ежедневно оператор формирует таблицу о выпущенной в эфир рекламе за текущий день и вводит информацию о вновь поступающих заявках. В конце месяца производится расчет общей выручки компании от рекламы и выручки, полученной каждой передачей. Проводится анализ популярности передач и корректируется стоимость рекламы.

## **27. Система: «АРМ кассира авиакомпании»**

Авиакомпания осуществляет перевозку пассажиров по различным направлениям. Разрабатываемая система должна рассчитывать дневную выручку от перевозки пассажиров и формировать проездной билет пассажира на основе полученного заказа, расписания движения самолетов, наличия свободных мест и имеющихся тарифов на билеты.

## **28. Расчет выручки авиакомпании**

Авиакомпания осуществляет перевозку пассажиров в различные регионы страны. Возможно наличие нескольких рейсов в день по одному направлению. Ежедневно оператор вводит данные о количестве проданных билетов. Разрабатываемая система должна рассчитывать общую выручку авиакомпании за месяц, выручку за каждый день и анализировать данные о рентабельности существующего расписания движения самолетов.

## **29. Система: "АРМ кассира железной дороги"**

Железная дорога осуществляет перевозку пассажиров по различным направлениям. Разрабатываемая система должна рассчитывать общую выручку от перевозки пассажиров и формировать проездной билет пассажира на основе полученного заказа, расписания движения поездов, наличия свободных мест и имеющихся тарифов на билеты.

## **30. Расчет выручки КБО "Снежинка" от оказываемых услуг**

Ежедневно в Комбинате бытового обслуживания оказываются услуги в прачечной и химчистке. По мере появления нового клиента и выполнения для него услуги оператор вводит в таблицу данные о клиенте и услуге. Разрабатываемая система должна рассчитывать итоговую стоимость услуг по подразделениям, общий доход КБО и формировать расчетную квитанцию каждому клиенту.

## **5. СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ**

### **5.1. Основная литература**

1. Уокенбах Д. Профессиональное программирование на VBA в Excel 2002. – М.: Диалектика, 2003.
2. Биллиг В.А. Средства разработки VBA-программиста. Офисное программирование. Том 1. – СПб.: Русская редакция, 2001.
3. Роман С. Использование макросов в Excel. 2-е изд. – СПб.: Питер, 2004.
4. Visual Basic для приложений (версия 5) в подлиннике П. Сканна и др. – СПб.: «ВНУ-Санкт-Петербург», 1997.

### **5.2. Дополнительная литература**

5. Гарнаев А. Самоучитель VBA. – СПб., 1999.
6. Информатика. Базовый курс: Учеб. пособие для студ. высш. технич. учеб. завед. / Под ред. С.В. Симоновича. – СПб.: Питер, 2002. – 638 с.
7. Острейковский В.А. Информатика: Учеб. пособие. – М.: Высш. шк., 2000. – 319 с.: ил.
8. Информатика: Учебник / Под ред. Н.В. Макаровой. – 3-е изд., перераб. – М.: Финансы и статистика, 2001. – 768 с.: ил.
9. Рыжиков Ю.И. Информатика. Лекции и практикум. – СПб.: КОРОНА принт, 2000. – 256 с.

# ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	3
1. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ .....	4
1.1. ЦЕЛЬ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ .....	4
1.2. ТРЕБОВАНИЯ К ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ.....	4
1.3. ОБЪЕМ И СОДЕРЖАНИЕ КУРСОВОЙ РАБОТЫ .....	4
1.4. ТЕХНИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ.....	6
1.5. ВИДЫ КОНТРОЛЯ ЗНАНИЙ СТУДЕНТОВ И ИХ ОТЧЕТНОСТИ ПО КУРСОВОЙ РАБОТЕ.....	6
2. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ. РАБОТА В РЕДАКТОРЕ VBA.....	7
2.1. АВТОМАТИЧЕСКАЯ ЗАПИСЬ МАКРОСА И ЕГО РЕДАКТИРОВАНИЕ .....	7
2.2. СОЗДАНИЕ МАКРОСА В РЕДАКТОРЕ VBA .....	8
2.3. СОЗДАНИЕ ПРОЕКТА В РЕДАКТОРЕ VBA .....	8
2.4. ОПИСАНИЕ РЕДАКТОРА VBA .....	9
2.5. ДОБАВЛЕНИЕ МОДУЛЯ В ПРОЕКТ.....	11
2.5.1. Использование окна модуля.....	11
2.5.2. Редактирование кода в окне модуля.....	12
2.6. ОПИСАНИЕ ПЕРЕМЕННЫХ И ОБЪЕКТОВ .....	12
2.7. ИСПОЛЬЗОВАНИЕ ПРОЦЕДУР.....	13
2.8. ОПИСАНИЕ СВОЙСТВ И МЕТОДОВ.....	14
2.9. СОЗДАНИЕ ФОРМ.....	15
2.9.1. Добавление форм в проект .....	15
2.9.2. Инструменты панели элементов .....	15
2.9.3. Размещение элементов управления в форме .....	16
2.9.4. Просмотр формы .....	16
2.9.5. Написание процедур для элементов управления в формах .....	17
2.9.6. Запуск форм, процедур и макросов .....	18
2.10. ОБЪЕКТЫ, СВОЙСТВА И МЕТОДЫ VBA .....	18
2.10.1. Элементы управления в приложениях VBA.....	20
3. ПРИМЕР СОЗДАНИЯ ПРИЛОЖЕНИЯ .....	26
3.1. ПОСТАНОВКА ЗАДАЧИ .....	26
3.2. АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ .....	26
3.3. ОПИСАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ СИСТЕМЫ .....	28
3.3.1. Структура основного макроса.....	28
3.3.2. Структура макроса для первичного ввода данных.....	30

3.3.3. Корректировка документа .....	34
3.3.4. Формирование итоговых документов .....	35
3.3.5. Печать выходных документов .....	43
3.3.6. Корректировка первичной информации .....	43
3.3.7. Выход из системы .....	48
3.3.8. Корректировка отчетных документов .....	48
3.3.9. Операция поиска данных.....	49
3.3.10. Работа со справкой системы .....	52
3.3.11. Обновление системы .....	52
4. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ К КУРСОВОЙ РАБОТЕ.....	54
5. СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ .....	61
5.1. ОСНОВНАЯ ЛИТЕРАТУРА .....	61
5.2. ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА .....	61

Учебное издание

## **ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ**

Руководство по выполнению курсовых работ

Сост. Бедрина Светлана Львовна, Богданова Ольга Борисовна

В авторской редакции  
Компьютерная верстка С.Ю. Заворотной

Лицензия на издательскую деятельность ИД № 03816 от 22.01.2001

Подписано в печать 25.03.2005. Формат 60×84/16.  
Бумага писчая. Печать офсетная. Усл. печ. л. 3,72.  
Уч.-изд. л. 3,05. Тираж 85 экз. Заказ

---

Издательство Владивостокского государственного университета  
экономики и сервиса  
690600, Владивосток, ул. Гоголя, 41  
Отпечатано в типографии ВГУЭС  
690600, Владивосток, ул. Державина, 57



