

## Содержание

	Введение .....	5
1	Пользовательский интерфейс VBA .....	7
1.1	Особенности среды программирования .....	7
1.2	Запуск VBA .....	8
1.3	Структура редактора Visual Basic .....	9
1.3.1	Окно проекта Project-VBAProject.....	9
1.3.2	Окно редактирования кода .....	11
1.3.3	Окно редактирования форм .....	14
1.3.4	Окно свойств “Properties” .....	16
1.3.5	Окно панели элементов управления (Toolbox).....	17
2	Разработка программного кода .....	17
2.1	Структура проекта VBA.....	17
2.2	Пользовательские подпрограммы-функции .....	18
2.3	Пользовательские подпрограммы-процедуры.....	20
2.4	Данные и их описание .....	20
2.4.1	Алфавит и лексемы языка.....	20
2.4.2	Переменные и константы.....	21
2.4.3	Объявление переменных.....	22
2.4.4	Объявление констант.....	23
2.4.5	Строковые переменные .....	24
2.4.6	Области видимости переменных и констант .....	24
2.4.7	Типы данных .....	25
2.5	Функции в VBA .....	27
2.5.1	Математические встроенные функции.....	27
2.5.2	Математические функции, не представленные в VBA.....	28
2.5.3	Функция форматирования данных.....	28
2.5.4	Функции преобразования типов.....	30
2.6	Операторы, выражения и операции .....	30
2.6.1	Операция присваивания .....	32
2.6.2	Математические операции.....	32
2.6.3	Операции отношения .....	33
2.6.4	Логические операции .....	34
2.6.5	Строковые операции .....	35
2.7	Отладка программ и обработка ошибочных ситуаций .....	35
2.7.1	Отладка .....	37
2.7.2	Точки останова (контрольные точки).....	38
2.7.3	Наблюдение за данными с помощью просматриваемых значений .....	39
2.7.4	Пошаговое выполнение программы .....	41
2.7.5	Обработка ошибок.....	42
2.8	Условный оператор .....	49
2.9	Оператор выбора (переключатель) .....	62
2.10	Операторы цикла .....	65
2.10.1	Циклы с параметром For ... Next .....	65

2.10.2 Циклы с условием (итерационные).....	67
2.11 Табулирование функции .....	75
2.11.1 Табулирование функции одной переменной .....	75
2.11.2 Табулирование функции двух переменных (вложенные циклы) .....	76
2.12 Массивы.....	81
2.12.1 Объявление массива .....	82
2.12.2 Динамические массивы .....	83
2.12.3 Действия над массивами .....	84
2.12.4 Действия над элементами массива.....	85
2.12.5 Формирование массива .....	85
2.12.6 Одномерные массивы.....	87
2.12.7 Двумерные массивы. Вложенные циклы. ....	89
3 Объектно-ориентированное программирование на VBA.....	96
3.1 Разработка программы создания приложения с помощью Форм.....	96
3.2 Встроенные диалоговые окна в VBA.....	109
3.2.1 Окна сообщений (MsgBox) .....	109
3.2.2 Окна ввода .....	112
3.3 Объекты формы в VBA: Кнопки-переключатели, Контрольные индикаторы, Рамки.....	119
3.4 Объекты формы в VBA: Полоса прокрутки и Счетчик .....	128
3.5 Объекты формы в VBA: Список и Поле со списком .....	135
3.5.1 Заполнение списка и удаление его элементов .....	137
3.5.2 Список с несколькими столбцами.....	138
3.5.3 Многоэлементный выбор из списка .....	141
3.5.4 Добавление и удаление данных.....	144
3.6 Объект формы в VBA: Рисунок .....	146
3.6.1 Добавление растровых изображений в форму.....	147
3.6.2 Вставка изображений в форму .....	148
3.7 Интеграция приложений: MS Excel и MS Word.....	151
3.7.1 Открытие документа MS Word функцией CreateObject.....	151
3.7.2 Открытие документа MS Word функцией GetObject .....	152
4 Литература, рекомендуемая для изучения .....	161

## Введение

Современный учебный план для специалистов экономического профиля построен таким образом, что предусматривает непрерывность и преемственность изучения дисциплин экономико-математического инструментария и эконометрики, образующих единый комплекс.

В настоящее время без использования информационных технологий невозможно представить изучение такого комплекса дисциплин. Дисциплина «Информатика» включена в учебные планы экономических специальностей как естественно-научная дисциплина Государственного образовательного стандарта, который содержит обязательный раздел для изучения алгоритмизации и программирования.

Изучая информатику, студенты должны осознавать, что компьютер является рабочим инструментом для решения профессиональных задач. Любой специалист в своей области должен знать, какие профессиональные задачи можно решать, используя компьютер, и уметь это делать. Таким образом, информатика является предметом не только общеобразовательным, но и профессиональным, в котором очень важна система межпредметных связей, поэтому, по мнению авторов, методическое обеспечение дисциплины «Информатика» при изучении различных разделов курса должно ориентироваться на задачи из предметной области.

Особое внимание при обучении информатике студентов-экономистов уделяется изучению возможностей работы с объектами MS Office, в частности, табличному процессору, позволяющему автоматизировать проведение типовых вычислений. Современная версия Excel 2000/XP представляет собой открытую структуру, которая может быть использована в качестве базовой компоненты программного обеспечения специалиста-экономиста.

Вместе с тем, многообразие экономической деятельности обусловило появление различных специфических задач, средства решения которых отсутствуют в табличных процессорах или не соответствуют их трудоемкости и сложности. В этой связи необходимо рассмотреть варианты создания дополнительных модулей и надстроек. Решать подобные задачи, а также расширять возможности стандартных приложений MS Office позволяет алгоритмический язык высокого уровня Visual Basic for Application (VBA), который предлагается авторами в качестве базового для изучения раздела алгоритмизации и программирования студентами экономического профиля.

VBA – система визуального программирования, которая эффективно применяется для автоматизации деятельности, связанной с обработкой экономических данных: выполнения расчетов, составления сводных отчетов, решения задач анализа и прогнозирования, экономико-математического моделирования. Основы для возможности решения подобных задач закладываются при выработке навыков алгоритмического мышления и овладении постановкой задач на ЭВМ.

Пользователь может создавать видимую часть приложения. Программный интерфейс разрабатывается на современных принципах объектно-

ориентированного подхода, реализованного в VBA применительно к приложениям, выполняемым под управлением Windows. Основные принципы структурного программирования и объектно-ориентированный подход целесообразно преподавать в курсе информатики, ориентируясь на дальнейшие возможности быстрого и квалифицированного освоения студентами профессиональных пакетов прикладных программ.

В учебном пособии в доступной форме рассмотрены основные средства и возможности языка VBA для повышения гибкости и расширения функциональности офисного приложения Excel. Описан синтаксис языка, даны практические рекомендации по объектно-ориентированному программированию, показано конструирование пользовательского интерфейса, как на этапе дизайна, так и во время выполнения приложения. Рассмотрен вопрос взаимодействия офисных приложений между собой, представлены материалы для самостоятельной работы по офисному программированию.

По мнению авторов, экономическая направленность заданий для выполнения лабораторных работ повысит мотивацию студентов к изучению возможностей языка VBA, развитию алгоритмического мышления на примерах простых задач, связанных с будущей профессиональной деятельностью.

Основной целью учебного пособия авторы видят в стремлении использовать программное обеспечение в учебном процессе для создания уникальной среды, в которой экономическая обработка данных становится не рутинным занятием, а увлекательным исследованием, позволяющим получать многовариантные решения с использованием компьютерных технологий и современных методов алгоритмизации и программирования.

# **1 Пользовательский интерфейс VBA**

## **1.1 Особенности среды программирования**

Цель: знакомство с интегрированной средой программирования VBA.

По сравнению с предыдущими поколениями Бейсик-систем MS Visual Basic for Applications (VBA) имеет ряд принципиальных отличий. VBA - мощная система программирования, позволяющая быстро и эффективно создавать многофункциональные интегрированные офисные приложения для MS Windows 95/98-XP. Язык VBA – это подмножество Visual Basic (VB), которое включает почти все его средства создания приложений, структуры данных и управляющие структуры, возможность создания пользовательских типов данных.

VBA отличается от VB и прочих языков программирования в следующем:

- предоставляет возможность непосредственной работы с объектами MS Office. Это позволяет эффективно его использовать для автоматизации деятельности, связанной с обработкой различных типов документов.
- VBA позволяет существенно расширить вычислительные средства MS Office;
- с помощью VBA можно легко и быстро создавать различные приложения, даже не являясь программистом;
- VBA имеет графическую инструментальную среду, обеспечивающую конструирование экранных форм и управляющих элементов;
- с помощью VBA можно программировать собственные пользовательские функции, разрабатывать макросы, создавать пользовательские меню и многое другое;
- VBA позволяет с легкостью решать задачи, которые средствами обычных офисных приложений практически решить невозможно;
- VBA реализует концепцию визуального программирования, управляемого событиями. Этот язык помогает интегрировать офисные проекты в единое целое.
- VBA тесно связан с VB – компоненты, сконструированные на VB, легко используются в офисных приложениях, а VB-приложения посредством технологии Automation имеют доступ ко всем возможностям и средствам офисных приложений.

Поскольку система является «визуальной», программист (пользователь) может создавать видимую часть приложения Windows. Эта часть является основой интерфейса «программа-пользователь», с помощью которого осуществляется взаимодействие пользователя с программой. Разработка программного интерфейса осуществляется на принципах объектно-ориентированного подхода, реализованного в VBA применительно к приложениям, выполняемым под управлением Windows.

Для таких приложений характерным является существование на экране в любой момент времени множества объектов: окон, кнопок, меню, текстовых и диалоговых окон, линеек прокрутки и т.п. Пользователь имеет определенную свободу выбора в части использования этих объектов.

Для каждого объекта на экране существует ряд возможных событий. Одни из них производятся пользователем: щелчок по кнопке мыши, перетаскивание объекта, нажатие клавиши клавиатуры, ввод текста в окно и т.п.

Каждое событие проявляется в определенных действиях (откликах, реакции, поведении) программы. Эти действия можно разделить на две группы.


Первая является следствием свойств объекта. Эти свойства устанавливаются программистом из некоторого стандартного перечня свойств, заданного системой программирования VBA и самой системой Windows. Примером таких действий является свертывание окна после щелчка по кнопке <Свернуть>.

Вторая группа действий (отклика) на события полностью определяется программистом. Это обеспечивается возможностью задания процедуры VBA для каждого возможного для объекта события. Теоретически можно создать процедуру для любого события, но на практике программист заполняет кодом процедуры только для тех событий, которые в данной программе представляют интерес.

Таким образом, VBA предоставляет пользователю объектно-ориентированную среду программирования, основанную на событиях. Процесс разработки программы в этой среде сводится к выбору набора объектов и их свойств, заданию событий и процедур их обработки, которые в совокупности обеспечивают решение поставленной задачи.


## 1.2 Запуск VBA

Запуск интегрированной среды разработки приложений IDE (Integrated Development Environment – интегрированная среда разработки) редактора VBA осуществляется через запуск Windows-приложения MS Excel следующими способами:

- 1) выбрать последовательность команд Сервис→Макрос→Редактор Visual Basic;
- 2) нажать комбинации клавиш <Alt> + <F11>;
- 3) нажать кнопку  Редактор Visual Basic панели инструментов Visual Basic MS Excel.

После запуска программы на экран будет выведено окно редактора VBA, показанного на рисунке 1.

Возвратиться из редактора Visual Basic в рабочую книгу можно:

- 1) выбрать команды View (Вид)→MS Excel;
- 2) нажать кнопку  View MS Excel панели инструментов Standard.

Главное окно содержит все стандартные элементы, которыми обладает любое Windows-приложение: строку заголовка, строку управляющего меню, строку панелей инструментов, два окна Project-VBAProject и Properties.

Строка заголовка содержит название проекта Microsoft Visual Basic–Книга1.

Строка управляющего меню обеспечивает доступ к большинству команд, управляющих средой программирования. Меню и команды работают в соответствии со стандартными соглашениями, общими для всех Windows-приложений.

Под строкой управляющего меню расположена строка панели инструментов (Standard), представляющая собой набор кнопок, являющихся ярлыками для команд, с помощью которых осуществляется работа в среде VBA.

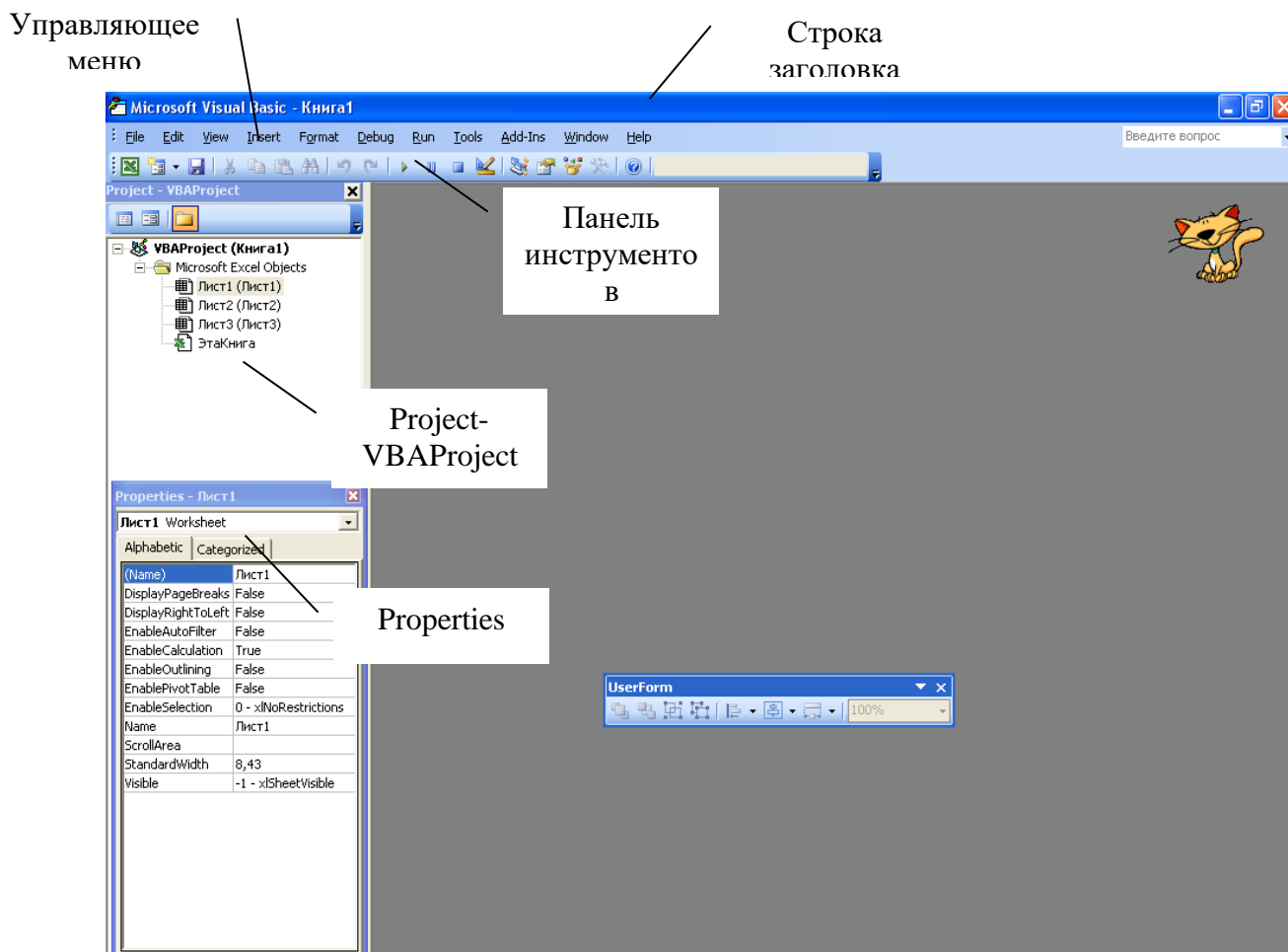



Рисунок 1 – Вид окна редактора VBA

## 1.3 Структура редактора Visual Basic

Пользовательский интерфейс редактора Visual Basic включает следующие основные компоненты:

- окно проекта Project-VBAProject;
- окно редактирования кода;
- окно редактирования форм;
- окно свойств;
- панель элементов управления.

### 1.3.1 Окно проекта Project-VBAProject

Окно проекта Project-VBAProject в редакторе Visual Basic активизируется выбором команды View→Project Explorer (Вид→Окно проекта) или нажатием кнопки Project Explorer  панели инструментов Standard.

В окне проекта Project-VBAProject, изображенного на рисунке 2, представлена иерархическая структура файлов форм и модулей текущего проекта.

В нем перечисляются все файлы проекта, доступ к которым осуществляется при помощи двух кнопок:

- <View code> (Просмотр кода);
- <View object> (Просмотр объекта).

Файл проекта имеет расширение \*.xls и содержит список всех файлов проекта. В окне отображается его структура в виде дерева, которое похоже на структуру папок в окне Explorer Windows.

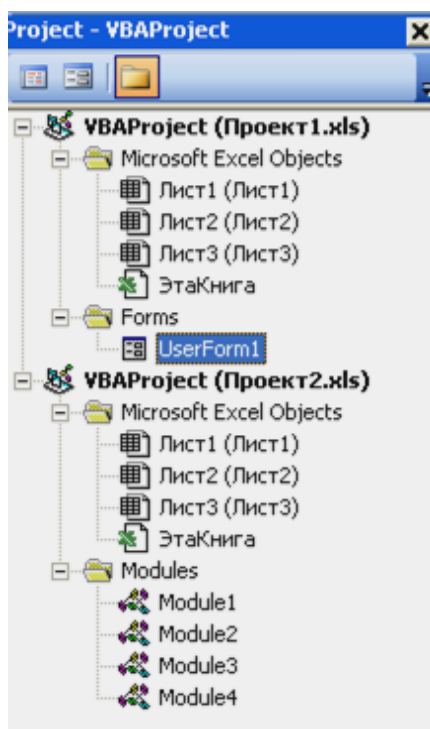


Рисунок 2 – Окно проекта Project-VBAProject

В проекте автоматически создаются модули для каждого рабочего листа и для всей книги. Модуль – это лист (не путать с рабочим листом), в котором набирается код. Кроме того, модули формируются для каждой пользовательской формы, макросов и классов.

По своему предназначению модули делятся на два типа: стандартные модули и модули объектов

К стандартным модулям относятся записываемые макросы. Такие модули добавляются в проект выбором команды Insert→Module (Вставка→Модуль).

К модулям объектов относятся модули рабочих книг, рабочих листов, форм и классов.

Двойным щелчком на значке модуля в окне проекта Project-VBAProject можно открыть соответствующий модуль. Значок активного модуля в окне проекта Project-VBAProject выделяется серым цветом.

Формы создаются выбором команды Insert→UserForm (Вставка→Пользовательская форма), а модули класса - Insert→Class Module



(Вставка→Модуль класса). По мере создания, добавления и удаления файлов из проекта эти изменения отображаются в окне проекта.

Удаление файла из окна проекта производится выделением значка файла с последующим выбором команды File→Remove 'имя\_модуля' (Файл→Удалить), где 'имя\_модуля' – имя выбранного модуля.

В окне проекта выводится список проектов всех открытых рабочих книг. Это позволяет легко копировать формы, модули из одного проекта в другой при помощи простой буксировки значка файла (для копирования файла – при нажатой клавише <Ctrl>).

### 1.3.2 Окно редактирования кода

Окно редактирования кода служит для ввода и изменения кода процедур и функций приложения.

Окно редактора кода для соответствующего модуля открывается двойным щелчком на значке файла в окне проекта.

Открыть модуль в редакторе кода для соответствующего объекта (например, рабочего листа) можно с помощью выделения значка этого объекта в окне проекта с последующим выбором команды View→Code (Вид→Код), представленного на рисунке 3. Код внутри модуля организован в отдельные разделы для каждого объекта, программируемого в модуле.

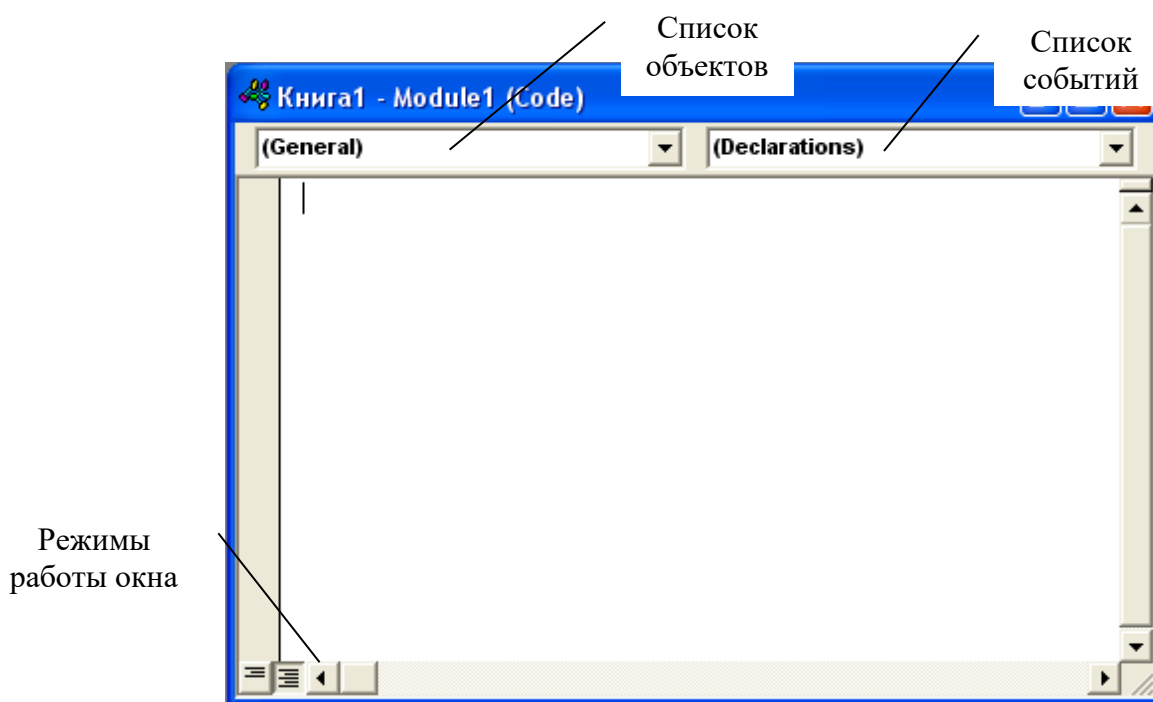


Рисунок 3 – Окно редактора кода

В верхней части окна кода помещены два раскрывающихся списка, облегчающих ориентацию в процедурах. Левый раскрывающийся список <Object> позволяет выбрать управляющий объект (элемент или форму), а правый

<Procedure> – событие, допустимое для указанного в левом списке объекта, показанного на рисунке 4.

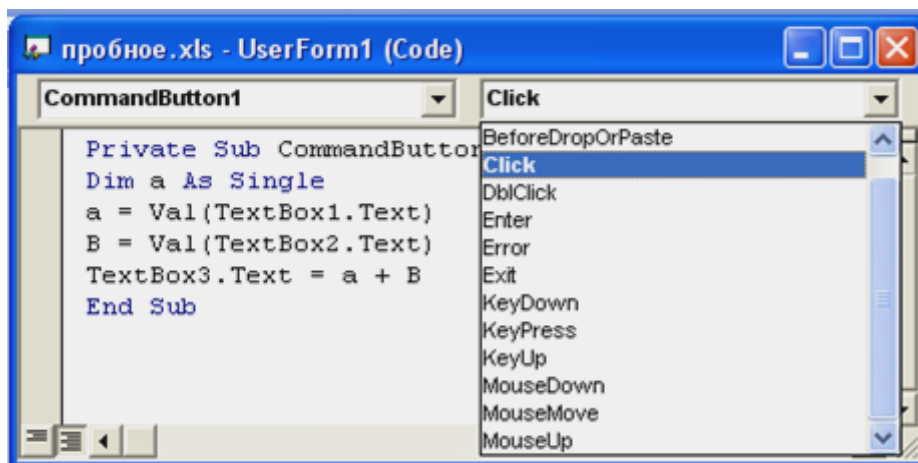


Рисунок 4 – Список компонентов правого раскрывающегося списка

В окне редактирования кода доступны два режима представления кода: просмотр кода отдельной процедуры и просмотр всего модуля. Переключение между режимами работы окна редактирования кода осуществляется следующими способами:

1) выбрать одну из двух кнопок в нижнем левом углу окна редактирования кода (левой <Procedure View> – отдельная процедура, правой <Full Module View> – все процедуры модуля);

2) установить или снять флажок Default to Full Module View (Просмотр всего модуля) вкладки Editor (Редактор) диалогового окна Options (Параметры), отображаемого на экране командой Tools→Options (Сервис→Параметры).

Для автоматического отображения списка компонентов, сведений о процедурах, функциях, свойствах и методах, а также для проверки синтаксиса нужно установить флажки, показанными на рисунке 5:

Auto List Members (Список компонентов), Auto Quick Info (Краткие сведения), Auto Syntax Check (Проверка синтаксиса) в диалоговом окне Options (Параметры)→вкладка Editor (Редактор).

Если установлен режим просмотра всех процедур модуля, то процедуры можно отображать с разделителями (горизонтальной чертой, разделяющей две соседние процедуры) или без них. Отображение или скрытие разделителей управляет флажок Procedure Separator (Разделитель процедур) диалогового окна Options (Параметры).

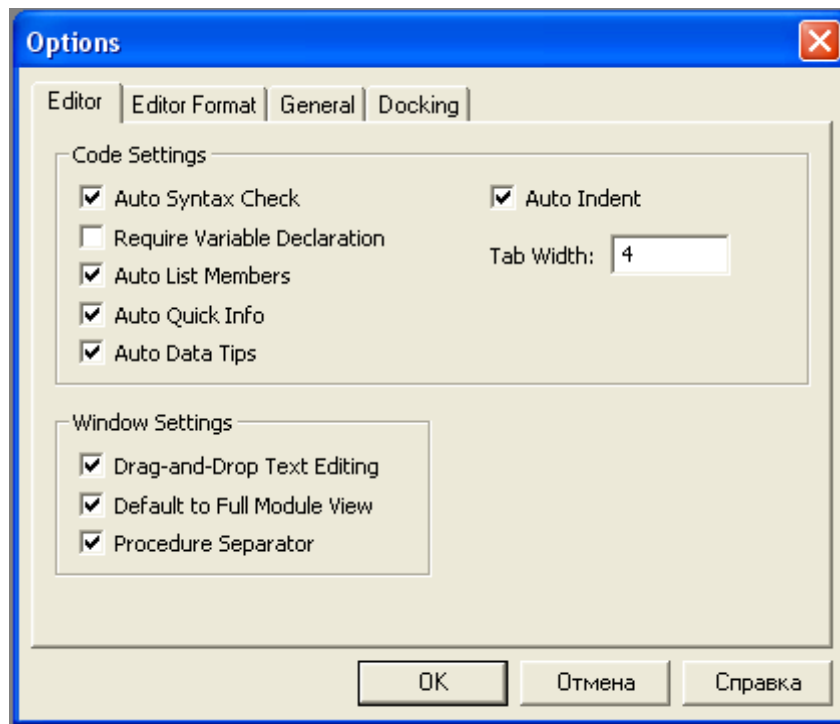


Рисунок 5 – Окно Options (Параметры)

### *Интеллектуальные возможности редактора кода*

Написание программ существенно облегчается за счет способности редактора кода автоматически завершать написание операторов, свойств и параметров. При вводе кода редактор сам предлагает пользователю список компонентов, логически завершающих вводимую пользователем инструкцию.

Например, набирая код

Range ("A1").

после ввода точки на экране отобразится список компонентов на рисунке 6, которые логически завершают данную инструкцию. Двойной щелчок на выбранном элементе из этого списка или нажатие клавиши <Tab> вставляет выбранное имя в код программы.

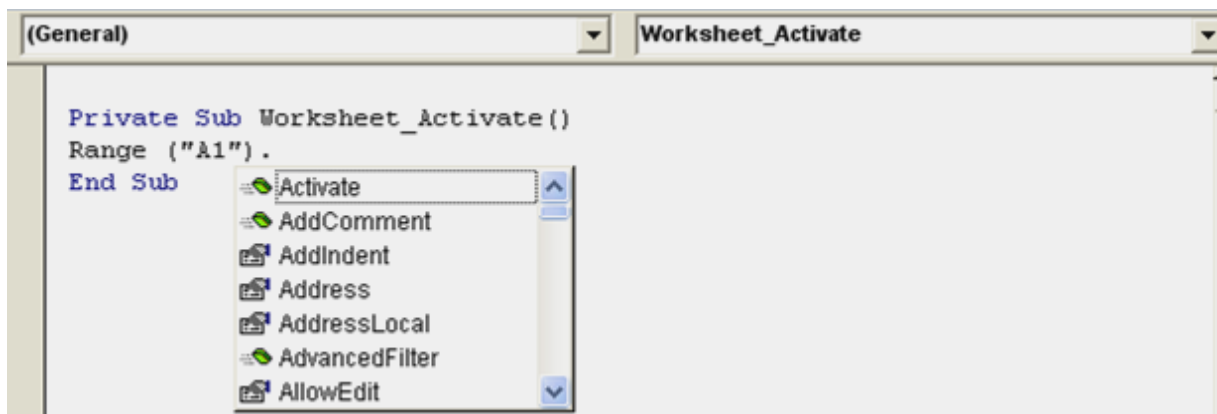


Рисунок 6 – Список компонентов

Автоматическое отображение списка компонентов происходит только при установленном флажке Auto List Members вкладки Editor диалогового окна Options, отображаемого на экране выбором команды Tools→Options.

Список компонентов отображается только для существующих в форме или на рабочем листе элементов управлений. Поэтому, если в проект входят элементы управления, сначала нужно создать их, а потом набирать код.

К интеллектуальным ресурсам редактора кода относится также автоматическое отображение на экране сведений о процедурах, функциях, свойствах и методах после набора их имени, показанным на рисунке 7.

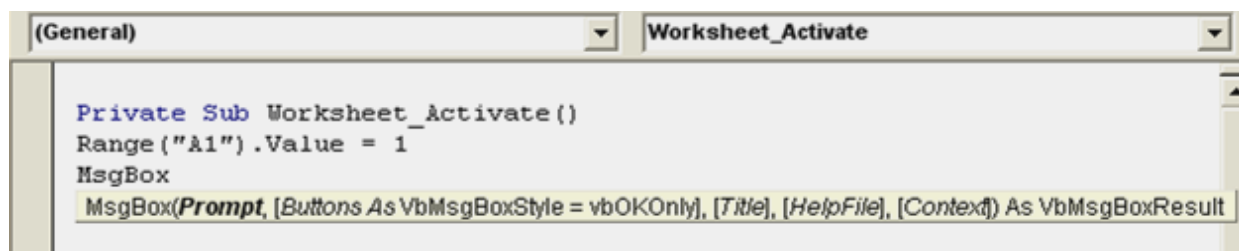


Рисунок 7 – Отображаемые сведения о вводимой процедуре

Автоматическое отображение на экране сведений о процедурах, функциях, свойствах и методах после ввода их имени происходит только при установленном флажке Auto Data Tips вкладки Editor диалогового окна Options, отображаемого на экране выбором команды Tools→Options.

В процессе ввода исходного кода VBA автоматически отслеживает опечатки. Всякий раз, когда нажимается клавиша <Enter>, VBA просматривает введенную строку, переводит в верхний регистр начальные буквы зарезервированных слов, а остальные их буквы – в нижний регистр (если, все было введено корректно). Кроме того, зарезервированные слова редактор VBA отображает синим цветом, добавляет пробелы вокруг знаков арифметических операций, а также операций сравнения. Если после нажатия клавиши <Enter> VBA ничего не отредактировал, то, скорее всего, он чего-то не понял в этой строке и поэтому придется искать ошибку самостоятельно.

Редактор кода также производит автоматическую проверку синтаксиса набранной строки кода сразу после нажатия клавиши <Enter>. Если после набора и нажатия клавиши <Enter> строка выделяется красным цветом, то это как раз и указывает на наличие синтаксической ошибки во введенной строке. Эту ошибку необходимо найти и исправить. Кроме того, если установлен флажок Auto Syntax Check вкладки Editor диалогового окна Options, помимо выделения красным цветом фрагмента кода с синтаксической ошибкой, на экране отображается диалоговое окно, поясняющее, какая именно ошибка возникла.

### 1.3.3 Окно редактирования форм

Редактор форм является основным инструментом визуального программирования. VBA представляет разработчику проектов не только

возможность использования уже существующих встроенных диалоговых окон, но и средства для разработки своих собственных форм. Пользовательские окна обеспечивают уникальный интерфейс, наилучшим образом приспособленный для решения конкретных задач, стоящих перед пользователем.

По своей сути форма (или пользовательская форма) представляет собой диалоговое окно, в котором можно размещать элементы управления (меню, кнопки, окна списков, полосы прокрутки и др. элементы). Формы позволяют реализовать интерфейс, к которому пользователь привык, работая с программными продуктами операционной системы Windows.

С точки зрения VBA форма представляет собой объект UserForm, со стандартной сеткой (группа регулярно расположенных точек). Сетка служит для удобства размещения элементов пользовательского интерфейса. Шаг сетки можно изменить, выполнив команду Tools→Options (Сервис→Параметры) (вкладка General (Общие)).

Как и любой объект, форма имеет свои свойства, методы и события.

Для того чтобы добавить форму в проект необходимо выполнить последовательность команд:

- 1) перейти в редактор Visual Basic;
- 2) выбрать команду Insert→UserForm.

В проекте появится пустое окно редактирования форм и Панель элементов управления, изображенных на рисунке 8.

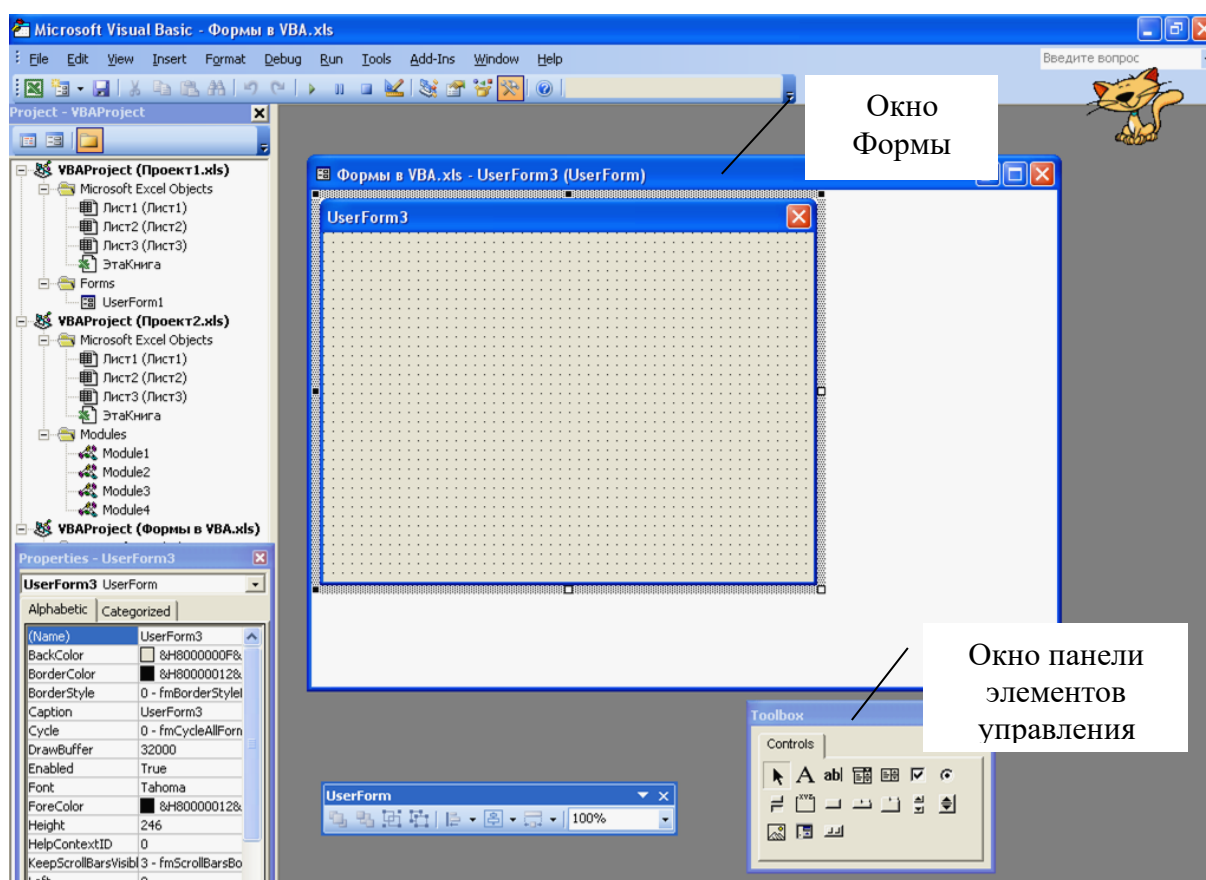



Рисунок 8 – Окно Формы

### 1.3.4 Окно свойств “Properties”

Каждый объект (форма или элемент управления) имеет набор свойств. Они определяют внешний вид формы или элемента управления и его поведение. Окно свойств, изображенное на рисунке 9, позволяет просматривать свойства и изменять характеристики (установки) объектов. Оно содержит список всех объектов, использующихся в конкретном пользовательском интерфейсе, и предназначено для установки свойств каждого объекта.

Для того чтобы просмотреть свойства выбранного объекта, надо выполнить следующее:

- выполнить команду меню View→Properties Window (Вид→Окно свойств);

- щелкнуть по кнопке  Properties Window в панели инструментов Standard. Окно свойств состоит из двух частей: верхней и рабочей.

В верхней части расположен раскрывающийся список, пользуясь которым можно выбрать элемент управления текущей формы или саму форму.

Рабочая часть имеет две вкладки: Alphabetic (по алфавиту) и Categorized (по категориям), отображающие набор свойств выбранного объекта по алфавиту или по категориям.

Значения свойств можно изменять либо путем их ввода с клавиатуры, либо используя раскрывающиеся списки.

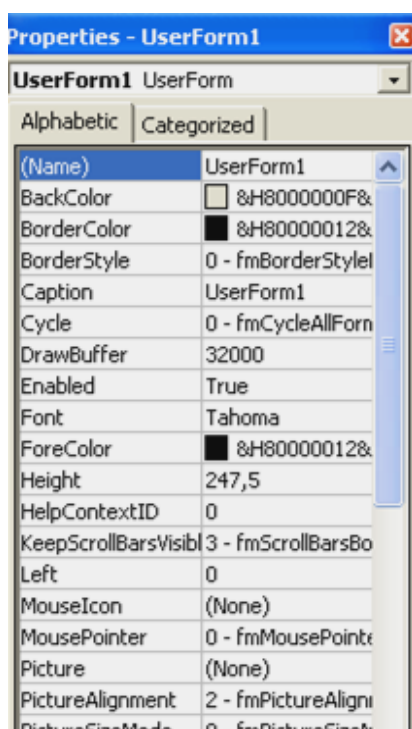


Рисунок 9 – Окно Свойства

### 1.3.5 Окно панели элементов управления (Toolbox)

Инструменты и средства управления на панели элементов управления служат для того, чтобы создавать новые элементы пользовательского интерфейса. Любое окно этого интерфейса содержит различные объекты: командные кнопки, текстовые окна, ярлыки (этикетки), переключатели и т.д. Все они называются Controls (элементами управления). Эти элементы добавляются в формы путем перетаскивания их с панели элементов управления в поле формы, указанных на рисунке 10.



Рисунок 10 – Окно инструментов

После того, как элементы управления внесены в форму, они становятся объектами или программируемыми элементами пользовательского интерфейса.

Имеется возможность расширения средств управления, представленных на панели управления. Для этого нужно выбрать команду Tools→Additional Controls (Сервис→Дополнительные элементы) и в появившемся диалоговом окне выбрать нужное средство управления.

Процесс выравнивания элементов управления в форме можно облегчить, используя команды меню Format (Формат).

## 2 Разработка программного кода

### 2.1 Структура проекта VBA

Цель: Научить студентов разрабатывать в проекте VBA программный код с помощью пользовательских подпрограмм-функций и подпрограмм-процедур.

Проект представляет собой реестр файлов, которые необходимы для генерации исполняемого файла приложения. Реестр содержится в файле проекта с расширением \*.xls.

Проект состоит из иерархической структуры файлов форм и модулей.

Формы служат для создания диалоговых окон приложений VBA. Редактор форм является основным инструментом визуального программирования.

Форма – это основа графического интерфейса Windows-приложения.

Модуль – это файл, содержащий программный код, не связанный ни с одной формой или элементом управления.



Модули содержат только программные инструкции, отображенные в окне «Module1(Code)». В них декларируются переменные, константы, размещаются коды пользовательских функции и процедур.

Модули создаются для каждого рабочего листа, для всей книги, для каждой пользовательской формы, класса, макроса. Различают стандартные модули и модули объектов.

Стандартные модули содержат макросы. Они добавляются в проект командой меню Вставка/Модуль.

К модулям объектов относятся модули рабочих книг, листов, форм и классов. Формы добавляются в проект командой меню Вставка/UserForm, а модули классов – командой Вставка/Модуль классов.

## 2.2 Пользовательские подпрограммы-функции

Функция – это специальным образом оформленный блок программного кода, который выполняет действия в пределах своего блока и возвращает значение.

Пользовательские функции строятся в стандартном модуле редактора VBA.

Процедура типа **Function** представляет собой группу операторов, содержащихся между оператором **Function** и оператором **End Function** в стандартном модуле. Операторы в функции обрабатывают текст, управляют входными данными или вычисляют значения. Функция вызывается в программу помещением имени функции в программный оператор с указанием требуемых аргументов (данных, обрабатываемых функцией).

Структура кода функции пользователя.

Функция состоит из трех частей: заголовка, тела функции, конца функции.

1 **Заголовок** содержит:

- зарезервированное слово **Function**;
- идентификатор (имя) функции,
- заключенный в круглые скобки список аргументов и тип возвращаемого функцией значения.

2 **Тело функции** состоит из группы операторов

3 **Конец функции** представлен оператором **End Function**

Синтаксис:

```
[Private | Public] Function ИмяФункции (СписокПараметров) As Тип  
    Тело функции (строки кода)  
    ИмяФункции = ВозвращаемоеЗначение  
End Function
```

где

[Private | Public] – необязательные ключевые слова, определяющие область видимости функции;

Function – ключевое слово, указывающее на то, что это функция;

ИмяФункции – имя функции (задается самим пользователем);

СписокПараметров – список формальных параметров, который может отсутствовать;

As – (как) ключевое слово, предваряющее значение типа



данных;

Тип – тип данных возвращаемого значения;

ВозвращаемоеЗначение – значение, возвращаемое функцией;

End Function – ключевые слова, указывающие на окончание блока функции.

Особенность:

1) функция вычисляет единственное результирующее значение простого типа;

2) носителем возвращаемого значения является ее имя, т.е. идентификатор Имя\_функции. Поэтому в теле функции должен присутствовать хотя бы один оператор присваивания, в левой части которого стоит имя описываемой функции, причем хотя бы один оператор такого вида должен быть выполнен. Этот оператор присваивания и определяет значение, вычисляемое с помощью функции.

Пример 1.

Постановка задачи. Построить функцию пользователя, вычисляющую значение функции в заданных точках.

$$z = \ln \left| \lg x - \sqrt{|\cos x - e^x|} \cdot \arcsin \left| \operatorname{tg} \frac{|ax - b|}{\sin|x|} + b \right| \right|, \text{ где } a = 0.126, b = 0.842, x = 0.34$$

Технология выполнения задания:

1 Исходные данные:

$a, b, x \in \mathbb{R}$

Результат:  $z \in \mathbb{R}$ .

2 Набрать в стандартном модуле проекта следующую пользовательскую функцию:

Function z(a, b, x)

z1 = Abs(Log(x) / Log(10)) - Sqr(Abs(Cos(x) - Exp(x)))

z2 = Abs(Tan(Abs(a \* x - b)) / Sin(Abs(x)) + b)

z3 = Atn(z2 / Sqr(Abs(1 - z2 ^ 2)))

z = Log(Abs(z1 \* z3))

End Function

По умолчанию она попадает в раздел Определенные пользователем списка Категория окна Мастер функций.

3 Вычислить:

- перейти из окна VBA в окно MS Excel;

- выбрать ячейку A1 и ввести число 0,126;

- выбрать ячейку A2 и ввести число 0,842;

- выбрать ячейку A3 и ввести число 0,34;

- выделить ячейку A4, в которой найти значение функции;

- выбрать команду Вставка→Функция;

Определенные пользователем;

- в списке Функция выбрать функцию z и нажать кнопку ОК;

- во втором окне Мастера функций в поле А ввести ссылку на ячейку А1, в поле В ввести ссылку на ячейку А2, в поле Х ввести ссылку на ячейку А3 и нажать кнопку ОК.

Пример 2.

Поставка задачи. Построить функцию, которая возвращает стоимость товара по его стоимости без НДС и значению ставки НДС.

Технология выполнения задания:

1 Исходные данные:

СтоимостьБезНДС, НДС  $\in \mathbb{Z}$

Стоимость  $\in \mathbb{Z}$ .

2 Набрать в стандартном модуле проекта следующую пользовательскую функцию:

Function Стоимость (СтоимостьБезНДС, НДС)

Стоимость = СтоимостьБезНДС \* (1 + НДС / 100)

End Function

3 Вычислить, например, стоимость товара с учетом НДС при условии, что НДС равен 25%, а стоимость товара без учета НДС равна 1000. Для этого:

- выбрать ячейку А2 и ввести число 1000 (стоимость без НДС);

- выбрать ячейку В2 и ввести число 25 (НДС);

- выделить ячейку С2, в которой найдем значение функции;

- выбрать команду Вставка→Функция;

Определенные пользователем

- в списке Функция выбрать функцию стоимость и нажать кнопку ОК.

4 Во втором окне Мастера функций в поле СтоимостьБезНДС ввести ссылку на ячейку А2, а в поле НДС – ссылку на ячейку В2 и нажать кнопку ОК.

## 2.3 Пользовательские подпрограммы-процедуры

Зарезервированное слово Sub происходит от слова subroutine (подпрограмма), которое является синонимом термина процедура. Процедура представляет собой именованную часть кода, выполняющую определенные действия: вывод текста, выполнение арифметических действий, проигрывание видеофрагмента. Процедура может иметь параметры и в результате выполнения последовательности инструкций изменять их значения.

Синтаксис:

**[Private | Public] Sub** ИмяПроцедуры (СписокПараметров) As Тип

Тело процедуры (строки кода)

**End Sub**

## 2.4 Данные и их описание

### 2.4.1 Алфавит и лексемы языка

Алфавит VBA включает:

- прописные и строчные буквы латинского алфавита: A-Z, a-z;

- прописные и строчные буквы кириллицы: А-Я, а-я;
- цифры от 0-9;
- символ подчеркивания «\_»;
- неизобразяемые символы, используемые для отделения лексем друг от друга (пробел, табуляция, переход на новую строку);
- специальные символы, участвующие в построении конструкций языка:
 
$$+,-,*,./,\backslash,\wedge,=,>,<,[,],(,),. ,: ,\{,\},',@,&;$$
- составные символы, воспринимаемые как один символ: <=, >=, <>;
- комментарии, используемые для читаемости и понятности текста программы. Представляют собой последовательность любых символов, которая начинается со знака «'» (апостроф) или с ключевого слова Rem.

Пример.

' Это комментарий

Rem Это тоже комментарий

Программный код VBA представляет собой последовательность лексических единиц (лексем), записанных в соответствии с принятыми синтаксическими правилами.

Лексема – это минимальная неделимая единица текста программы, которая имеет определенный смысл для компилятора.

Различают шесть классов лексем:

- свободно выбираемые и используемые идентификаторы;
- служебные (зарезервированные) слова;
- константы;
- строки (строковые константы);
- операции (знаки операций);
- разделители (знаки пунктуации).

## 2.4.2 Переменные и константы

Переменная – это объект, занимаемый именованную область оперативной памяти. Содержимое этого объекта может изменяться в ходе выполнения программы.

Константа – это объект, значение которого фиксировано и не может быть изменено во время выполнения программы.

Идентификатор - это имя переменной, константы, процедуры, функции, и других объектов.

Правила написания идентификаторов:

1) в программе он должен быть уникальным, не должен совпадать с зарезервированными словами VBA;

2) он должен начинаться только с буквы;

3) он может состоять из букв, цифр и знака подчеркивания. Не допустимы - пробелы, точки, разделительных символов, знаков операций и др. спецсимволов. Прописные и строчные буквы различаются;

4) длина идентификатора может включать до 255 символов, но VBA учитывает только первый 31 символ;

5) они могут быть простыми и составными.

Пример,

N, I, Number, Номер – простые имена переменных;

ФамилияСтудента – составное имя переменной.

Соглашения по стилю идентификаторов:

- лучше использовать имена из строчных букв, в случае составных названий нужно отделять друг от друга составляющие их слова подчеркиванием или начинать новое слово с прописной буквы;
- имена из прописных букв используются для определения констант;
- название идентификатора нужно начинать со специального знака (приставки или префикса), который указывает на тип данных, связанный с этим идентификатором.

### 2.4.3 Объявление переменных

Переменные – это объекты, предназначенные для хранения данных. В разные моменты времени переменные могут хранить различные значения. Имена переменных позволяют их различать в программе, осуществлять доступ к различным участкам памяти для записи данных и их извлечения.

Основной принцип VBA.

Каждый именованный объект (переменная, константа), перед своим использованием должен быть предварительно объявлен (декларирован).

Декларация переменных:

При объявлении переменной надо указать, что объявляется переменная:

- задать имя переменной
- указать ее тип.

Тип указывает способ представления переменной. В переменных можно хранить любые типы данных: число, строку текста, экземпляр объекта, элементы управления, базы данных.

Синтаксис:

Static | Public | Private | Dim <имя Переменной1> [As <тип данных1>], [<имя Переменной2>[As <тип данных2>]]...

где

Dim (dimension – размер) – ключевое слово, которое сообщает VBA, что декларируется переменная и резервируется область памяти для хранения;

Имя переменной – имя переменной (идентификатор, не входящий в перечень ключевых слов VBA);

As (как) – ключевое слово, которое сообщает VBA, что определяется тип данных для переменной;

Тип – тип данных для объявляемой переменной;

Private (Частный), Public (Общий) – ключевые слова, определяющие область видимости переменной;

Static (Статический) – ключевое слово, которое определяет, сохраняет ли переменная свое значение при завершении блока программы (процедуры, функции) и выходе из него.

Пример 1.

```
Dim intName As Integer
```

Если Тип переменной не объявлен, по умолчанию переменная получает тип Variant.

Примечание. Распространенная ошибка в VBA заключается в использовании следующей инструкции:

```
Dim a, b As Long
```

Переменная b имеет тип Long, а переменная a – тип Variant.

## 2.4.4 Объявление констант

Константы могут быть неименованными и именованными.

Неименованные константы: символьные, целые и вещественные.

1 Символьная константа служит для изображения отдельных знаков и представляет собой лексему, состоящую из символов, заключенных в кавычки.

Пример 1.

“P”, “Program”, “3.14”, “+”

2 Целые константы

Пример 1.

-3, -485, 44, 0

3 Вещественные константы

Представляются в памяти ЭВМ в двух формах:

- в форме с десятичной точкой (фиксированной): 0.1, -0.125;
- в форме с плавающей точкой вида <мантисса>E<порядок>: 7.32E+00 (=7.32), 1.5E-02 (=0.015).

Встроенные константы (предопределенные).

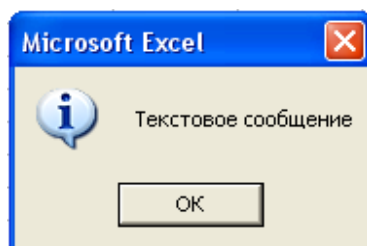
Такие константы используются при работе с объектами приложения. Эти константы не требуют предварительного описания.

Встроенные в VBA константы имеют префикс VB. Для того чтобы узнать конкретное значение константы, можно воспользоваться окном “Object Browser” (Обозреватель объектов).

Пример 2.

```
MsgBox "Текстовое сообщение", vbInformation
```

(предопределенная константа vbInformation указывает, что в окне сообщения должен быть значок «Информация»).



Константы, создаваемые пользователем

Синтаксис:

[Public/Private] Const <имя\_константы> [As <тип данных>] = <значение константы>,

где

<значение константы> - любое значение или формула, возвращающая значение, которое должно использоваться в качестве константы.

Имя константы и тип данных устанавливаются так же, как и для переменной.

Пример 3.

Const PI As Single = 3.14 ‘Объявлена именованная числовая константа для хранения значения числа Pi

Const Firm = “ООО Бахус”

## 2.4.5 Строковые переменные

Различают строки переменной и фиксированной длины.

Строки переменной длины могут содержать до двух миллиардов символов.

Строки фиксированной длины – это строка постоянного размера, указанного при объявлении переменной. Если такой строке присваивается значение более длинное, то лишние символы заполняются пробелами.

Синтаксис:

Dim VarName As String \* ДлинаСтроки

где

ДлинаСтроки – целочисленная переменная или константа, содержащая число, которое указывает длину строковой переменной.

Пример 1.

Dim strMyName As String \* 20 (объявляется строковая переменная фиксированной длины в 20 символов)

В VBA имеется строковая операция – конкатенация, которая применяется для объединения нескольких строк в одну, и обозначается символом амперсанда &. Этот символ должен слева и справа обрамляться пробелами.

Пример 2.

strMy = “Visual” & “Basic for Applications”

## 2.4.6 Области видимости переменных и констант

Переменная или константа может существовать и быть доступной для некоторых частей программы и при этом быть недоступной (невидимой) для других частей программы.

Область программы, в которой может быть использована переменная или константа, называется областью видимости **переменной**.

Если переменная видима, то она доступна и, следовательно, существует.

Переменные или константы можно объявлять в трех местах:

- внутри процедуры (видимы только внутри процедуры);
- в верхней части модуля, которая называется областью общих объявлений (видимы всем процедурам данного модуля),

- глобальный уровень видимости (видимы в любой процедуре любого модуля приложения). Объявляются в верхней части модуля в области общих объявлений с помощью оператора **Public**.

**Локальные объекты (местные)**- это все объекты (константы, переменные), которые описываются после заголовка процедуры или функции, и доступны только в их пределах. Они создаются при входе в подпрограмму и уничтожаются при выходе из нее.

**Глобальные объекты** – это все объекты, описанные в вызывающей программе, которые являются доступными внутри всех процедур и функций.

Операторы объявления:

**Dim** – объявляет локальные переменные только внутри процедуры или функции, в которых они объявлены. Они сохраняют свои значения, только пока выполняется процедура. При завершении процедуры значения этих переменных теряются.

**Static** – аналогичный оператору **Dim**. Отличие: переменные сохраняют свое значение после выхода из процедуры, пока работает программа.

**Dim, Private** – объявляет переменные на уровне модуля (формы), в котором они описаны, но для других модулей данного проекта не доступны. Объявляются в области описания модуля (перед описанием процедур или функций).

**Public** – объявляет глобальную переменную на уровне проекта, которая доступна для всех его модулей.

## 2.4.7 Типы данных

Типы данных относятся к самым фундаментальным понятиям любого языка. Тип данных определяет множество допустимых значений, которое может принимать указанная переменная.

Переменную можно представить как простейший объект программы следующим образом:

Переменная		
Логический уровень	Имя переменной (идентификатор)	Тип переменной
Физический уровень	Область памяти	Формат представления в памяти

Имя переменной представляет логический уровень. Оно однозначно связывает переменную с некоторой физической областью памяти. Имена переменных позволяют их различать в программе, осуществлять доступ к различным участкам памяти для записи данных и их извлечения.

Для эффективного использования памяти необходимо правильно выбрать тип переменной. В таблице 1 приведены базовые типы переменных VBA, необходимая для их размещения память и диапазон возможных значений.

Таблица 1 - Основные типы данных

Тип	Занимаемая память	Интервалы значений
Целочисленные типы		
Byte (целые числа)	1 байт	0...255
Boolean (логические значения)	2 байта	True...False
Integer (целые числа)	2 байта	-32768...32767
Long Integer (длинные целые числа)	4 байта	+/-2.1E9
Вещественные типы (типы с плавающей точкой)		
Single (одинарной точности с плавающей точкой)	4 байта	-3.402823E38...-1.401298E-45 для отрицательных чисел 1.401298E-45...3.402823E38 для положительных чисел
Double (двойной точности с плавающей точкой)	8 байт	-1.7976313486232E308...- 4.94065645841247E-324 для отрицательных чисел 4.94065645841247E324...1.7976313486232E 308 для положительных чисел
Строковые типы		
String (текстовая информация - строка фиксированной длины)	1 байт на каждый символ	От 1 до 65400
String (текстовая информация - строка переменной длины)	10 байт + 1 байт на каждый символ	От 0 до двух миллиардов символов
Объектные типы		
Object (Рисунок или ссылка на любой другой объект)	4 байта	Ссылка на объект
Типы Variant		
Variant значения любого из перечисленных типов данных	16 байт для чисел, 22 байта + 1 байт на	Любое числовое или строковое значение



	каждый символ для строк	
--	-------------------------------	--

## 2.5 Функции в VBA

В VBA используются следующие виды функций:

- математические встроенные функции;
- математические функции, не представленные в VBA;
- функции форматирования данных;
- функции преобразования типов

### 2.5.1 Математические встроенные функции

Таблица 2 - Встроенные математические функции VBA

Функция	Возвращаемое значение
Abs (x)	$ x $ - абсолютная величина числа
Atn (x)	arctg (x) – арктангенс от значения параметра, заданного в радианах
Sin (x)	sin (x) – возвращает синус угла от значения параметра, заданного в радианах
Cos (x)	cos (x) – косинус указанного в радианах угла
Tan (x)	tg (x) – возвращает тангенс угла от значения параметра, заданного в радианах
Exp (x)	$e^x$ – возвращает число e, возведенное в указанную степень, где e – основание натурального логарифма
Log (x)	ln (x) – возвращает натуральный логарифм от значения числового выражения
Sqr (x)	$\sqrt{x}$ - возвращает квадратный корень числового выражения
Rnd (x)	Случайное число из интервала [0,1). Перед вызовом функции надо использовать оператор Randomize (рандомизации) – запуск генератора псевдослучайных чисел)
Sgn (x)	Возвращает +1, если значение параметра положительно, -1, если отрицательное, 0, если 0
Fix (x)	Возвращает результат округления выражения с плавающей точкой до целой части. В случае отрицательного параметра возвращает ближайшее большее отрицательное число
Int (x)	Возвращает результат округления выражения с плавающей точкой до целой части. В случае отрицательного параметра возвращает ближайшее меньшее отрицательное число

## 2.5.2 Математические функции, не представленные в VBA

Таблица 3 - Математические функции, не представленные в VBA

Функция	Возвращаемое значение
Log(X)/Log(10)	Ig(x) – возвращает десятичный логарифм от значения числового выражения
$\text{Atn} \left( \frac{X}{\sqrt{1-X^2}} \right)$	arcsin (x) – возвращает арксинус угла от значения параметра, заданного в радианах
$\text{Atn} \left( \frac{\sqrt{1-X^2}}{X} \right)$	arccos (x) – возвращает арккосинус угла от значения параметра, заданного в радианах
Cos (x)/Sin (x)	ctg (x) - возвращает котангенс угла от значения параметра, заданного в радианах
Значение числа $\pi$	$\text{Pi} = 4 * \text{Atn} (1)$

## 2.5.3 Функция форматирования данных

Для того чтобы представить выражение отформатированным в специфицированном формате, необходимо воспользоваться функцией Format. Она возвращает значение типа Variant (String), содержащее выражение, отформатированное согласно указанным спецификациям.

Синтаксис:

Format (выражение [ , “Имя формата (или символ формата)“])

Именованные числовые форматы, показанные в таблице 4.

Таблица 4 - Именованные числовые форматы

Имя формата	Описание
General Number	Число без разделителя тысяч
Currency	Отображает две цифры справа от десятичной точки
Fixed	Отображает одну цифру слева и две справа от десятичной точки
Standard	Отображает одну цифру слева и две справа от десятичной точки и выводит разделитель тысяч
Percent	Отображает число в виде процентов и выводит две цифры справа от десятичной точки
Scientific	Использует формат с плавающей десятичной точкой
Yes/No	Отображает No, если число равно 0, и Yes – в противном случае
True/False	Отображает False, если число равно 0, и True – в противном случае
On/Off	Отображает Off, если число равно 0, и On – в противном случае

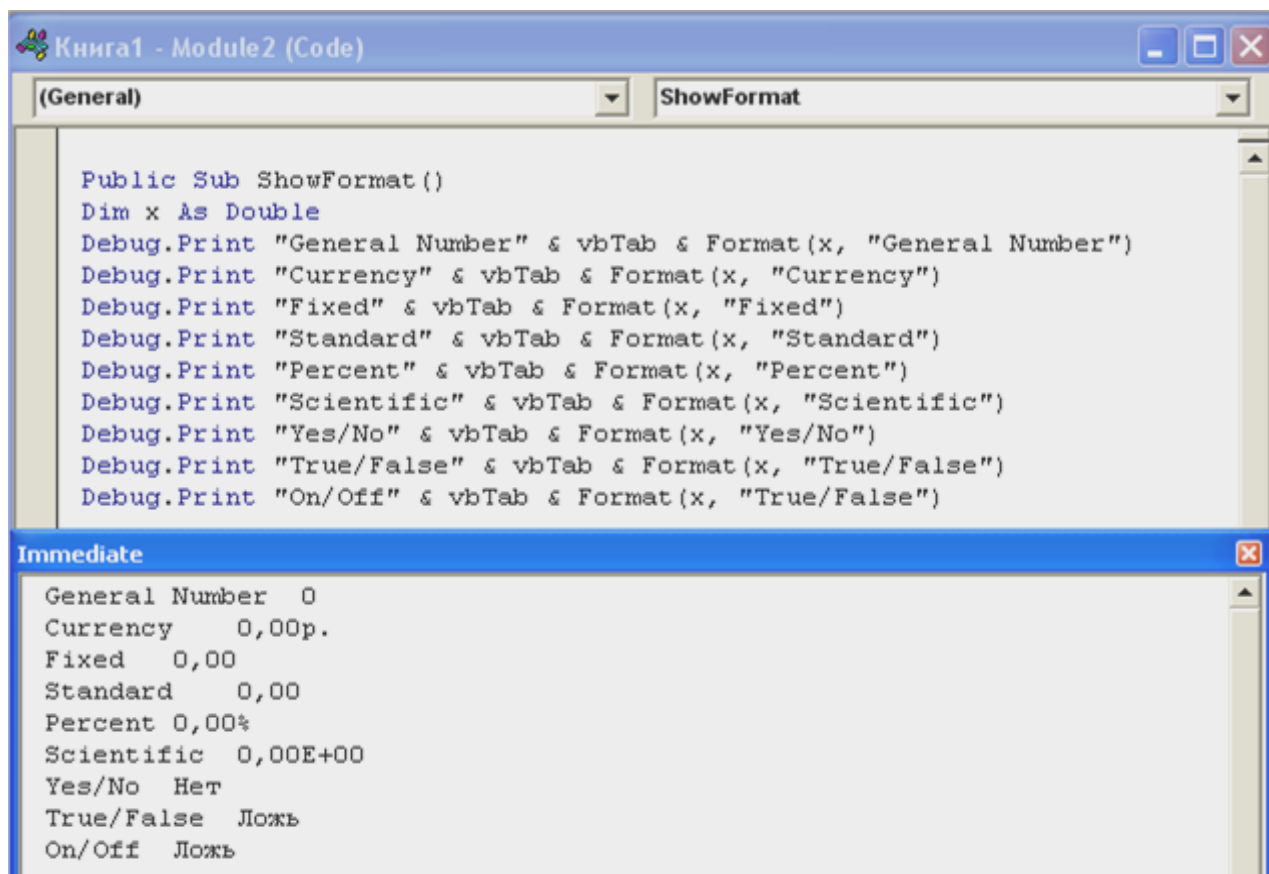


Рисунок 11 - Программа, показывающая работу различных именованных форматов с помощью окна Immediate

## 2 Пользовательские числовые форматы

При построении пользовательского формата возможно использование символов, указанных в таблице 5.

Таблица 5 - Символы пользовательского формата

Символ	Описание
1	2
0	Резервирует позицию цифрового разряда. Отображает цифру или ноль. Если у числа, представленного параметром, есть какая-нибудь цифра в той позиции разряда, в которой в строке формата находится 0, функция отображает эту цифру параметра, если нет – в этой позиции отображается ноль. Пример. <code>Debug.Print Format (1.2 ^2, “##.000”)</code>
#	Резервирует позицию цифрового разряда. Отображает цифру или ничего. Если у числа, представленного параметром, есть какая-нибудь цифра в той позиции разряда, в которой в строке формата находится #, функция отображает эту цифру параметра, если нет – в этой позиции ничего не отображается. Пример. <code>Debug.Print Format (1.2 ^2, “##.###”)</code>

Продолжение таблицы 5

1	2
. (точка)	Резервирует позицию десятичного разделителя. Указание точки в строке формата определяет, сколько разрядов необходимо отображать слева и справа от десятичной точки.
E+,E-, e+, e-	Разделитель мантиссы и порядка в экспоненциальном формате. Пример. Debug.Print Format (sin(x) * exp(5), “#.###e+##”)
%	Резервирует процентное отображение числа Пример. Debug.Print Format (0.5, «#.»») (отобразится 50 %)
d,m,y	Резервирует позицию при выводе дня, месяца, года в категории форматов Data Пример. Debug.Print Format (Now, “dd/mm/yyyy”) (отобразится текущая дата, например 25.09.2005)
/	Разделитель дня, месяца, года в категории форматов Data

## 2.5.4 Функции преобразования типов

Преобразование строки в число и обратно осуществляют следующие две функции, представленные в таблице 6.

Таблица 6 - Функции преобразования типов

Функция	Описание
Val (String)	Возвращает число, содержащееся в строке string, как числовое значение (Val от англ. Value – значение, String – строка)
Str (number)	Возвращает значение типа Variant (String), являющееся строковым представлением числа number. В качестве допустимого десятичного разделителя функция Str воспринимает только точку.
CStr (Выражение)	Возвращает числовое выражение или строку в строку
CInt (Выражение)	Возвращает числовое выражение или строку в число типа Integer
CSng (Выражение)	Возвращает числовое значение или строку в число типа Single

## 2.6 Операторы, выражения и операции

Строка с кодом в исходном тексте программы VBA называется программным оператором.

Программный оператор – это неделимое предложение, выполняющее какое-либо действие. Он может состоять из любой комбинации ключевых слов VBA, свойств, функций, операций и символов, совокупность которых представляет собой корректную конструкцию, распознаваемую компилятором VBA. Например,  
Label1.Caption = Time

Правила, применяемые при построении программных операторов, называются синтаксисом.

Программный оператор может включать выражения.

Выражение (Expression) – это комбинация знаков операций и операндов, а также скобки. Назначение любого выражения – получение некоторого значения.

Синтаксическая конструкция выражения:

Операнд1 [операция Операнд2 [операция Выражение]]

В зависимости от типа формируемых значений определяются типы выражений. Например, если значениями выражения являются целые и вещественные числа, то говорят об арифметических выражениях.

Для формирования и последующего вычисления выражений служат операции. Для записи операций VBA имеет знаки операций, которые воспринимаются компилятором как отдельные лексемы. Последовательность вычислений в программном операторе определяется приоритетом операций.

Каждая операция имеет свой приоритет (ранг), как показано в таблице 7. Операции ранга 1 имеют наивысший приоритет и в программном операторе выполняется первым. Операции одного ранга в выражениях выполняются в соответствии с правилами ассоциативности (слева направо или наоборот).

Таблица 7 - Приоритеты операций

Приоритет	Операция
1	Выражение, заключенное в скобки ()
2	Вызов функции
3	Степень (^)
4	- (смена знака)
5	(*) и (/)
6	(\ ) деление нацело
7	Mod (остаток от деления нацело)
8	(+) и (-)
9	>, <, >=, <=, <>, =
10	Not
11	And
12	Or

Операнды связаны между собой знаками операций

В программах на VBA можно использовать стандартный набор операций над данными. Различают следующие типы операций:

- операция присваивания;
- математические, выполняются над числами и их результатом являются числа;
- отношения, применяются не только к числам, и их результатом являются логические значения;
- логические операции;
- строковые операции.

### 2.6.1 Операция присваивания

При объявлении переменной происходит связывание имени переменной с областью памяти, в которой будет храниться ее значение. Однако это значение после объявления может оказаться произвольным. Для того чтобы присвоить переменной нужное значение, используется операция присваивания.

Назначает результат вычисления выражения переменной, константе или свойству объекта. Операция присваивания всегда включает знак равенства =.

Синтаксис:

Имя\_Переменной = Выражение

где

Имя\_Переменной – имя переменной (идентификатор);

Символ «=» – знак операции присваивания;

выражение – значение (число), комбинация переменных, констант, функций, связанных знаками операций.

Работа операции присваивания.

Операция присваивания предписывает выполнить выражение, заданное в его правой части, и присвоить результат имени переменной, имя которой указано в левой части.

Пример 1.

$x = 2$

$x = x + 2$

(переменной  $x$  будет присвоено 4)

Для присваивания переменной ссылки на объект применяется инструкция

Set.

В общем случае инструкция Set имеет следующий синтаксис:

**Set objectvar = [New] objectexpression [Nothing]**

где

New – ключевое слово, которое используется при создании нового экземпляра объекта;

Nothing – позволяет освободить все системные ресурсы и ресурсы памяти, выделенные для объекта, на который имелась ссылка (т.е. она удаляет объект из памяти).

Пример 2.

Dim A as Object

Set A = cmdOK

MsgBox A.Caption

Инструкция Set присваивает переменной A элемент управления Кнопка с именем cmdOK и далее выводит в окне надпись отображаемую на поверхности кнопки.

### 2.6.2 Математические операции

Математические операции применяются для записи формул.

Формула представляет собой программный оператор, содержащий числа, переменные, операторы и ключевые слова или же комбинацию этих элементов и вычисляющий новое значение. Список математических операций VBA и их рангов приведен в таблице 8.

Операции сложения, вычитания, умножения и деления называют *основными математическими операциями* и пояснений не требуют.

Остальные математические операции называются *дополнительными*. Они применяются в специальных математических формулах и при обработке текстовой информации.

Таблица 8 - Математические операции

Операция	Математическое действие
[Операнд1] + [Операнд 2]	Сложение
[Операнд 1] – [Операнд 2]	Вычитание
-[ Операнд 1]	Изменение знака числа
[Операнд 1] * [Операнд 2]	Умножение
[Операнд 1] / [Операнд 2]	Деление
[Операнд 1] \ [Операнд 2]	Целочисленное деление
[Операнд 1] mod [Операнд 2]	Остаток от деления по модулю
[Операнд 1] ^ [Операнд 2]	Возведение в степень

Rezult = 10\3 – результат 3 (целая часть от деления);

Rezult = 10 mod 3 – результат 1 (остаток от деления);

Rezult = 9 ^ 0.5 – результат 3

Rezult = 2 ^ -2 – результат 0.25

Общие правила применения математических операций:

Синтаксис:

Rezult = Операнд1 Операция Операнд2 ... Операция ОперандN

где

Rezult – переменная, содержащая результат выполнения оператора;

Операнд1, Операнд2, ..., ОперандN – переменные, константы, числовые значения, функции.

### 2.6.3 Операции отношения

В отличие от математических операций, результатом выполнения которых может быть любое значение, операция отношения может иметь только два результирующих значения – True (Истина) и False (Ложь), которые могут быть присвоены переменным типа Boolean или определенному свойству объекта. Перечень операций отношения в VBA приведен в таблице 9.

Таблица 9 - Операции отношений в VBA

Операция	Описание
выражение1 > выражение2	Больше. Результат - True, если первый операнд больше второго
выражение1 >= выражение2	Больше или равно. Результат - True, если первый операнд больше или равен второму
выражение1 < выражение2	Меньше. Результат - True, если первый операнд меньше второго
выражение1 <= выражение2	Меньше или равно. Результат - True, если первый операнд меньше или равен второму
выражение1 <> выражение2	Не равно. Результат - True, если первый операнд не равен второму
выражение1 = выражение2	Равно. Результат - True, если первый операнд равен второму

Операции отношения применяются для записи выражений условия, результатом выполнения которых являются значения True (Истина) или False (Ложь). Примеры использования операций отношения приведены в таблице 10.

Таблица 10 - Операции отношения

Выражение условия	Результат
25 <> 30	True (25 не равно 30)
25 < 30	True (25 меньше 30)
25 > 30	False (25 не больше 30)
TextBox1.Text = «Петров»	True, если слово Петров является содержимым первого текстового поля, а в противном случае - False
Number >= 100	True, если переменная Number содержит значение не меньше 100, в противном случае False

#### 2.6.4 Логические операции

Логические операции применяются в логических выражениях. Если существует несколько условий выбора в операциях отношения, то эти операции связываются между собой логическими операциями. Логические операции VBA приведены в таблице 11.

Таблица 11 - Логические операции VBA

Операция	Описание
(Операнд1) AND (Операнд 2)	Логическое умножение. Возвращает True тогда, когда только оба операнда возвращают True



(Операнд 1) OR (Операнд 2)	Логическое сложение. Возвращает True тогда, когда хотя бы один операнд возвращает True
----------------------------	----------------------------------------------------------------------------------------

Продолжение таблицы 11

Not [Операнд]	Если условие имеет значение True, то результирующим значением будет False. Если условие имеет значение False, то результирующее значение будет True
[Операнд1] Xor [Операнд2]	Если только одно из связываемых условий имеет значение True, то результирующее значение также будет True. Если оба условия имеют одинаковые значения, то результирующее значение будет False

## 2.6.5 Строковые операции

В VBA есть только одна операция для работы со строками – это операция конкатенации. Конкатенация позволяет объединить значения двух или нескольких строковых переменных или строковых констант. Знаком операции конкатенации является символ амперсанд (&). При конкатенации строк значение второй строки добавляется в конец первой. Результатом операции является более длинная строка, составленная из исходных строк.

Синтаксис:

strВыражение\_1 & strВыражение\_2 [... & strВыражение\_N]

где

strВыражение – строковые выражения, которые могут быть любыми допустимыми строками (строковыми переменными, строковыми константами или функциями обработки строк).

“&” - знак между строковыми выражениями указывает, что производится конкатенация этих выражений. Он отделяется от выражения пробельными символами. В одном операторе может объединить любое количество строковых выражений.

Пример 1.

strMy = “Visual” & “Basic for Applications”

Примечание. Здесь объединяются две неименованные строковые константы. Результатом операции конкатенации будет значение Visual Basic for Applications).

## 2.7 Отладка программ и обработка ошибочных ситуаций

При разработке приложений VBA, как это бывает в программировании, возможны ошибки. Их можно разделить на три вида: синтаксические, ошибки при выполнении программы и логические ошибки.

Первый вид ошибок связан с допущенными опечатками или с недостаточным знанием языка программирования. Система указывает на некоторые типы синтаксических ошибок при вводе текста программы, на другие типы синтаксических ошибок – при попытке выполнения программы. Как правило,

количество синтаксических ошибок в программах резко сокращается по мере изучения языка программирования и приобретения опыта разработки программ.

Ошибки при выполнении программы случаются, если внешнее событие или необнаруженная синтаксическая ошибка вызывает останов программы в процессе ее работы. Например, в аргументе функции использовано неправильное имя файла, не закрыт замок дисковода гибких дисков, произошло обращение к выключенному принтеру и т.д.

Логическая ошибка является следствием плохо разработанного алгоритма, неучёта возможных значений входных данных и т.д. В этих случаях при выполнении программы получаются неверные результаты. Большая часть времени отладки программ является следствием допущенных логических ошибок.

В среде VBA имеются различные средства, которые можно использовать для отслеживания и исправления ошибок в программах.

Прежде всего, нужно отметить возможность простого обнаружения необъявленных переменных, что позволяет устранить большинство синтаксических ошибок. Способы идентификации логических ошибок состоят в установке точек останова (контрольных точек), пошаговом режиме выполнения отлаживаемой программы, наблюдении за значениями переменных и др. Имеется возможность создания обработчиков ошибок. Основные средства отладки доступны через меню Debug (Отладить).

Проще и быстрее выполнять отладочные операции с помощью панели инструментов Debug. Чтобы отобразить эту панель, нужно войти в меню View (Вид) выбрать команду Toolbars (Панели инструментов) и щелкнуть мышью по строке Debug.

Инструменты панели Debug слева направо имеют следующее назначение:

Run Sub/UserForm – Запуск программы/Формы

Break – Приостановка программы

Reset – Останов программы

Toggle Breakpoint – Включить/Выключить контрольную точку

Step Into – Пошаговая обработка с заходом в процедуры

Step Over – Пошаговая обработка без захода в процедуры

Step Out – Выход из процедуры

Locals Window – Включить/Выключить локальное окно

Immediate Window – Окно непосредственного выполнения

Watch Window – Включить/Выключить окно наблюдаемых переменных

Quick Watch – Быстрый просмотр значения переменной

Call Stack – Просмотр списка вызова процедур.

Как видно из рисунков 12 и 13, некоторые инструменты панели Debug дублируют или взаимосвязаны с меню Debug.

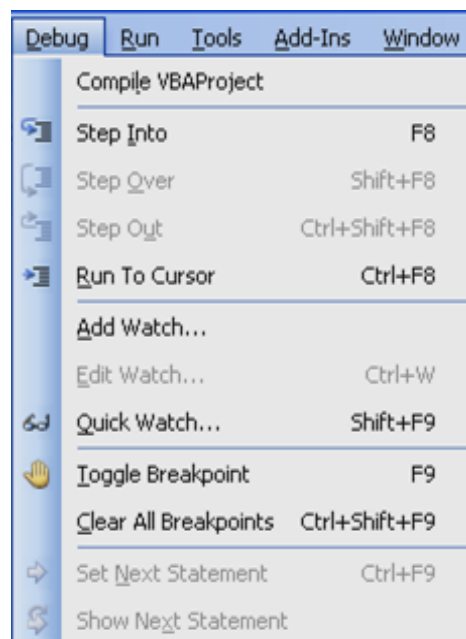


Рисунок 12 – Меню Debug



Рисунок 13 – Панель инструментов Debug

### 2.7.1 Отладка

При выполнении программы, если компилятор встречается с проблемами и не может их разрешить, он выводит на экран сообщение об ошибке, изображенном на рисунке 14, дающее пользователю возможность либо отладить программу, либо завершить ее работу. Если выбрать команду Run Sub/UserForm, на экран будет выведено кодовое окно, в котором строка, являющаяся причиной проблемы, будет выделена, как на рисунке 15.

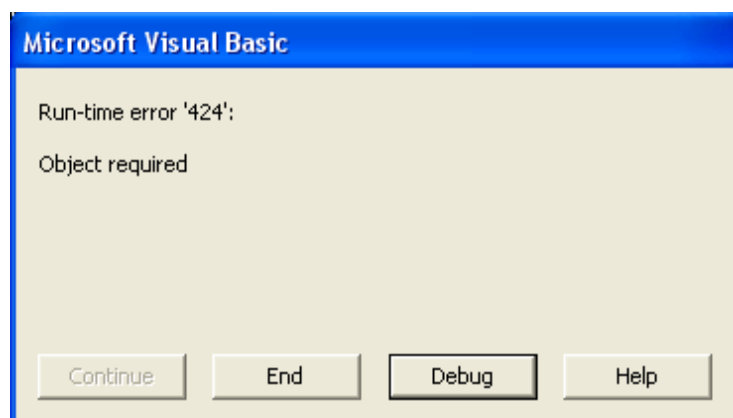


Рисунок 14 – Сообщение об ошибке на экране

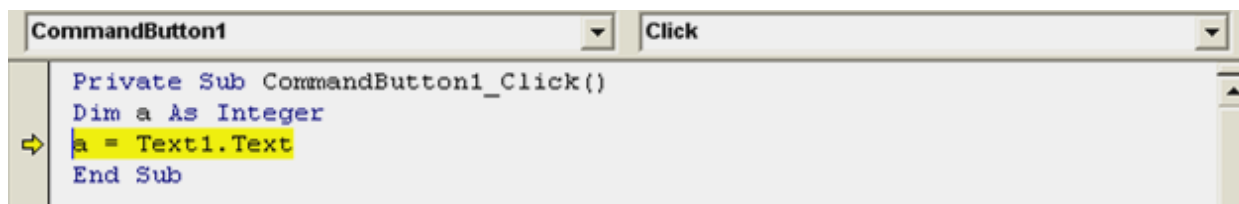


Рисунок 15 – Выбор команды Run Sub/UserForm

При этом программа не останавливается; она только временно прекращает свою работу. Следовательно, если ошибка не слишком серьезная, можно внести исправления и продолжить работу программы.

В такой ситуации доступны несколько опций.

1 Можно изменить строку, являющуюся причиной проблемы, выбрав в меню Run команду continue и продолжить работу программы. При этом программа запускается со строки, следующей за строкой, которая является причиной проблемы.

2 Можно запустить программу с другого места, щелкнув по другой строке, выбрав в меню Debug команду Set Next Statement и нажав клавишу F5. Однако эта опция обычно не приводит к успеху, поскольку она может нарушить логику программы.

3 Можно завершить программу, выбрав в меню Run команду Reset.

### 2.7.2 Точки останова (контрольные точки)

При отладке можно остановить выполнение программы в любой точке ее кода с помощью точек останова. Точка останова – это строка в программном коде, где прерывается выполнение программы. Создать точку останова можно одним из следующих способов.

1 Щелкнуть мышью в той строке кода, в которой требуется остановить выполнение программы, и нажать клавишу F9.

2 Вставить точку останова из меню Debug, выбрав в нем пункт Toggle Breakpoint (Установить точку останова).

Строка, на которой будет установлена точка останова, выделится красным цветом. Запущенная программа прервется на этой строке, после чего строка будет выделена желтым цветом и обозначена стрелкой, как показано на рисунке 16. Имея прерываемую таким способом программу, легко проверять значения переменных, перемещаться строка за строкой по коду, делать небольшие изменения в коде или продолжать работу программы, нажимая клавишу F5.

Точка останова может быть отменена щелчком мыши по строке, содержащей точку останова, или повторным нажатием клавиши F5. Для отмены всех точек останова нужно войти в меню Debug и выбрать команду Clear All Breakpoints (Убрать все точки останова).

При отладке программы точки останова используются для сужения той области программного кода, в которой предполагается наличие ошибки. Далее

исследовать состояние программы более детально можно с помощью просматриваемых значений (watches).

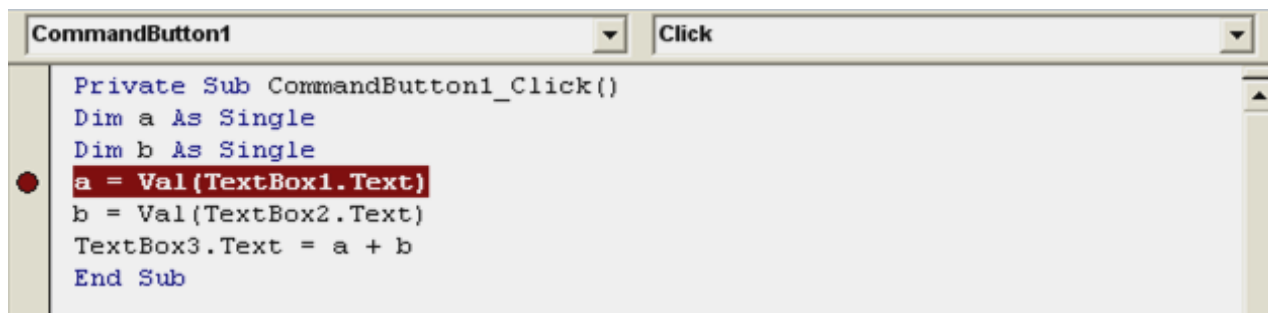


Рисунок 16 – Строка, на которой установлена точка останова

### 2.7.3 Наблюдение за данными с помощью просматриваемых значений

Для установки наблюдения за значением требуемой переменной нужно выполнить следующие действия.

1 Установить контрольную точку на той строке программы, которая содержит проверяемую переменную (клавиша F9).

2 Запустить программу, нажав клавишу F5 или выбрав команду Run Sub/UserForm.

3 Задержать на пару секунд указатель мыши над переменной, значение которой требуется просмотреть. Появится небольшое окно с текущим значением переменной, изображенным на рисунке 17.

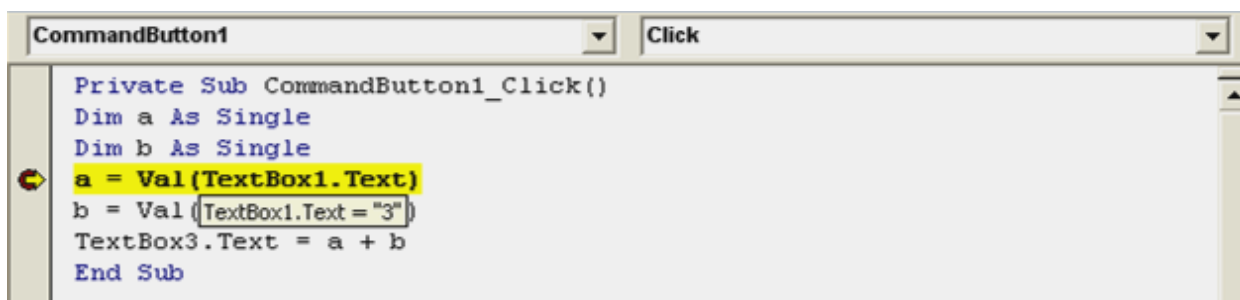


Рисунок 17 – Окно с текущим значением переменной

Иногда необходимо отслеживать значения нескольких переменных. Для этого используется окно “Watches” (Наблюдаемые переменные). Последовательность действий может быть следующей.

1 Установить точки останова в тех строках кода, где находятся проверяемые переменные, как показано на рисунке 18. Нажать клавишу F5 для запуска программы.

2 В точке останова подсветить переменную, которую необходимо добавить к списку просматриваемых значений.

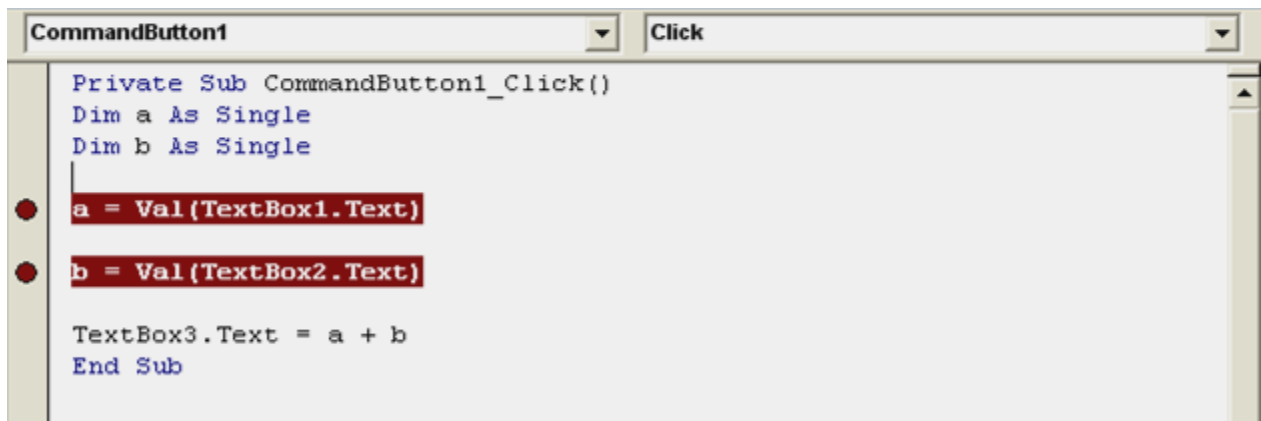


Рисунок 18 – Окно «Watches»

3 Щелкнуть правой кнопкой мыши и в появившемся контекстном меню на рисунке 19 выбрать команду Add Watch (Добавить наблюдаемую переменную).

4 В появившемся диалоговом окне “Add Watch” выбрать нужные установки: имя переменной (Expression), область видимости переменной (Context), тип наблюдения (Watch Type). Аналогично добавить другие наблюдаемые переменные и щелкнуть по кнопке <OK>, как показано на рисунке 20.

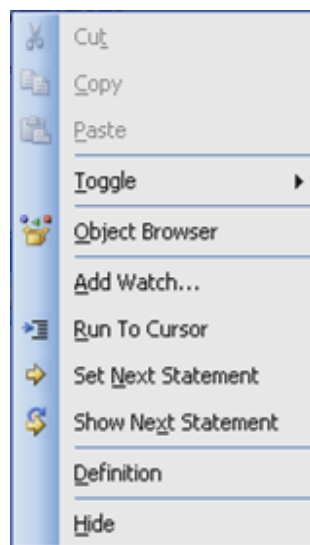


Рисунок 19 – Контекстное меню

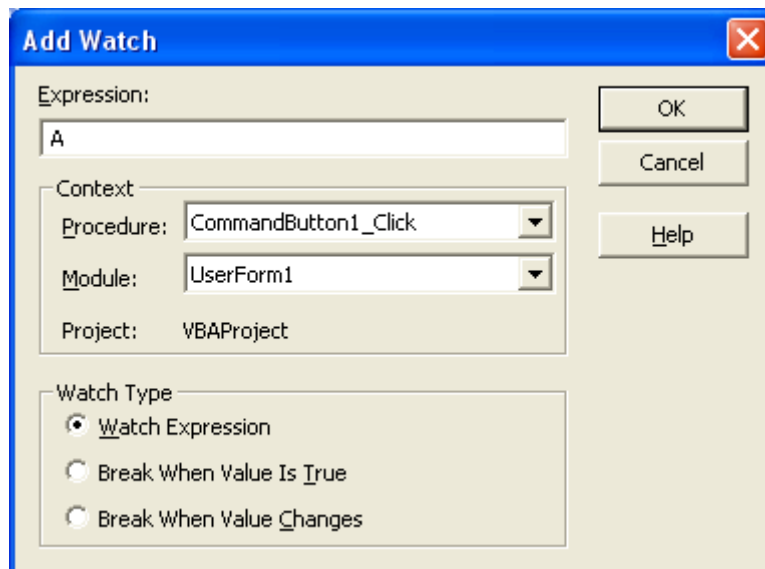


Рисунок 20 – Окно для добавления наблюдаемых переменных

5 Для того чтобы отобразить окно “Watches”, показанному на рисунке 21, войти в меню View и выбрать команду Watch Window (Окно наблюдения).

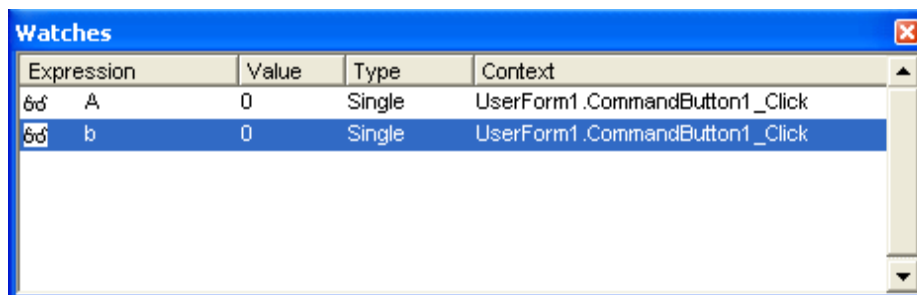


Рисунок 21 – Окно «Watches»

## 2.7.4 Пошаговое выполнение программы

Пошаговое выполнение применяется для проверки каждой строки кода программы в порядке ее выполнения. Есть два режима такого выполнения: Step Into (пошаговая обработка с заходом в процедуры) и Step Over (Пошаговая обработка без захода в процедуры). В первом случае выполняются все строки программы, в том числе строки вызываемых процедур. Во втором случае вызываемые процедуры выполняются как один оператор, а затем выполняется следующая строка программного кода.

Последовательность действий по использованию панели Debug для отладки программы.

- 1 Открыть отлаживаемый проект.
- 2 В меню View выбрать команду ToolBars (Панели инструментов) и щелкнуть на строке.
- 3 Отобразить окно “Code”, в котором находится текст программы.

4 Щелкнуть на кнопке <Step Into> (Войти в инструкцию) на панели Debug для выполнения первого оператора в процедуре. Он будет выделен желтым цветом. Это выделение позволит далее проследить логику работы программы.

5 Будет выполнен оператор Sub, а следующий оператор будет выделен. Если в этом операторе есть проверяемая переменная, определяющая дальнейший ход программы, ее нужно поместить в окно “Watches” (Наблюдения). Это позволит видеть, как изменяется ее значение в ходе выполнения программы.

6 Выделить проверяемую переменную мышью, а затем щелкнуть по кнопке <Quick Watch> (Быстрый просмотр переменных) на панели Debug. На экране появится диалоговое окно, в котором будут содержаться имя переменной, ее значение, тип и назначение.

7 Выбрать в диалоговом окне команду Add. В нижней части экрана появится окно “Watches”, в котором будет отображена выбранная переменная. Аналогично в это окно можно поместить другие переменные, определяющие логику программы.

8 Щелкнуть по кнопке <Step Into> для выполнения следующего оператора и т.д.

Таким образом, в пошаговом режиме выполнения программы, наблюдая значения переменных в окне “Watches”, можно определить реальную логику программы и выявить ее отклонения от нужного алгоритма, т.е. обнаружить ошибку.

### 2.7.5 Обработка ошибок

Каждая ошибка в VBA имеет свой код. В таблице 12 приведены коды наиболее часто встречаемых ошибок.

Таблица 12 - Основные коды ошибок в VBA

Код ошибки	Сообщение
5	Приложение не запущено
6	Переполнение
7	Не хватает памяти
9	Индекс выходит за пределы допустимого диапазона
11	Деление на нуль
13	Несоответствие типа
18	Произошло прерывание, вызванное пользователем
52	Неправильное имя файла или идентификатор
53	Файл не найден
54	Неверный режим работы с файлом
55	Файл уже открыт
56	Ошибка ввода/вывода
61	Переполнение диска
68	Устройство недоступно
71	Диск не готов



72	Повреждена поверхность диска
Продолжение таблицы 12	
335	Невозможен доступ к системным ресурсам
368	Истек срок действия данного файла. Программе требуется файл более новой версии
82	Ошибка принтера

### Лабораторная работа № 1

#### Программирование алгоритмов линейной структуры

Цель: приобретение навыков программирования алгоритмов линейной структуры с помощью подпрограммы-функции, вычисляющей значение арифметических выражений. Индивидуальные варианты лабораторной работы № 1 представлены в таблицах 13, 14 (2 часа).

Таблица 13

№ В	Варианты индивидуальных заданий
Разработать функцию пользователя, вычисляющую значение арифметического выражения в заданных точках	
1	$y = \sqrt{ax^2 + \sqrt[3]{\lg(2 \cdot \sin^2 x)}} - \operatorname{arctg}\left(\frac{a-x}{4.5}\right);$ $a=0.83, x=1.2$
2	$y = \arccos(\lg x^2 - a^{(x+4.3)} ) - e^{\sin x-a^{1/4} };$ $a=0.12, x=0.36$
3	$z = \sqrt[5]{\frac{a-b^3}{\operatorname{tg}(e^{ x+ab })}} + \sin^3(x^{1/4});$ $a=0.394, b=0.124, x=0.842$
4	$y = \sqrt[3]{2 \cdot \sqrt{\frac{\ln x-a }{a+b \cdot x}}} + \sqrt[5]{\sin\left(\lg\left[\frac{x-b}{\operatorname{arctg}(x)}\right]\right)};$ $a=0.124, b=1.14, x=2.125$
5	$z = \frac{x^{1/3} - a}{3} \cdot \left( \frac{\sin^3(\ln^2 x) \cdot e^{ax}}{x \cdot \arcsin\left(\frac{a}{x}\right) + \sqrt{a^2 + x^2}} \right)^{1/4};$ $a=0.343, x=0.834$

6	$y = \ln \lg x   - \frac{(\ln x)^2}{a - \operatorname{arctg}\left \operatorname{tg}\frac{x}{a}\right };$ $a=0.334, x=1.256$
---	-----------------------------------------------------------------------------------------------------------------------------

Продолжение таблицы 13

7	$z = \left  \frac{\left(\arcsin\left(\frac{x}{a}\right)\right)^{3x-a} + \cos^3 x-a }{e^{\lg(x-\ln ax )}} \right ;$ $x=0.104, a=4.213$
8	$y = \left[ \arcsin\left(\frac{x}{a}\right) \cdot e^{ ax } \cdot \cos^2\sqrt[3]{x-a} \right]^{\ln\left \frac{x}{a-3.17}\right };$ $x=0.117, a=2.346$
9	$\alpha = \beta^{\frac{1}{3}} \cdot \ln x - e^{\beta \cdot x}  + \operatorname{arctg}^{\frac{2}{3}}\left(\frac{x + \beta \cdot \sqrt{4.75 - \beta \cdot x}}{\operatorname{arctg}(\sin x )}\right);$ $\beta=0.2, x=0.114$
10	$z = \left  e^{ x-a \cdot \arcsin^2(ax-0.441) } - \sqrt[3]{\frac{a^5 - \sqrt{\ln x}}{\cos ax }} \right ;$ $x=2.512, a=0.135$
11	$y = \left  \frac{\sin\left[\arcsin\left(\sqrt{ x-0.8 \cdot \ln(bx) }\right)\right]}{a - b \cdot \sin^2(x-a) \cdot e^{\ln x}} \right ;$ $x=2.126, b=2.438, a=0.324$
12	$z = \frac{x}{\operatorname{arctg}(e^{x-a})} - \frac{\ln\left(\sqrt[3]{b+a \cdot e^{ax}}\right)}{ax-1};$ $x=0.834, b=0.242, a=0.125$
13	$y = x \cdot \left(\ln x^{\frac{1}{3}}\right)^{\sqrt{ x^3-a }} + \frac{e^{\sin^3 x}}{1 + a \cdot e^{\ln 4.2-x }};$ $a=0.344, x=0.125$
14	$z = \frac{1}{2ab} \cdot \ln \left  \frac{b \cdot \operatorname{tg} x + a}{a^2 \cdot \cos^2 x-a  - b \cdot \sin x^2} \right  \cdot \operatorname{arctg}\left(\frac{b \cdot \operatorname{tg} x}{a-1}\right);$ $a=2.124, b=0.835, a=0.54$

15	$y = \frac{\left[ a + \frac{1}{2} \ln  \sin x + \cos(\ln  ax - b )  \right]^{a-x}}{tg^{\frac{1}{3}}(\ln^3 2x)};$ $a=0.344, b=1.124, x=0.455$
----	----------------------------------------------------------------------------------------------------------------------------------------------

Продолжение таблицы 13

16	$z = \frac{tgx \cdot \left[ 1 + \frac{1}{a^2} \cdot tg^2(\sqrt{\ln  x - a }) \right]}{\arccos \left( \frac{\sqrt{ x - a }}{\sqrt{x}} \cdot \cos x \cdot \sin^2 x \right)};$ $x=0.345, a=1.242$
17	$y = \frac{ctg \left[ \sqrt{\cos  x - e^{ax-1.2} } \right]}{\ln  \sin^2 x + \cos x^2  - tg(\arcsin  x - a )};$ $x=0.117, a=0.832$
18	$z = a + \sqrt{a^2 + e^{ x-b \cdot \ln  x-a  }} - \ln \left  \frac{a - b \cdot \lg x}{a + b \cdot tg^2 x} \right ;$ $a=0.124, b=0.338, x=0.817$
19	$y = \frac{1}{a^{\frac{1}{3}} - b \cdot \sqrt{ x - ab }} \cdot \left[ x - \left  \frac{b}{a} \right  \cdot arcctg \left( - \left  \frac{b}{ax - e^x} \right  \cdot tgx \right) \right];$ $a=0.274, b=0.392, x=0.736$
20	$z = \arccos \left[ \frac{\sqrt[3]{b - \sqrt{a}}}{\sqrt{bx + \sin  x - a }} \cdot \cos(e^{\ln^3 x}) \right];$ $b=0.614, a=0.125, x=1.23$

Таблица № 14

№ В	Варианты индивидуальных заданий
Разработать экономическую функцию пользователя	
1	<p>Рассчитать темп инфляции, если известен индекс цен прошлого года и текущего года</p> $T = ((\text{ид.т.г} - \text{ид.п.г}) / \text{ид.т.г}) * 100$

2	<p>Вычислить доход на акцию по формуле:</p> $Y=(D/P)*100$ <p>где  Y - доход на акцию;  D - дивиденд;  P - цена приобретения.</p>
---	----------------------------------------------------------------------------------------------------------------------------------------------

Продолжение таблицы 14

3	<p>Рассчитать индекс совокупной факторной производительности (SFP), если известны:</p> $SFP=UZP/(SP+A+M).$ <p>SP - фонд зарплаты рабочих и служащих;  A - сумма амортизационных отчислений т.г. в неизменных ценах того года, когда было приобретено оборудование;  UZP - объем условно-чистой продукции;  M - стоимость.</p>
4	<p>Вычислить сумму оборотных средств</p> $S=VP/360*(D1-D2)$ <p>где  VP - валовый национальный продукт;  D1- прогнозируемая длительность оборота;  D2- фактическая длительность в днях.</p>
5	<p>Рассчитать прибыль от вложенного в производство капитала за периоды от 1 до 5 лет по формулам:</p> $Vp=Vt/(1+r)^t; r=d/k*100\%$ <p>где  Vp - будущая стоимость сегодняшней суммы денег;  Vt - сегодняшняя стоимость будущей суммы денег;  r - ставка процента;  t - количество лет;  k - капитал;  d - доход на капитал.</p>
6	<p>Налогооблагаемый оборот фирмы составил: по производству непродовольственных товаров (ставка налога na=20%)-a=1340000 р; выпуску продовольственных товаров (ставка налога nb=10%)-b=460000 р; транспортным услугам (ставка налога nc=16.67%) c=200000 р; общественному питанию (ставка налога nd=9.09%)-d=100000 р.</p> <p>Рассчитать среднюю ставку налога</p> $ch=(na*a+nb*b+nc*c+nd*d)/(a+b+c+d)*100$

7	<p>Начислить зарплату рабочему предприятия, используя следующие данные: месячный оклад, количество рабочих дней в феврале месяце и количество отработанных рабочим дней.</p> $З = О / Д * Рд$ <p>где  З - зарплата за месяц;  О - месячный оклад;  Д - рабочие дни в феврале;  Рд - количество отработанных дней.</p>
---	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Продолжение таблицы 14

8	<p>Коммерческий банк выдал краткосрочный кредит юридическому лицу на сумму 200 млн.р. под 10% годовых. Составить программу, вычисляющую годовую процентную ставку за кредит, если известно, что общее число платежей за год (m) равно 12, общая сумма процента (ch) - 200000000 р., а реальное число сделанных вкладов равно 10. Формула вычисления годовой процентной ставки: <math>apr = 2 * mch / (p * (n + 1))</math>.</p>
9	<p>Рассчитать прибыль фирмы на единицу продукции (p) по формуле:</p> $P = c - (a + z1 + z2) / b$ <p>где  с - цена единицы продукции;  а - амортизационные отчисления;  z1 - постоянные затраты;  z2 - переменные затраты;  b - выпуск продукции;  при постоянных затратах 100000.</p>
10	<p>Составить программу для вычисления коэффициента экономической эффективности, используя следующую формулу:</p> $K = (S1 - S2) / (K1 - K2)$ <p>где  S1 и S2 - себестоимость до и после внедрения;  K1 и K2 - капитальные вложения до и после внедрения.</p>
11	<p>Рассчитать себестоимость продукции растениеводства, которая вычисляется по формуле:</p> $c = t + z + u + s$ <p>где  с - себестоимость продукции;  т - транспортные издержки;  s - стоимость посадочного материала;  z - зарплата рабочим.</p>

12	<p>Вычислить рыночную стоимость облигации по формуле</p> $PV = P/EXP(N*LN(1+R))$ <p>где  P - государственная стоимость;  R - биржевая стоимость.</p>
13	<p>Рассчитать годовую производственную мощность выработки муки мелькомбинатом по формуле:</p> $M=MS*B*P/100(\text{тонн/год})$ <p>где  M - годовая мощность;  B - выход муки,%;  P - годовой рабочий период в днях;  MS - суточная мощность: <math>MS=R*N/1000</math>, где R-размер рабочих органов машины;  N-техническая норма нагрузки на единицу рабочего органа машины, кг.</p>

Продолжение таблицы 14

14	<p>Вычислить уровень инфляции, которая вычисляется по формуле:</p> $R=\frac{\text{цены текущего периода}-\text{цены прошлого периода}}{\text{цены прошлого периода}}*100\%,$ <p>если известны:  a - цены текущего периода;  b - цены прошлого периода;  где R-уровень инфляции.</p>
15	<p>Подсчитать скорость обращения денег (v) в кругообороте доходов, которая вычисляется по формуле:</p> $V=P*S/M$ <p>где  M=38526000 - предложение денег;  P=12865.7 - средняя цена товаров;  S=2976 - общий объём товаров.</p>
16	<p>Подсчитать средние общие издержки фирмы, используя формулы:</p> $A=F+V, F=T/Q, V=N/Q$ <p>где  A - средние общие издержки фирмы;  F - средние постоянные;  V - средние переменные;  T - сумма постоянных издержек;  N - сумма переменных издержек;  Q - количество произведенной продукции.</p>

17	<p>Вычислить темп инфляции этого года по формуле:</p> $T=(IN-IP)*100/IP$ <p>где  IN - индекс цен этого года=113.6;  IP - индекс цен прошедшего года=118.3;  Ответ должен быть в процентах.</p>
18	<p>Вычислить показатель ценовой эластичности спроса <math>E_p</math>, используя следующую формулу:</p> $E_p=(Q1-Q2)/(P1-P2),$ <p>где  Q1,Q2 - величина спроса (%);  P1, P2 - прирост цены (%).</p>

#### Продолжение таблицы 14

19	<p>Рассчитать валовые издержки по формуле:</p> $TC=FC+VC$ <p>где TC - валовые издержки; FC – постоянные издержки; VC - переменные издержки.</p> <p>Рассчитать прибыль фирмы по формуле:</p> $P=C-((A+Z1+Z2)/B)$ <p>где A – амортизация; B – продукция; C – цена; P – прибыль; Z1 - постоянные издержки; Z2 - переменные издержки.</p>
20	<p>Какой доход получит владелец акции, если ее номинальная стоимость составляет 250 р., цена приобретения - 1000 р., а дивиденд - 100 р.</p> $Y=(D*100)/P$ <p>где  Y-доход на акцию;  D-дивиденд;  P-цена приобретения.</p>

## 2.8 Условный оператор

Условный оператор VBA позволяет проверить некоторое условие и в зависимости от результатов проверки выполнить то или иное действие. Таким образом, условный оператор – это средство ветвления вычислительного процесса.

В VBA существует 2 типа условного оператора: линейный и блочный.

1 Линейный условный оператор используется для того, чтобы выполнить какой-либо один оператор, если некоторое условие будет истинным.

Синтаксическая конструкция линейного оператора имеет две формы: безальтернативную, альтернативную.

Структура безальтернативного условного оператора (сокращенный вариант):

If <условие> Then <оператор 1>

Структура альтернативного условного оператора (полный вариант):

If <условие> Then <оператор 1> Else <оператор 2>

где

If, Then, Else - зарезервированные слова (если, то, иначе);

<условие> - произвольное выражение логического типа;

<оператор 1>, <оператор 2> - любые операторы языка VBA.

Работа. Вначале вычисляется условное выражение <условие>. Если результат есть True (истина), то выполняется <оператор 1>, а <оператор 2> пропускается. Если результат есть False (ложь), наоборот, <оператор 1> пропускается, а выполняется <оператор 2>.

2 Блочный условный оператор используется в случае истинности условия необходимо выполнить несколько программных операторов (блок операторов). Блочный оператор имеет две формы: безальтернативный, альтернативный.

Структура безальтернативного блочного оператора (сокращенный вариант)

If <условие> Then

<оператор1>

<оператор2>

.....

<оператор n>

End If

где

End If - указывает на окончание блока оператора If.

Структура альтернативного блочного оператора:

If <условие> Then

<оператор1>

<оператор2>

.....

<оператор n>

Else

<оператор1>

<оператор2>

.....

<оператор n>

End If

Пример 1.

Постановка задачи. Создать в стандартном модуле пользовательскую процедуру вычисления уравнения вида  $ax^2 + bx + c = 0$ .

Технология выполнения задания:



1 Исходные данные:

$a, b, c \in \mathbb{R}$

Результат:  $x_1, x_2 \in \mathbb{R}$ .

2 Набрать в стандартном модуле проекта следующую пользовательскую процедуру:

```
Private Sub uravnenie ()  
a = InputBox("a=", a)  
b = InputBox("b=", b)  
c = InputBox("c=", c)  
d = b ^ 2 - 4 * a * c  
If d >= 0 Then  
x1 = (-b + Sqr(d)) / (2 * a)  
x2 = (-b - Sqr(d)) / (2 * a)  
MsgBox (x1)  
MsgBox (x2)  
Else  
MsgBox ("Решений нет")  
End If  
End Sub
```

3 Вычислить корни квадратного уравнения при произвольных исходных данных.

Альтернативный блочный оператор If применяется в тех случаях, когда при выполнении условия необходимо осуществить один набор программных операторов, а при невыполнении – другой.

Операторы IF могут быть вложенными друг в друга. Такое вложение операторов применяется, если нужно проверить какое-либо условие при другом условии, которое является истинным.

Формат вложенного оператора If:

```
If <условие1> Then  
    If <условие2> Then  
        <оператор1>  
        <оператор2>  
        .....  
        <оператор n>  
    Else  
        <оператор1>  
        <оператор2>  
        .....  
        <оператор n>  
    End If  
End If
```

Пример 2.

Постановка задачи. Создать в стандартном модуле пользовательскую функцию нахождения максимального среди трех заданных чисел  $y_1 = a+2*b$ ;  $y_2 = a*b+c$ ;  $y_3 = c^2 + 1$ .

Технология выполнения задания:

1 Исходные данные:

$a, b, c \in R$

Результат:  $Max \in R$ .

2 Набрать в стандартном модуле проекта следующую пользовательскую функцию:

```
Function Max(a,b,c)
y1 = a+2*b
y2 = a*b+c
y3 = c^2+1
If y1 > y2 Then
If y1 > y3 Then y = y1 Else y = y3
Else
If y2 > y3 Then y = y2 Else y = y3
End If
End Function
```

3 Вычислить корни квадратного уравнения при произвольных исходных данных.

При использовании вложенных операторов If важно не перепутать варианты сочетания условий. Нужно помнить правило: альтернатива Else считается принадлежащей ближайшему оператору If, не имеющему ветви Else.

В VBA предусмотрена конструкция для работы с несколькими операторами If. Эти операторы применяются в случаях, когда необходимо рассмотреть еще несколько условий в дополнение к исходному. Для этого служит конструкция: If...Then...ElseIf. В отличие от вложенных операторов конструкция с несколькими операторами If позволяет проверить дополнительное условие, если исходное условие принимает значение False.

Формат записи:

```
If <условие1> Then
    <оператор1>
ElseIf <условие2> Then
    <оператор2>
Else
    <.оператор3>
EndIf
```

Пример 3.

Постановка задачи. Менеджеру по продажам необходимо разработать функцию, позволяющую рассчитывать комиссионные. Процент комиссионных зависит от объема проданного товара и начисляется по следующему правилу, представленному в таблице 15.

Таблица 15 - Правила расчета комиссионных

Объем продаж за неделю, р.	Комиссионные, %
От 0 до 9999	8

От 10000 до 19999	10
От 20000 до 39999	12
Более 40000	14

Технология выполнения задания:

1 Исходные данные:

Продажи  $\in \mathbb{Z}$ .

Результат: Комиссионные  $\in \mathbb{R}$ .

2 Построить для расчета комиссионных в стандартном модуле пользовательскую функцию:

Function Комиссионные (Продажи)

If Продажи  $\leq 9999$  Then

Комиссионные = Продажи \* 0.08

ElseIf Продажи  $\leq 19999$  Then

Комиссионные = Продажи \* 0.1

ElseIf Продажи  $\leq 39999$  Then

Комиссионные = Продажи \* 0.12

Else

Комиссионные = Продажи \* 0.14

End If

End Function

3 Вычислить.

## Лабораторная работа № 2

### Программирование алгоритмов разветвляющейся структуры

Цель: приобретение навыков программирования алгоритмов разветвляющейся структуры с помощью пользовательской подпрограммы-процедуры, где на определенном этапе производится выбор очередного выполняемого оператора в результате анализа некоторых условий. Индивидуальные варианты лабораторной работы № 2 представлены в таблицах 16, 17 (2 часа)

Таблица 16

№ В	Варианты индивидуальных заданий
Разработать процедуру пользователя, вычисляющую значение функции	
1	$y = \begin{cases} \arctg(2w + e^{w-a}), & w < 2 \\ \cos w - (a+b)^2 , & w \geq 2 \end{cases} \quad \text{где } w = \ln \left  \frac{2x^2 + 3x + 4}{16x - a} \right $ <p>a=2.85, b=1.56, x=0.63</p>

2	$\gamma = \begin{cases} \alpha\beta^{0.8} - \ln e^x - \sin^3(x - \pi) , 1 \leq x \leq 1.5 \\ \sqrt{\left  \arcsin \frac{x - \alpha\beta}{\pi} - \alpha^2 \right }, 2.5 \leq x \leq \pi \\ \text{не определено в остальных случаях} \end{cases}$ $\alpha=12.5, \beta=0.03, x=5.256$
3	$y = \begin{cases} x^2 - a \cdot \ln \sin(x^{3.2} - a^3) ; \frac{1}{a} \leq x \leq 1 \\ \arctg^3 \frac{a+b}{x-a}; x=5 \text{ или } x=8.1 \\ \text{не определено в остальных случаях} \end{cases}$ $a=8.7, b=-5.2, x=8.1$

Продолжение таблицы 16

4	$s = \begin{cases} \frac{\sqrt{(ax)^2 + 2} - 1.8ax}{0.5 \sin(2ax) + \ln ax - 1  + x}; 0 < x < \frac{\pi}{2a} \text{ и } x \neq \frac{1}{a} \\ e^{ax} - 2\pi \cdot (ax + 0.8); \frac{\pi}{2a} \leq x \leq \frac{\pi}{a} \\ \text{не определено в остальных случаях} \end{cases}$ $x=1.204, a=0.8$
5	$\tau = \begin{cases} e^{z - \sin z} + e^{z+1} - z, \text{ при } z - \sin z \geq 0 \\ 1 + \ln z - \sin z , \text{ при } z - \sin z < 0 \end{cases}$ $\text{где } z = \begin{cases} \cos\left(\frac{x^2 + 2}{x} - \pi x\right), x > 1 \text{ или } x < -2.5 \\ \frac{x}{\pi}, \text{ в остальных случаях} \end{cases}$

6	$y = \begin{cases} \frac{a^5 \cdot \sin 1-x ^{0.3}}{\ln \left  \cos \frac{1}{x} \right }, -1.5 \leq x \leq -1 \\ e^{-x+4\pi \cdot \lg x}, -5 < x \leq -2 \\ 0 - \text{в остальных случаях} \end{cases}$ <p><math>a=3.08, x=-0.004</math></p>
7	$y = \begin{cases} 3\cos^2(2x-16x^2+8), x < 0.5 \\ 2\sin^2 \frac{2x-9}{x^2+8}, x \geq 0.5 \end{cases}$
8	$y = \begin{cases} \sqrt[3]{ a - e^x \cdot \cos x }, -2 \leq x \leq 5 \\ x^{2.5} - \arctg\left(a - \frac{b}{x}\right)^{1/2}, 5 < x \leq 6 \\ 2.5, \text{ в остальных случаях} \end{cases}$ <p><math>a=16.01, b=18.1, x=5.93</math></p>

Продолжение таблицы 16

9	$y = \begin{cases} \frac{\alpha\beta}{x-\pi} \cdot e^x, -1 \leq x \leq 3 \\ \sqrt[3]{ \alpha - x^3 \beta \sin x }, -3 < x \leq -2 \\ \alpha^x, \text{ в остальных случаях} \end{cases}$ <p><math>\alpha=7.02, \beta=16.02, x=1.085</math></p>
10	$y = \begin{cases} c \cdot e^{\sin(3x-1)} - a \cdot \cos x, x \geq 2.4 \\ b \cdot e^{\ln x-a }, x \leq -0.15 \\ x, \text{ в остальных случаях} \end{cases}$ <p><math>x=2.4, a=2.65, b=4.24, c=1.83</math></p>

11	$y = \begin{cases} ( z - ax^2 + b  + a)^{x+1}, & x > 0.82 \\ a \cdot \ln \left  \frac{ax^2 + bx - z}{\cos^2 x + b} - \sin(\ln x  - b) \right , & x \leq 2, \text{ где } z = \sqrt{ ax^2 - bx^3 } \\ \text{не определено в остальных случаях} \end{cases}$ <p>a=0.36, b=1.18, x=1.16</p>
12	$t_1 = \begin{cases} ax - bx^2 + \cos(\ln ax^2 - b  + 1), &  x  < \sqrt{b/a} \text{ и } x \neq 0 \\ e^{x^2 - a \cdot \ln b} + 1, & \text{в остальных случаях} \end{cases}$ $t_2 = \begin{cases} \ln t_1 , & x \neq 0 \\ t_1, & \text{в остальных случаях} \end{cases}$ <p>x=0.12, a=0.111, b=0.348</p>
13	$y = \begin{cases} x^{4/5} \cdot \cos^{3/2} x, & 0.5 \leq x \leq 1 \\ \lg(\sqrt[3]{x} + e^{x-1}), & x \geq 1 \\ 0, & \text{в остальных случаях} \end{cases}$
14	$y = \begin{cases}  \arccos x  - \ln x - 1.2 , & 0.1 < x < 0.8 \\ \frac{x+1}{e^x}, & x \geq 1.2 \\ \text{не определено в остальных случаях} \end{cases}$

Продолжение таблицы 16

15	$y = \begin{cases} 2 \sin \frac{x+1}{\sqrt{x}}, & x \geq 2.5 \\ \frac{\arcsin x^2}{\operatorname{arctg} x} + x^{2/3}, & 0.5 < x < 1 \\ -x, & \text{в остальных случаях} \end{cases}$
----	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

16	$y = \begin{cases} 1 + \frac{x}{1+x^2}, x < 0 \text{ или } x > 3 \\ \ln(1 + x \cdot e^{x-1}), 1 \leq x \leq 2 \\ \text{не определено в остальных случаях} \end{cases}$
17	$y = \begin{cases} \sqrt[3]{x^2 - 1} + \frac{1}{\sqrt{2 + \sin x}}, x \geq 1 \\ 2x^2 +  x + 1 , -2 \leq x < 0 \\ 1, \text{ в остальных случаях} \end{cases}$
18	$y = \begin{cases} e^{ x -4}, x \geq \ln 0.256 \\ \sqrt{ x+1 } \cdot [2x^2 - 6(\ln x  - 1.83) + 0.18], -5 < x < \ln 0.256 \\ \text{не определено в остальных случаях} \end{cases}$
19	$z = \begin{cases} \ln \left  \frac{y^2 - e^x}{y^2 - 1} - y^2 \right ,  y  < 1 \\ x \cdot \arccos \frac{\pi}{y},  y  > \pi \\ x - y \cdot e^{-y}, \text{ в остальных случаях} \end{cases} \quad \text{где } y = \operatorname{arctg} \frac{x^2 - \pi}{\pi - 1}$ <p><math>x = -0.37</math></p>
20	$y = \begin{cases} x \cdot \operatorname{tg}^2(\alpha x + \beta), 0 < x \leq 2 \\ 2 \sin(\alpha x + \beta), -1 < x \leq 0 \\ \alpha x + \beta, \text{ в остальных случаях} \end{cases} \quad \text{где } \alpha = e^{-2x}, \beta = \begin{cases} \alpha \ln x, x > 0 \\ \cos 2x, x \leq 0 \end{cases}$

Таблица 17

<b>№ В</b>	<b>Варианты индивидуальных заданий</b>
	Разработать экономическую функцию пользователя

1	<p>Вычислить надбавку к зарплате сотрудника предприятия в зависимости от стажа его работы. Значения рассчитать по формулам: Стаж работы = (Текущая дата – Дата поступления на работу)/365. Результат округлить до целого. Значение зарплаты подобрать самостоятельно.</p> <p>Надбавка = <math>\begin{cases} 0\% \text{ от зарплаты, если стаж работы меньше 5 лет,} \\ 5\% \text{ от зарплаты, если стаж работы от 5 до 10 лет,} \\ 10\% \text{ от зарплаты, если стаж работы больше 10 лет} \end{cases}</math></p>																		
2	<p>Вычислить годовой подоходный налог с физических лиц. Принята следующая налоговая сетка:</p> <table><tr><th colspan="2">Доход</th><th>%</th></tr><tr><th>от:</th><th>до:</th><th>налога</th></tr><tr><td>0</td><td>10000</td><td>12%</td></tr><tr><td>10001</td><td>30000</td><td>18%</td></tr><tr><td>30001</td><td>60000</td><td>25%</td></tr><tr><td>60001</td><td>и выше</td><td>35%</td></tr></table>	Доход		%	от:	до:	налога	0	10000	12%	10001	30000	18%	30001	60000	25%	60001	и выше	35%
Доход		%																	
от:	до:	налога																	
0	10000	12%																	
10001	30000	18%																	
30001	60000	25%																	
60001	и выше	35%																	
3	<p>Менеджеру по продажам необходимо разработать функцию, позволяющую рассчитывать комиссионные. Процент комиссионных зависит от объема проданного товара и начисляется по следующему правилу, представленному в таблице</p> <table><tr><th>Объем продаж за неделю, р.</th><th>Комиссионные, %</th></tr><tr><td>От 0 до 29 999</td><td>3</td></tr><tr><td>От 30 000 до 59 999</td><td>4,5</td></tr><tr><td>От 60 000 до 119 999</td><td>6,8</td></tr><tr><td>Свыше 120 000</td><td>7,4</td></tr></table>	Объем продаж за неделю, р.	Комиссионные, %	От 0 до 29 999	3	От 30 000 до 59 999	4,5	От 60 000 до 119 999	6,8	Свыше 120 000	7,4								
Объем продаж за неделю, р.	Комиссионные, %																		
От 0 до 29 999	3																		
От 30 000 до 59 999	4,5																		
От 60 000 до 119 999	6,8																		
Свыше 120 000	7,4																		
4	<p>Вычислить сдельную зарплату работника, а также сумму налога и сумму на руки. Значения рассчитать по формулам: &lt;зарплата&gt; = &lt;обработано деталей&gt; * &lt;стоимость работы&gt; - &lt;деталей брака&gt;*&lt;стоимость детали&gt;.</p> <p>Налог исчисляется в зависимости от зарплаты: если она выше 5000 р. – налог составляет 20%, если нет – 13%.</p> <table><tr><td>Стоимость работы:</td><td>80</td></tr><tr><td>Стоимость детали:</td><td>200</td></tr></table>	Стоимость работы:	80	Стоимость детали:	200														
Стоимость работы:	80																		
Стоимость детали:	200																		

Продолжение таблицы 17



	Вычислить эффект мультипликатора (М) по формуле $M=Q/D,$ где Q - объем производства; D - национальный доход.																	
5	В зависимости от результата вывести на экран следующие сообщения: для получения высоких результатов необходимо повысить объёмы производства (M=1); необходимо снизить объём национального дохода (M<1); вы добились успехов: эффект мультипликатора работает (M>1).																	
6	<p>Вычислить зарплату с учетом увеличивающегося подоходного налога и числа детей.</p> <p>Значения рассчитать по формулам: &lt;сумма обложения&gt;=&lt;доход&gt;-&lt;детей&gt;*&lt;миним.зарплата&gt; &lt;налог&gt;=&lt;процент налога&gt;*&lt;сумма обложения&gt;. &lt;сумма на руки&gt; = &lt;сумма обложения&gt; – &lt;налог&gt;</p> <p>Процентная ставка налога</p> <table><tr><td>Доход, р.</td><td>% налога</td></tr><tr><td>от 0 до 1 999</td><td>0</td></tr><tr><td>от 2000 до 3 999</td><td>12</td></tr><tr><td>от 4000 до 5 999</td><td>15</td></tr><tr><td>от 6000 до 9 999</td><td>20</td></tr><tr><td>свыше 10 000</td><td>30</td></tr></table>						Доход, р.	% налога	от 0 до 1 999	0	от 2000 до 3 999	12	от 4000 до 5 999	15	от 6000 до 9 999	20	свыше 10 000	30
Доход, р.	% налога																	
от 0 до 1 999	0																	
от 2000 до 3 999	12																	
от 4000 до 5 999	15																	
от 6000 до 9 999	20																	
свыше 10 000	30																	
7	<p>Составить программу, позволяющую произвести расчет зарплаты медсестры кардиологического отделения с окладом 1 категории по единой тарифной сетке, при условии, что если количество отработанных часов (t) равно 160, то</p> $z=1345.5*160-(s+k-75900)*12\%-1\%-1\%-5.4\%, \text{ иначе}$ $z=1345.5*t-(s+k-75900)*12\%-1\%,$ <p>где z - зарплата; s - оклад; k - уральский коэффициент.</p>																	
8	<p>Банковский процент зависит от величины вклада следующим образом:</p> <table><tr><td>Процент</td><td>10%</td><td>12%</td><td>14%</td><td>17%</td><td>20%</td></tr><tr><td>Вклад</td><td>до 10000</td><td>от 10000 до 20000</td><td>от 20000 до 30000</td><td>от 30000 до 40000</td><td>свыше 40000</td></tr></table>						Процент	10%	12%	14%	17%	20%	Вклад	до 10000	от 10000 до 20000	от 20000 до 30000	от 30000 до 40000	свыше 40000
Процент	10%	12%	14%	17%	20%													
Вклад	до 10000	от 10000 до 20000	от 20000 до 30000	от 30000 до 40000	свыше 40000													

9	<p>С помощью мультипликатора Кейнса рассчитать величину инвестиций в данный период.</p> $k=1/(1-mc), \text{ если } mc=1, \text{ то } k=1/ms$ $I=K/MS,$ <p>где</p> <p>I - инвестиции;</p> <p>ms - склонность к сбережениям:</p> $ms = (s_2-s_1)/(y_2-y_1), (s_2-s_1) - \text{изменение величины сбережений};$ <p>y<sub>2</sub>-y<sub>1</sub> - изменение величины дохода;</p> <p>k - мультипликатор Кейнса: <math>k = 1/ms = 1/(1-mc)</math>;</p> <p>mc - склонность к потреблению: <math>mc = (c_2-c_1)/(y_2-y_1)</math>;</p> <p>(c<sub>2</sub>-c<sub>1</sub>) - изменение расходов на потребление.</p>
10	<p>Рассчитать годовой экономический эффект G,</p> $G = [(S_1+E*K_1)-(S_2+E*K_2)]*O,$ <p>если цена реализации единицы продукции в новом варианте - P<sub>2</sub> равна цене реализации единицы продукции в базисном варианте - P<sub>1</sub>, иначе</p> $G = [(S_1+E*K_1)-(S_2+E*K_2)+(P_2-P_1)]*O,$ <p>где S<sub>1</sub>, S<sub>2</sub> - себестоимость единицы продукции по базисному и новому варианту;</p> <p>K<sub>1</sub>, K<sub>2</sub> - удельные капитальные вложения в базисном и новом варианте;</p> <p>E - нормативный коэффициент эффективности капитальных вложений;</p> <p>O - объём применений результатов научных исследований.</p>
11	<p>Определить цену продажи 10000 единиц продукции, если известны затраты на их производство; причем, если себестоимость 1 единицы &lt; 600 рублей, то надбавка составит 300 рублей, если себестоимость 1 единицы продукции будет ≥ 600 рублей, то продавать будут на 400 рублей дороже.</p> <p>Формула:</p> <p>себестоимость 1 единицы = общая сумма затрат/кол-во единиц,</p> <p>где S - себестоимость 1 единицы;</p> <p>Z - затраты;</p> <p>Z<sub>p</sub> - цена продажи 1 единицы.</p>
12	<p>Некоторым категориям жильцов положена скидка при оплате квартплаты: инвалидам – 70%, участникам войны – 50%, многодетным – 25%. Номинальная квартплата известна. Вычислить квартплату.</p>
13	<p>На фирме производство характеризуется следующими показателями а = 167, р = 12, б = 1654, с<sub>1</sub> = 165, т = 154, ц = 176 натуральных единиц. Однако произошло сокращение производства. Если к<sub>1</sub> меньше 1, тогда прибыль рассчитать по формуле:</p> $p = ц - ((a + c_1 + t) / (б * k_1)),$ <p>в противном случае правительством повышаются нормы амортизованных списаний, следовательно прибыль рассчитывается по формуле:</p> $p = ц - ((a * p + c_1 + t * k_1) / (k_1 * б)).$

Продолжение таблицы 17

14	<p>Выполнить расчеты по страхованию дома от пожара на садовом участке. Страховой взнос равен 3% от стоимости кирпичного дома. Если дом деревянный, страховка составляет 5% при наличии в непосредственной близости (на расстоянии до 20 м) водоема. Если ближайший водоем находится на расстоянии до 400 м – страховка деревянного дома увеличивается до 7%, если до 2000 м – до 10%, если более – до 15%.</p>
15	<p>На фирме производство характеризуется следующими показателями:  С - цена единицы продукции, <math>C=100000</math>;  А - амортизационные отчисления, <math>A=200</math>;  з.пост - постоянные затраты, <math>з.пост = 50</math>;  з.пер - переменные затраты, <math>з.пер = 100</math>;  В - выпуск продукции, <math>V = 100</math>.  Рассчитать прибыль производства П, если при коэффициенте сокращения производства <math>k=0.7</math>  <math>П = C-(A+з.пост+з.пер*k)/k*V</math>, иначе  <math>П = C-(k*A+з.пост+0.7*з.пер)/0.7*V</math></p>
16	<p>Определить минимальную стоимость билета для пассажира, следующего на заданное расстояние, стоимость билета с учетом выбранного вида вагона и поезда. Перемещение на скором поезде увеличивает стоимость билета на 20%, размещение в плацкартном вагоне – на 25%, а в купейном – на 35%. Далее следует установить окончательную стоимость билета. Его стоимость может быть снижена, если пассажир следует на дальнее расстояние. При длине маршрута от 500 до 1500 км скидка составляет 5% от полной стоимости билета, при расстоянии до 5000 км – 7%, если расстояние больше – 10%.</p>
17	<p>Найти значение кредита в рублях, который может выдать банк заемщику. Для ежемесячного дохода от \$100 кредит может составить 1000% от заработка, для дохода от \$500 – 3000%, для дохода от \$1000 – 6000% и от \$3000 – 7000%.</p>
18	<p>Расчет с постоянными клиентами-покупателями, которым товар отпускается в кредит. Вычислить величину кредита, которая зависит от его длительности. При сроке свыше 5-и дней – это 2% от исходной стоимости, свыше 10-и – 4%, свыше 15-и и еще по 1% за каждый день после 15-го. При задержке в оплате свыше 20-го дня должно появляться слово «Звонить».</p>
19	<p>Расчет страхования имущества от кражи, пожара и протечки. Клиент может застраховаться на любую Страховую сумму, для чего делает Страховой взнос в размере 10% от страховой суммы, если имущество страхуется от пожара, 8% - если от протечки, 7% - от кражи. Страховка от кражи снижается на 1% при наличии стальной двери и еще на 2%, если в подъезде имеется вахтер.</p>

## Продолжение таблицы 17

20	Расчет страхования жизни. Клиент может застраховаться на любую Страховую сумму, для чего делает Взнос в размере, зависящем от возраста застрахованного (до года – 10% от страховой суммы, от года до пяти – 8%, от пяти до 20 лет – 5%, от 20 до 45 лет -3%, от 45 до 50 лет – 5%, от 50 до 65 лет – 8%. Лица старше 65 лет вносят 10% и еще по одному проценту за каждый год после 65-ти лет.
----	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 2.9 Оператор выбора (переключатель)

С помощью оператора выбора VBA можно выбрать вариант из любого количества вариантов. Параметром, по которому осуществляется выбор, служит ключ выбора.

Структура оператора выбора:

```

Select Case <ключ_выбора>
    Case Значение_1
        <Оператор1>
    Case Значение_2
        <Оператор2>
    .....
    Case Значение_n (константа)
        <Оператор n>
End Select

```

список  
выбора

где

Select, Case, End - зарезервированные слова VBA (выбирать, случай, конец);

<ключ\_выбора> - переключатель варианта (некоторое выражение);

<список\_выбора> - одна или более конструкций вида: <константа\_выбора>  
<оператор>;

Значение\_n - <константа\_выбора> того же типа, что и выражение <ключ\_выбора>;

<операторы> - произвольные операторы VBA, выполняемые при совпадении Значения\_n и значения ключа\_выбора.

Работа оператора выбора.

Вначале вычисляется значение выражения <ключ\_выбора>, а затем в последовательности операторов <список\_выбора> отыскивается такой, которому предшествует константа, равная вычисленному значению. Найденный оператор выполняется, после чего оператор выбора завершает свою работу. Если в списке выбора не будет найдена константа, соответствующая вычисленному значению ключа выбора, управление передается операторам, стоящим за End Select.

В операторе Select Case можно использовать операции отношения. Для этого надо включить в выражение ключевое слово Is или ключевое слово To.

Ключевое слово Is дает указание компилятору сравнить значение проверяемой переменной со значением выражения, следующего за ключевым словом Is.

Ключевое слово To задает диапазон значений.

Пример 1.

Постановка задачи. Составить пользовательскую подпрограмму процедуру, расшифровывающую количество баллов за полученный ответ по введенному числу с клавиатуры.

Технология выполнения задания:

1 Исходные данные:

$x \in \mathbb{Z}$ .

2 Набрать в стандартном модуле проекта следующую пользовательскую процедуру:

```
Sub DemoSelect ()  
Dim x As Integer  
x = InputBox ("введите целое число")  
Select Case x  
Case 8 To 10  
MsgBox ("Отлично")  
Case 6 To 7  
MsgBox ("Хорошо")  
Case 4 To 5  
MsgBox ("Удовлетворительно")  
Case Is < 4  
MsgBox ("Неудовлетворительно")  
End Select  
End Sub
```

3 Запустить подпрограмму-процедуру.

Примечание. Процедура MsgBox выводит на экран диалоговое окно, содержащее сообщение для пользователя.

### Лабораторная работа № 3

Программирование алгоритмов ветвлений со многими вариантами

Цель: приобретение навыков программирования алгоритмов ветвлений со многими вариантами с помощью пользовательской подпрограммы-функции, позволяющей выбрать необходимый вариант из любого предложенного количества вариантов. Индивидуальные варианты лабораторной работы № 3 представлены в таблице 18 (2 часа)

Таблица 18

<i>№ В</i>	<i>Варианты индивидуальных заданий</i>
	Разработать функцию пользователя, позволяющую выбрать вариант из любого количества вариантов

1	С клавиатуры вводятся цифры (от 1 до 7). Составить программу, выводящую на экран название дня недели соответственно введенной цифре. Если введенная цифра меньше 1 или больше 7, то выводится надпись «нет соответствия».
---	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Продолжение таблицы 18

2	С клавиатуры вводятся числа (от 1 до 12). Составить программу, выводящую на экран название месяца соответственно введенному числу. Если число меньше 1 или больше 8, то выводится надпись "нет соответствия".
3	С клавиатуры вводится число ( $n > 0$ ). Составить программу, выводящую на экран надпись: "...эти порядки существовали n лет". Слово лет должно изменяться на год, года в соответствии с правилами русского языка.
4	С клавиатуры вводится числа (от 1 до 25). Составить программу, выводящую на экран название возрастной категории соответствующей данному числу: 0-1 грудной возраст 2-3 ясельный 4-6 дошкольный 7-10 младший школьный 11-16 подростковый 17 допризывной 18 призывной 25 студенческий.
5	С клавиатуры вводятся числа (от 0 до 99). Составить программу, выводящую на экран название региона России, соответствующее заданному числу: 56 – Оренбургская область; 77 – Московская обл. и т.д. Если число отсутствует в списке, то вывести надпись “нет данных”.
6	С клавиатуры вводятся числа (от 0 до 99). Составить программу, выводящую на экран название страны по международной классификации (штрих-код): 01-03 – США + Канада, 46 – Россия и т.д.
7	Составить программу определения названия специальности Финансово-экономического факультета по ее числовому коду. Например, 351200 – Налоги и налогообложение и т.д.
8	Составить программу, выводящую на экран характеристики компьютеров относящихся к n-ому поколению. Число n вводится при помощи клавиатуры.
9	Составить программу, выводящую на экран сроки выполнения лабораторной работы в зависимости от её номера.
10	С клавиатуры вводятся числа (от 0 до 10). Составить программу, выводящую на экран английские названия, соответствующие вводимому числу.

11	Составить программу, сопоставляющую арабским цифрам – римские. Предусмотреть вариант отсутствия соответствия.
12	Составить программу, выводющую на экран названия знаков зодиака, в соответствии с вводимым номером месяца. Если вводимое число не принадлежит диапазону 0 – 12, то вывести надпись “нет знака”.

#### Продолжение таблицы 18

13	Расположить районы Оренбургской области в алфавитном порядке и пронумеровать их. Составить программу, выводющую на экран название района, в зависимости от введенного числа.
14	Написать программу, определяющую время суток по вводимому с клавиатуры часу, 8 – утро, 21 – вечер и т.д.
15	Для числа k напечатать фразу: “Мы нашли k грибов” согласовав окончание слова “гриб” с числом k.
16	Составить программу, выводющую на экран названия телеканалов. 1 – ОРТ, 2 – РТР и т.д.
17	Составить программу, выводющую название ноты в соответствии с ее порядковым номером.
18	Составить программу, выводющую на экран номера телефонов экстренного вызова 01, 02 и т.д.
19	Напишите программу-модель анализа пожарного датчика в помещении, которая выводит сообщение «пожароопасная ситуация», если температура (в вашей модели она будет вводиться с клавиатуры) в комнате превысила 600 °С.
20	Составить программу, выводющую расшифровку телефонных кодов городов России в зависимости от введенного числа n.

## 2.10 Операторы цикла

В VBA существуют два основных типа циклов – циклы со счетчиком (параметрические) и циклы с условием (итерационные).

Циклы со счетчиком используют в тех случаях, когда необходимо выполнить некоторые действия определенное число раз

Циклы с условием применяются тогда, когда некоторые действия в программе должны повторяться до тех пор, пока выполняется определенное условие.

### 2.10.1 Циклы с параметром For ... Next

Структура цикла:

For Параметр\_Цикла = Начальное\_Значение To Конечное\_Значение  
[Step Шаг]

Операторы

[Exit For]

Next [Параметр\_Цикла]

где For – ключевое слово VBA (от), обозначающее начало цикла;  
Параметр\_цикла – переменная, определенная в качестве счетчика цикла;  
Начальное\_Значение – число, задающее начальное значение параметра цикла;  
То – ключевое слово VBA (до), разделяющее  
Начальное\_значение и Конечное\_Знаение;  
Конечное\_Значение – число, задающее значение параметра цикла,  
при котором цикл завершается;  
Step – ключевое слово VBA (шаг), используемое для  
задания шага цикла, необязательный аргумент;  
Шаг – число, задающее шаг цикла, т.е. значение, на которое  
увеличивается (или уменьшается) значение параметра  
цикла на каждом шаге. Это число может быть  
отрицательным;  
Exit For – оператор досрочного выхода из цикла (необязательный);  
Next – ключевое слово VBA (следующий), обозначающее  
конец цикла.

Работа цикла:

Шаг 1 Сначала определяется параметр цикла, а также вычисляются и запоминаются начальное и конечное значения этой переменной.

Шаг 2 Параметру цикла присваивается начальное значение.

Шаг 3 Начальное значение параметра цикла сравнивается с конечным значением.

Если параметр цикла больше конечного значения, программа сразу выходит из цикла и переходит к той строке кода, которая следует за циклом.

Шаг 4 Выполняется тело цикла.

Шаг 5 После выполнения тела цикла происходит присваивание параметру цикла следующего значения. Переход к шагу 3.

Примечание.

1 Если используется ключевое слово Step, то параметр цикла изменяется в соответствии с числом, указанным после этого слова. Если слово Step отсутствует, то значение шага равно единице.

Пример 1.

For I = 0 To 10 Step 2 (Значение I будет увеличиваться на 2)

2 Цикл For ... Next может быть прерван досрочно при достижении какого либо условия. Для этого в нужном месте цикла нужно поместить оператор Exit For.

Пример 2.

Dim S As Integer

Dim j As Integer

S = 2

For j = 1 To 10

S = S + j

If S > 6 Then

Exit For (Выход из цикла, если значение S > 6)

End If

Next j



MsgBox (S)

## 2.10.2 Циклы с условием (итерационные)

Если какое-то действие (несколько действий) необходимо выполнить много раз, но заранее неизвестно сколько раз и это зависит от какого-то условия, то тогда следует воспользоваться циклом с предусловием или с постусловием.

В VBA есть два основных цикла **DO ... LOOP** – с условием, вводимым ключевым словом **While**, и с условием, вводимым ключевым словом **Until**. Оба они могут быть с предусловием или с постусловием.

Синтаксис:

Цикл с предусловием	Цикл с постусловием
Do While   Until <условие> <Тело цикла> Loop	Do <Тело цикла> Loop While   Until <условие>

где            **Do** – ключевое слово (делать);  
                **While** – ключевое слово (пока);  
                **Until** – ключевое слово (до тех пор пока);  
                **Loop** – ключевое слово, указывающее на окончание цикла;  
                <условие> – логическое выражение, истинность которого проверяется  
                                в начале каждого выполнения тела цикла;  
                <тело\_цикла> – произвольная последовательность операторов;

Конструкция **Do ... While** читается: делать пока выполняется условие. В конструкции **Do ... While** для увеличения шага следует писать специальный оператор, т.к. в ней в отличие от конструкции **For**, не делается это автоматически.

Конструкция **Do ... Until** читается: делать до тех пор, пока не будет выполнено условие. В конструкции **Do ... Until** для увеличения шага следует писать специальный оператор, т.к. в ней в отличие от конструкции **For**, не делается это автоматически.

Условие, записанное после ключевого слова **Until**, проверяется в конце каждой итерации (после выполнения тела цикла). Обратите внимание, что оно работает здесь не совсем так, как в цикле **While**. Если условие является истинным (**True**), то выполнение цикла завершается. Если условие не выполняется (является ложным – **False**), то вновь выполняется тело цикла.

Пример 1.

Постановка задачи. Вычислить сумму конечного ряда  $\sum_{i=1}^{10} i^2$  с помощью подпрограммы-процедуры.

Технология выполнения задания:

1 Исходные данные:  $i \in \mathbb{Z}$

Результат:  $S \in \mathbb{R}$ .

2 Набрать в стандартном модуле проекта следующую пользовательскую процедуру, используя цикл с предусловием **While**:

Sub summa()

Dim S As Integer

```

Dim i As Integer
S = 0
i = 1
Do While i <= 10
S = S + i ^ 2
i = i + 1
Loop
MsgBox (S)
End Sub

```

3 Набрать в стандартном модуле проекта следующую пользовательскую процедуру, используя цикл с предусловием Until:

```

Sub summa()
Dim S As Integer
Dim i As Integer
S = 0
i = 1
Do Until i > 10
S = S + i ^ 2
i = i + 1
Loop
MsgBox (S)
End Sub

```

4 Набрать в стандартном модуле проекта следующую пользовательскую процедуру, используя цикл с постусловием While:

```

Sub summa()
Dim S As Integer
Dim i As Integer
S = 0
i = 1
Do
S = S + i ^ 2
i = i + 1
Loop While i <= 10
MsgBox (S)
End Sub

```

5 Набрать в стандартном модуле проекта следующую пользовательскую процедуру, используя цикл с постусловием Until:

```

Sub summa()
Dim S As Integer
Dim i As Integer
S = 0
i = 1
Do
S = S + i ^ 2
i = i + 1

```

```

Loop Until i > 10
MsgBox (S)
End Sub

```

#### Лабораторная работа № 4

##### Программирование алгоритмов циклической структуры

Цель: приобретение навыков программирования алгоритмов циклической структуры с помощью подпрограммы-процедуры, позволяющую вычислять сумму (произведение) конечного ряда с помощью операторов цикла с параметром, с предусловием, с постусловием. Индивидуальные варианты лабораторной работы № 4 представлены в таблицах 19, 20 (2 часа).

Таблица 19

<b>№ В</b>	<b>Варианты индивидуальных заданий</b>
Разработать подпрограмму процедуру, вычисляющую сумму (произведение) конечного ряда	
1	$\sum_{i=1}^{25} \frac{x^i}{i}$
2	$\sum_{i=1}^{10} \left( \frac{1}{i} + \sqrt{ x } \right)$
3	$\sum_{i=1}^{10} \frac{x + \cos(ix)}{2^i}$
4	$\prod_{k=1}^{10} \left( 1 + \frac{\sin(kx)}{k} \right)$
5	$\prod_{k=1}^{30} \left( \frac{1}{k+1} - \cos^k x \right)$
6	$\prod_{k=1}^{20} \frac{(1-x)^{k+1} + 1}{((k-1)+1)}.$
7	$\sum_{k=1}^{30} \frac{1}{k}$
8	$\sum_{k=1}^{15} \frac{1}{k^5}$

9	$\sum_{k=1}^{10} \frac{1}{(2k+1)^2}$
10	$\sum_{k=1}^{35} \frac{(-1)^k}{(2k+1)k}$

Продолжение таблицы 19

11	$\prod_{i=2}^{10} \left(1 - \frac{1}{i}\right)^2$
12	$\sum_{i=1}^{100} \frac{1}{i^2}$
13	$\sum_{k=1}^{50} \frac{(-1)^{k+1}}{k(k+1)}$
14	$\sum_{k=0}^{10} \frac{(-1)^k (k+1)}{k}$
15	$\sum_{k=1}^{10} \frac{k}{\frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{k+1}}$
16	$\prod_{i=1}^{52} \frac{i^2}{i^2 + 2i + 3}$
17	$\prod_{i=1}^{23} \left(2 + \frac{1}{i}\right)$
18	$\prod_{i=2}^{40} \frac{i+1}{i+2}$
19	$\sum_{i=1}^{50} \frac{1}{i^5}$
20	$\sum_{i=1}^{10} \frac{1}{i}$

Таблица 20

<b>№ В</b>	<b>Варианты индивидуальных заданий</b>
	Разработать экономическую пользовательскую процедуру

1	<p>Вычислить 5 значений уровня цен на автомобили на рынке, если объем денежной массы остается неизменным.</p> $P=M*V/Q$ <p>где  P - уровень цен;  M - объем денежной массы;  V - скорость обращения денег;  Q - количество обращающегося товара.</p>
---	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Продолжение таблицы 20

2	<p>Мы положили в банк 10000 р. на 5 лет при ставке 10% в год. Текущая стоимость составляет 10000 р. Вычислить ежегодную прибыль</p> $FV=PV*(1+R)^N$ <p>где  FV - прибыль;  PV - текущая стоимость;  R - ставка процента;  N - число лет.</p>
3	<p>Рассчитать реальную заработную плату VP по формуле <math>VP=VN/P</math>. Если уровень цен P меняется от P1 до P2, шаг изменения уровня цен = 1.</p> <p>где  P1 - начальный уровень цен;  P2 - конечный уровень цен;  VP - заработная плата;  VN - номинальная заработная плата;  P - уровень цен.</p>
4	<p>Рассчитать какой доход получит владелец акции, купив ее по рыночной цене P, если рыночная цена изменяется на 300 денежных единиц каждый год, а дивиденд на акцию фиксирован. Доход на акцию рассчитывается по формуле:</p> $\text{ДОХОД} = \text{ДИВИДЕНД} * 100 / \text{РЫНОЧНАЯ ЦЕНА АКЦИИ}$
5	<p>Начислить сумму износа по объекту основных средств за 6 месяцев, если дана первоначальная стоимость объекта, годовая норма в %, используя формулу:</p> $\text{износ за 1 месяц} = \text{первонач.стоимость} * \text{норму годовую} / 12 \text{ мес}$ <p>т.е. <math>I = S * N / 12</math></p> <p>где  I - сумма износа за месяц;  S - первоначальная стоимость;  N - норма годовая (8.9%).</p>
6	<p>Оформлен депозитный вклад в банке на сумму 5 млн.р. при ставке (R) 10% в год. Составить программу, вычисляющую будущую стоимость вклада (FV), если известно, что текущая стоимость (PV) составляет 10 млн. р., а число лет, на которые оформлен вклад равно 5.</p>

7	В 1626 г индейцы продали о.Манхеттен за 20\$. Если бы эти деньги были помещены в банк под 4% в год, то какова была бы стоимость капитала в 1996 г?
8	<p>Выявление зависимости экономической ситуации от величины Спроса (С) и Предложения (Р), по формуле:</p> $C = F \cdot Q, P = F \cdot S$ <p>где</p> <p>F - коэффициент функциональной зависимости; Q, S – цена.</p>

Продолжение таблицы 20

9	<p>Рассчитать темпы роста безработицы с А года по В год и в целом за данный период по формулам:</p> $C = (D - E) / D \cdot 100, F = D - J / D \cdot 100$ <p>где</p> <p>С - темп роста безработицы в А+I+1 году; D - индекс безработицы в А+I+1 году; Е - индекс безработицы в А+I году; F - на сколько выросла безработица в период с А по В годы, где А = 1990 год, В = 1993 год.</p>
10	<p>В фирме работают 20 человек, каждый из них получает разную зарплату: 5 из них получают 800.000 рублей, 7 из них получают 500.000 рублей, 8 из них получают 300.000 рублей, нужно вычислить среднюю зарплату всех рабочих, по формуле:</p> $P = K / N$ <p>где</p> <p>P - средняя зарплата рабочих, K - количество получаемых денег; N - количество людей получающих деньги.</p>
11	<p>Определить рыночную стоимость облигаций по формуле</p> $PV = \text{сумма } (I = 1.. N) P / (1 + R)$ <p>где</p> <p>PV - рыночная стоимость облигации; P – доход; R - процент от облигаций; N - количество лет.</p>
12	<p>Как будет меняться прибыль предприятия при изменении цены за единицу продукции, если объём выпускаемой продукции и себестоимость за единицу продукции останутся неизменными:</p> <p>объём выпуска - 1500 ед.; себестоимость за единицу продукции – 1004; начальное значение цены – 140; конечное значение цены – 200; шаг изменения цены – 20.</p>

13	<p>Вычислить среднемесячный уровень безработицы в течении года ,используя формулу:</p> $\text{уровень безработицы} = \frac{\text{среднемесячная численность безработных}}{\text{численность рабочей силы}}$ <p>Учитывая, что численность рабочей силы растет, увеличиваясь каждый раз на 2 процента.</p>
14	<p>Пенсионерка положила в банк в январе 1995 года 500 тыс.р. Сколько она возьмет через полгода (18 месяцев), если банк начисляет 5 процентов в месяц?</p>

Продолжение таблицы 20

15	<p>Вычислить доход на акцию предприятия, учитывая, что дивиденд на акцию изменяется в течение некоторого времени с <math>[DN, DK]</math> с шагом <math>H</math>.</p> $Y = ((D \cdot D + 1) / D) \cdot 100\%$ <p>где D – дивиденд.</p>
16	<p>Вычислить среднегодовую стоимость (<math>C_c</math>) оборудования за период с момента окончания планового срока лизингового соглашения до полной амортизации оборудования на начало года по формуле</p> $C_c = (C_n + C_k) / 2 \cdot T$ <p>где  <math>C_n</math> - стоимость оборудования на начало года;  <math>C_k</math> - стоимость оборудования на конец года;  <math>T</math> - период с момента окончания планового срока лизингового соглашения до полной амортизации оборудования.</p>
17	<p>Цех располагает 50 ткацкими станками, из которых 30 работают в течение 20 лет. 12 в течение 6 лет, 8 в течение 6 лет. Вычислить средний возраст оборудования.</p>
18	<p>Вычислить процентные деньги с <math>N</math> рублей за <math>t</math> дней по формуле:</p> $A = N \cdot P\% \cdot t / 100\% \cdot 360$ <p>где  <math>A</math> - процентные деньги;  <math>t</math> - число дней;  <math>P</math> - ставка процента.</p>
19	<p>Списан автомобиль грузоподъемностью до 2 тонн. Первоначальная стоимость автомобиля составляла 10000 рублей, норма амортизации за месяц составляла 1.4% в месяц. Вычислить сумму амортизации на автомобиль за любой месяц первых пяти лет по формуле:</p> $S = P \cdot N \cdot K / 100\%$ <p>где  <math>K</math> - количество месяцев;  <math>P</math> - цена машины;  <math>N</math> - норма амортизации;  <math>S</math> - сумма амортизации.</p>

20	<p>Вычислить средний возраст оборудования, вычисляемый по отношению общей суммы лет фактической эксплуатации всех машин <math>\sum t</math> к количеству единиц этого оборудования <math>n</math>, где <math>t</math>-средний возраст оборудования, лет.</p>
----	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

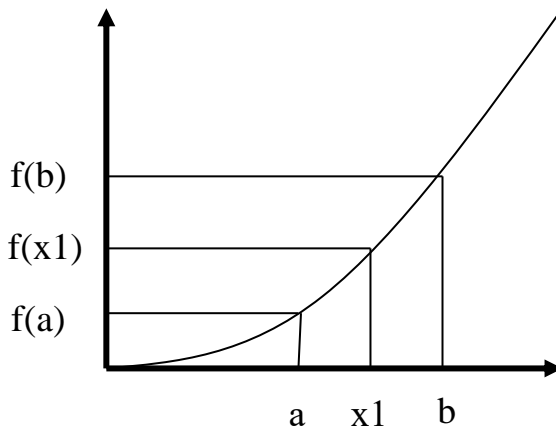


## 2.11 Табулирование функции

### 2.11.1 Табулирование функции одной переменной

Постановка задачи. На отрезке  $[a, b]$  с шагом  $h$  протабулировать функцию  $f(x)=x^2+\ln|x|$  и вывести на печать значения этой функции.

Технология выполнения задания:



Для решения задачи в области определения функции вводится сетка – совокупность точек  $\{x_0, x_1, x_2, \dots, x_i\}$ , где  $x_0 = a$ ,  $x_1 = x_0 + h$ ,  $\dots$ ,  $x_i = x_{i-1} + h$ ,  $x_n = b$ . Тогда задача сводится к вычислению функции  $f$  в точках  $x_i$ , то есть нужно вычислить  $f(x_0)$ ,  $f(x_1)$ ,  $\dots$ ,  $f(x_i)$ ,  $f(x_n)$ .

I вариант

1 Введем обозначения:

$x_n, x_k$  – границы отрезка;

$h$  – шаг;

$x$  – точка;

$f$  – значение функции в точке.

2 Набрать в стандартном модуле проекта следующую пользовательскую процедуру, используя цикл For:

```
Sub primer()  
Dim x As Single  
Dim xn As Single  
Dim xk As Single  
Dim h As Single  
Dim f As Single  
xn = InputBox("xn=", xn)  
xk = InputBox("xk=", xk)  
h = InputBox("h=", h)  
For x = xn To xk Step h  
f = x ^ 2 + Log(Abs(x))  
f = Format(f, "##.####")  
x = Format(x, "##.###")  
Debug.Print " x= " & x & " f= " & f  
Next x  
End Sub
```

II вариант

1 Введем обозначения:

$x_n, x_k$  – границы отрезка;

$h$  – шаг;

$x$  – точка;

$f$  – значение функции в точке .

2 Набрать в стандартном модуле проекта следующую пользовательскую процедуру, используя итерационный цикл:

```
Sub primer ()
```

```
Dim x As Single
```

```
Dim xn As Single
```

```
Dim xk As Single
```

```
Dim f As Single
```

```
Dim h As Single
```

```
xn = InputBox("xn=", xn)
```

```
xk = InputBox("xk=", xk)
```

```
h = InputBox("h=", h)
```

```
x = xn
```

```
Do
```

```
f = x^2+log(ABS(x))
```

```
Debug.Print "x= " & x & " f= " & f
```

```
x = x + h
```

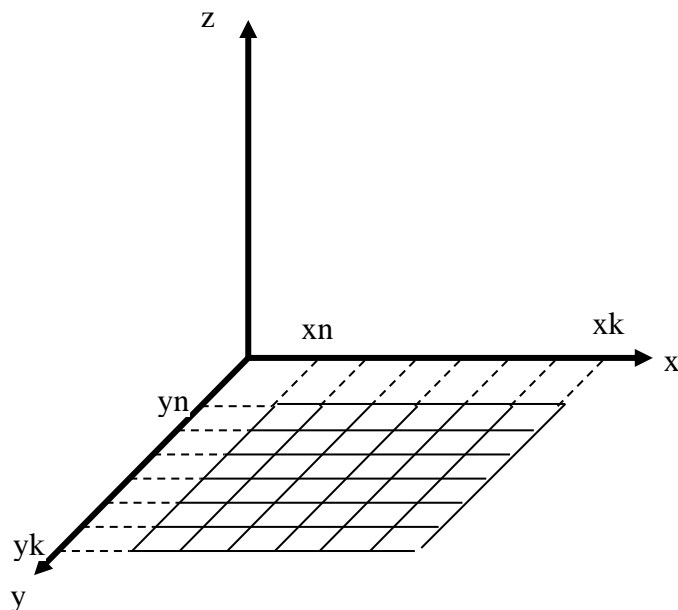
```
Loop While x <= xk
```

```
End Sub
```

### 2.11.2 Табулирование функции двух переменных (вложенные циклы)

Постановка задачи. На интервале  $x_n \leq x \leq x_k$  с шагом  $\Delta x$ , для каждого  $y$  из интервала  $y_n \leq y \leq y_k$  с шагом  $\Delta y$  вычислить значение функции  $z = x^2 + \ln|y|$ .

Технология выполнения задания:



Для решения задачи в области определения функции вводится разностная сетка  $\{x_i, y_i\}$ , где  $x_0 = x_n$ ,  $x_1 = x_0 + h_x$ , ...,  $x_i = x_{i-1} + h_x$ , ...,  $x_n = x_k$ ;  $y_0 = y_n$ ,  $y_1 = y_0 + h_y$ , ...,  $y_j = y_{j-1} + h_y$ , ...,  $y_m = y_k$ .

$n = \frac{x_k - x_n}{h_x}$  - количество узлов сетки по оси  $x$ ,

$m = \frac{y_k - y_n}{h_y}$  - количество узлов сетки по оси  $y$ .

Тогда задача сводится к вычислению функции  $z$  в точках  $x_i$  и  $y_j$ , то есть  $z(x_i, y_j)$ , то есть  $z(x_i, y_j) = x_i^2 + \ln|y_j|$ , где  $i=1, 2, \dots, n$ ,  $j=1, 2, \dots, m$ .

I вариант:

1 Введем обозначения:

$x, y$  – координаты узла сетки;

$f$  – значение функции в нем;

$x_n, x_k, y_n, y_k$  – границы отрезка;

$h_x, h_y$  – шаги сетки.

2 Набрать в стандартном модуле проекта следующую пользовательскую процедуру, используя цикл For:

```
Sub primer()
```

```
Dim x As Single
```

```
Dim xn As Single
```

```
Dim xk As Single
```

```
Dim yn As Single
```

```
Dim yk As Single
```

```
Dim f As Single
```

```
Dim hx As Single
```

```
Dim hy As Single
```

```
xn = InputBox("xn=", xn)
```

```
xk = InputBox("xk=", xk)
```

```
yn = InputBox("yn=", yn)
```

```
yk = InputBox("yk=", yk)
```

```
hx = InputBox("hx=", hx)
```

```
hy = InputBox("hy=", hy)
```

```
For y = yn To yk Step hy
```

```
For x = xn To xk Step hx
```

```
f = x^2 + log(ABS(y))
```

```
Debug.Print "x= " & x & " y= " & y & " f= " & f
```

```
Next x
```

```
Next y
```

```
End Sub
```

II вариант

1 Введем обозначения:

$x, y$  – координаты узла сетки;

$f$  – значение функции в нем;

$x_n, x_k, y_n, y_k$  – границы отрезка;

hx,hy – шаги сетки.

2 Набрать в стандартном модуле проекта следующую пользовательскую процедуру, используя итерационный цикл:

```
Sub primer ()
Dim x As Single
Dim xn As Single
Dim xk As Single
Dim yn As Single
Dim yk As Single
Dim f As Single
Dim hx As Single
Dim hy As Single
xn = InputBox("xn=", xn)
xk = InputBox("xk=", xk)
yn = InputBox("yn=", yn)
yk = InputBox("yk=", yk)
hx = InputBox("hx=", hx)
hy = InputBox("hy=", hy)
y = yn
Do
x = xn
Do While x <= xk
f = x^2+log(ABS(x))
Debug.Print "x= " & x & " y= " & y & " f= " & f
x = x + hx
Loop
y = y + hy
Loop Until y > yk
End Sub
```

#### Лабораторная работа № 5

##### Табулирование функции

Цель: приобретение навыков программирования вычисления значений функции вида  $y=f(x)$  на промежутке  $[a,b]$  с шагом  $h$  и  $z=f(x,y)$  на промежутке  $[a,b]$  и  $[c,d]$  с шагом  $h_x$  и  $h_y$  с помощью пользовательской подпрограммы-процедуры. Индивидуальные варианты лабораторной работы № 5 представлены в таблице 21 (2 часа).

Таблица 21

№ В	Варианты индивидуальных заданий
Разработать пользовательскую процедуру вычисления значения выражения	
1	<p>Вычислить 7 значений функций</p> $y = \frac{0.15x + a}{a^2 + x^2},$ <p>где начальное значение <math>a=0.14x</math>, шаг <math>\Delta a=0.22</math></p>

Продолжение таблицы 21

2	<p>На интервале <math>0.1 \leq a \leq 0.5</math> и <math>0.1 \leq b \leq 0.5</math> с шагом <math>\Delta a = 0.1</math> и <math>\Delta b = 0.05</math> протабулировать функцию</p> $c = b * \arcsin(ax)$
3	<p>Для каждого <math>0.1 \leq a \leq 0.2</math> при всех <math>0.5 \leq b \leq 0.8</math> с шагом <math>\Delta b = 0.1</math> вычислить</p> $c = \frac{ax^2 + bx + \ln a}{x + e^{-(a+b)}} + x^2,$ <p><math>\Delta a = 0.01, x=0.792</math></p>
4	<p>На интервале <math>\frac{x^2 - 1}{x} \leq y \leq \frac{x^2 + 2x + 1}{x - 0.2}</math> с шагом <math>\Delta y = \frac{0.01}{x^3}</math> протабулировать функцию <math>z = \ln y + e^{-y}</math></p>
5	<p>На интервале <math>0.011 \leq a \leq 0.874</math> с шагом <math>\Delta a = 0.312</math> для каждого «b» из интервала <math>0.014 \leq b \leq 0.021</math> с шагом <math>\Delta b = 0.001</math> протабулировать функцию</p> $c = \arccos \frac{a + b}{x}$
6	<p>Для каждого <math>0.16 \leq y \leq 1.6</math> с шагом <math>\Delta y = 0.24</math> при всех <math>0.1 \leq z \leq 0.5</math> вычислить <math>s = (yx^2 + z) \cdot \arcsin z</math>, причём, если <math>y &lt; 0.8</math>, <math>\Delta z = 0.1</math>, иначе <math>\Delta z = 0.05</math></p>
7	<p>На интервале <math>0.5 \leq y \leq 1</math> с шагом <math>\Delta y = 0.01</math> определить количество отрицательных значений функции <math>p = ay^2 - b</math>, где <math>a=0.8x</math>, <math>b=2.2x</math></p>
8	<p>Для каждого <math>0.116 \leq b1 \leq 1.185</math> с шагом <math>\Delta b = 0.225</math> вычислить по 6 значений функции <math>c = \ln b1 + x \ln b2</math>, где начальное значение <math>b2 = 2.146</math>, а шаг <math>\Delta b2 = -0.231</math></p>
9	<p>На интервале <math>0 \leq a \leq 5</math> протабулировать функцию <math>b = 1 - e^{-a/x}</math>, причём шаг <math>\Delta a = \begin{cases} 1, \sin c &gt; 1 \\ 0.5, \text{ в остальных случаях} \end{cases}</math>, где <math>c = \sin(x^2 - a\sqrt{x})</math>, <math>x=0.44</math></p>
10	<p>Для каждого <math>\pi - 1 \leq a \leq 2\pi</math> с шагом <math>\Delta a = 1.8</math> при всех <math>0.5 \leq b \leq 0.9</math>, вычислить</p> $c = \frac{ax^2 + bx + 0.16}{\sqrt{a + b + x^2}},$ <p>причём, если <math>\frac{ax - b}{c} &gt; 0</math>, то <math>\Delta b = 0.1</math>, иначе <math>\Delta b = 0.05</math>, <math>x=0.087</math></p>

Продолжение таблицы 21

11	<p>На интервале <math>0.15 \leq y \leq 2.83</math> с шагом <math>\Delta y = 0.274</math> протабулировать функцию</p> $z = \begin{cases} a \cdot x \cdot y, & y < b - a \\ b \cdot x^2 - y, & b > y \geq a \\ \text{не определено в остальных случаях} \end{cases}$ <p>где <math>x=0.881</math>, <math>a=0.96</math>, <math>b=1.44</math></p>
12	<p>Для каждого <math>0.01 \leq b1 \leq 0.5</math> с шагом <math>\Delta b1 = 0.09</math> вычислить по 4 значения <math>y = (b1 + b2 \cdot e^{-x})^{0.5}</math>, где начальное значение <math>b2 = 0.36</math>, а шаг <math>\Delta b2 = 0.12</math>, <math>x=1.44</math></p>
13	<p>На интервале <math>a1 \leq c1 \leq b1</math> с шагом <math>\Delta c1 = 0.3</math> для которого <math>c2</math> из интервала <math>a2 \leq c2 \leq b2</math> с шагом <math>\Delta c2 = 0.22</math> протабулировать функции</p> $y = \left( e^{-\frac{c1x}{a1+b2}} + e^{-\frac{c2x}{a2+b2}} \right)^{\frac{c1+c2}{x}}, \quad z = \cos(c1x + c2), \quad a1=0.74, \quad b1=1.56,$ <p><math>a2=0.22</math>, <math>b2=0.88</math>, <math>x=0.567</math></p>
14	<p>На интервале <math>0.25 \leq \delta \leq 0.75</math> протабулировать функцию</p> $z = \frac{\cos \delta - \cos x}{\sin^2 x + \sin^2 \delta},$ <p>причём, если <math>\operatorname{tg} \delta - x &gt; 0</math>, то <math>\Delta \delta = 0.05</math>, иначе <math>\Delta \delta = 0.1</math>, <math>x = 0.416</math></p>
15	<p>Для каждого <math>0.12 \leq y \leq 0.84</math> с шагом <math>\Delta y = 0.21</math> вычислить по 4 значения <math>s = \frac{\sqrt{y+z} - x^2}{\ln y + \ln z + x}</math>, где начальное значение <math>z=0.2</math>, а шаг <math>\Delta z = 0.05</math></p>
16	<p>На интервале <math>0.41 \leq h \leq 1.23</math> с шагом <math>\Delta h = 0.08</math> протабулировать функцию:</p> $\omega = \begin{cases} a, & h \leq 5 \\ b, & 0.6 \leq h \leq 0.7 \\ c, & \text{в остальных случаях} \end{cases}$ <p>где <math>a = x^2 - \cosh</math>; <math>b = \ln h - \ln x</math>; <math>c = x^3 + \operatorname{arctgh}</math>; <math>x=0.14</math></p>
17	<p>На интервале <math>x \leq y \leq 2x</math> с шагом <math>\Delta y = x/5</math> для каждого <math>z</math> из интервала <math>3x \leq z \leq 4x</math> с шагом <math>\Delta z = x/6</math> протабулировать функции</p> $s = \frac{y^2 - z^2}{y^2 + z^2}; \quad p = s \cdot e^{-s}$ <p><math>x=1.15</math></p>

Продолжение таблицы 21

18	<p>Для каждого <math>0.01 \leq a1 \leq 0.56</math> с шагом <math>\Delta a1 = 0.012</math> вычислить по 5 значений <math>a3 = x^{a2} \left[ 1 - e^{-\frac{a1+a2}{x}} \right]</math>, где начальное значение <math>a2 = 0.64</math>, шаг <math>\Delta a2 = 0.01</math>, <math>x=0.548</math></p>
19	<p>На интервале <math>0.1 \leq y1 \leq 0.8</math> с шагом <math>\Delta y1 = 0.1</math> для каждого <math>y2</math> из интервала <math>0.2 \leq y2 \leq 0.9</math> с шагом <math>\Delta y2 = 0.1</math> протабулировать функции <math>y3 = y5(y1 - y2)^2 + x^{y1}</math> и <math>y4 = y5 - e^{-\frac{x-y1}{y2}}</math>, где <math>y5 = x/y2</math>, <math>x=0.833</math></p>
20	<p>На интервале <math>a \leq y \leq b</math> протабулировать функцию <math display="block">d = \frac{(a-b)^2 + \ln y}{1 + \sin^2(x+y)} + \ln \left( \frac{a-b}{y} + \sqrt{x} \right),</math> причём, если <math>y &lt; \frac{b-a}{2.83}</math>, то <math>\Delta y = 0.2</math>, иначе <math>\Delta y = 0.25</math>, <math>a=1.18</math>, <math>b=2.44</math>, <math>x=0.564</math></p>

## 2.12 Массивы

В VBA различают два вида переменных – простые переменные и переменные структурного типа. Простые переменные служат для идентификации и резервирования памяти для одного данного. Переменные структурного вида предназначены для идентификации и резервирования памяти для нескольких данных. Частным случаем переменной структурного вида является массив.

Массивы – это формальное объединение нескольких однотипных элементов (чисел, символов, строк и т.д.), упорядоченных по индексам, определяющим положением каждого элемента в массиве.

Пример 1.

Результаты многократных замеров температуры воздуха в течение одного года удобно рассматривать как совокупность вещественных чисел, объединенных в один сложный объект, - массив измерений.

Значения температур в марте

Число	1	2	3	4	5	6	7	8	9
T, 0С	-10	-9	-11	-5	0	+3	-1	0	+4

Первая ячейка содержит значение температуры 1 марта, что можно обозначить  $T[1] = -10$ ;  $T[2] = -9$  и т.д. Заметим, что все ячейки таблицы содержат однотипные данные (T).

Пример 2.

Таблица, содержащая однотипные данные – классный журнал со списком учащихся, где каждая ячейка определяется двумя значениями – фамилией ученика и датой, а в ячейку заносятся отметки – целые числа от 1 до 5.

Определение.

Массив – это совокупность конечного числа элементов одного и того же типа, в которой каждый элемент имеет номер, а все элементы – общее имя.

Массивом называется ряд ячеек памяти, отведенных для хранения индексной переменной.

Массив – это структура данных, состоящая из фиксированного числа компонентов одного и того же типа, называемого базовым.

Номер элемента массива называется его индексом.

Если в форме описания массива задан один индекс, массив называется одномерным, если два индекса – двумерным, если  $n$  индексов –  $n$ -мерным.

Одномерный массив соответствует понятию линейной таблицы (вектора), двумерный – понятию прямоугольной таблицы (матрицы, набору векторов). Размерность ограничена только объемом памяти конкретного компьютера.

Характеристики массива:

- 1 Тип – общий тип элементов массива;
- 2 Размерность – количество индексов массива;
- 3 Диапазон – количество допустимых значений каждого индекса;
- 4 Размер – количество элементов массива.

Пример 3.

$x_i (5, 6, 7) \rightarrow$             размер – 3  
                                 размерность – 1  
                                 тип – integer  
                                 диапазон – 1..3

Все элементы массива имеют общее имя (имя массива) и различаются индексами. Элементами массива могут быть данные любого типа. Тип элементов массива называется базовым. Элементы, образующие массив, упорядочены таким образом, что каждому элементу соответствует совокупность номеров (индексов), определяющих его местоположение в общей последовательности. Доступ к каждому отдельному элементу осуществляется путем индексирования элементов массива. Индексы представляют собой выражения любого скалярного типа, кроме вещественного. Тип индекса определяет границы изменения значений индекса.

### 2.12.1 Объявление массива

Массивы должны быть объявлены до их использования в программе.

Синтаксис:

`Dim | Public | Private Имя_Массива (Размерность_Массива) As Тип_Данных`

где

`Dim | Public | Private` – ключевые слова, декларирующие массив и область его видимости;

`Имя_Массива` – идентификатор, определяющий имя массива;

`Размерность_Массива` – значение нижней и верхней границы значения индекса (номера) соответствующей размерности.

Допускается описание до 60 размерностей.

`As` – ключевое слово, предваряющее указание типа



элементов массива;

Тип\_Данных – любой, действительный для VBA тип данных (базовый или созданный пользователем).

Пример 1.

Dim A (7) As Integer

Объявляется одномерный массив (вектор) A из 8 целых чисел, причем по умолчанию первый элемент массива будет A(0), а последний A(7).

При объявлении многомерного массива в поле индекса указывается несколько индексов в соответствии с размерностью массива.

Пример 2.

Dim B (4, 5) As Single

Объявляется двумерный массив B 4 x 5 (матрица), состоящая из 5 строк и 6 столбцов действительных чисел.

По умолчанию значение нижнее границы массива при таком объявлении равно нулю. В этом случае говорят, что 0 – базовый индекс массива. Можно изменить базовый индекс, написав в области описания модуля (General Declarations) инструкцию Option Base 1. После этого индексы массивов A и B будут нумероваться с единицы.

Другим способом изменения базового индекса является использование ключевого слова To при объявлении массива.

Пример 3.

Dim A (1 To 7) As Integer

Dim B (1 To 4, 1 To 5) As Single

В этом примере базовому индексу массива установлено значение 1.

Подобным образом можно устанавливать как положительные, так и отрицательные базовые индексы

Пример 4.

Dim Z (-3 To 7) As Single.

### 2.12.2 Динамические массивы

Иногда в процессе выполнения программы требуется изменять размер массива. В этом случае первоначально массив объявляют как динамический. Для этого при объявлении массива не надо указывать его размерность.

Пример 1.

Dim R () As Single.

Затем в программе следует вычислить необходимое количество элементов массива и сохранить его в некоторой переменной, например n, а затем изменить размер динамического массива с помощью оператора ReDim.

Пример 2.

ReDim R (1 To 3).

Синтаксическая конструкция переопределения массива:

ReDim [Preserve] Имя\_Массива (Размерность\_Массива) As Тип\_Данных

где

ReDim – ключевое слово, указывающее, что переопределяются размеры массива;

Preserve – необязательное ключевое слово, используемое для сохранения данных в существующем массиве при изменении значения последней размерности. Если оно не используется, то данные в массиве при изменении размерности не сохраняются.

Имя\_Массива – имя массива, удовлетворяющее стандартным правилам именования переменных;

Размерность\_Массива - значение нижней и верхней границы значения индекса (номера) соответствующей размерности (до 60). Если отсутствует инструкция Option Base, нижняя граница индекса по умолчанию равна 0.

Тип\_Данных – тип данных массива.

Пример 3.

Dim A () As Integer ‘Объявление динамического массива

Dim n As Integer ‘Объявление переменной, для хранения размерности

N = 9 ‘Инициализация значения

ReDim A (n) ‘Одномерный массив из 10 элементов

ReDim A (3 To n, 1 To n) ‘Двумерный массив с базовыми индексами

Примечание. Оператор ReDim изменяет размерность массива, но не тип его элементов.

### 2.12.3 Действия над массивами

Для работы с массивом как единым целым используется идентификатор массива без указания индекса в круглых скобках. Массив может участвовать в операторе присваивания. Массивы, участвующие в этих действиях, должны быть идентичны по структуре, то есть иметь одинаковые типы индексов и одинаковые типы компонентов.

Пример 1.

Если массивы A и B описаны как

Dim A (1 To 2) As Byte, B (1 To 2) As Byte,

то применение к ним допустимой операции даст следующий результат:

#### Выражение

#### Результат

A = B      Все значения элементов массива B присваиваются соответствующим элементам массива A. Значения элементов массива B остаются неизменными

Пример 2.

Значение одного массива присваивается другому поэлементно в операторе цикла.

Sub primer ()

Dim A (1 To 2) As Byte, B (1 To 2) As Byte

Dim I As Byte

B(1) = 2

B(2) = 4

```
For I = 1 To 2
A(i) = B(i)
Debug.Print A(i), B(i)
Next i
End Sub
```

#### **2.12.4 Действия над элементами массива**

После объявления массива каждый его элемент можно обрабатывать, указав идентификатор (имя) массива и индекс элемента в круглых скобках.

Пример 1.

Запись

Mas (2), VectorZ (10)

позволяет обратиться к третьему элементу массива Mas и одиннадцатому элементу массива VectorZ.

Индексированные элементы массива называются индексированными переменными и могут быть использованы так же, как и простые переменные. Например, они могут находиться в выражениях в качестве операндов, использоваться в операторах параметрических и итерационных циклов, входить в качестве параметров в операторы ввода и вывода; им можно присваивать любые значения, соответствующие их типу.

#### **2.12.5 Формирование массива**

1 способ.

Формирование массива в программе определяется поэлементно с помощью последовательности операторов.

Пример 1.

Dim B (1 To 2, 1 To 3) As Single

B (1,1) = 2 : B (1,2) = 4 : B (1,3) = 6

B (2,1) = 1 : B (2,2) = 6 : B (2,3) = 2

2 способ.

Формирование массива с помощью датчика случайных чисел.

Пример 2.

Sub primer ()

Dim A(1 To 12) As Variant

Randomize

For i = 1 To 10

A(i) = Rnd (40)

Debug.Print A(i)

Next i

End Sub

3 способ

Инициализация элементов массива оператором цикла.

Наиболее эффективно эта операция выполняется с помощью оператора For.

Пример 3.

```
Dim M (1 To 9, 1 To 9) As Integer
Dim i As Integer
Dim j As Integer
For i = 1 To 9
For j = 1 To 9
M(i, j) = i * j
Next j
Next i
```

4 способ

Использование функции Array (массив) для задания одномерного массива

Удобным способом определения одномерных массивов является функция Array, преобразующая список элементов, разделенных запятыми, в вектор из этих значений, и присваивающая их переменной типа Variant.

Пример 4.

```
Sub ShowArray ()
Dim A As Variant
A = Array (10, 20, 30)
Debug.Print A(0) 'Отобразится: 10
Debug.Print A(2) 'Отобразится: 30
End Sub
```

Использование функции Array (массив) для задания многомерного массива

Функция Array позволяет задавать не только одномерные, но и многомерные массивы. Это обеспечивается использованием вложенной конструкции с функциями Array.

Пример 5.

```
Option Base 1
Sub MultiDim ()
Dim A As Variant
A = Array (Array("Санкт-Петербург", "Россия"), Array ("Киев", "Украина"))
Debug.Print A (1) (1) 'Отобразится: Санкт-Петербург
Debug.Print A (1) (2) 'Отобразится: Россия
End Sub
```

Функция Array позволяет задавать не просто многомерные массивы, а массивы, у которых строки имеют различное число компонентов.

Пример 6.

```
Private Sub VarArray ()
Dim M As Variant
M = Array(Array(1,2), Array (3,4,5))
Debug.Print M (0) (0), M (1) (2) 'Отобразится: 1 5
End Sub
```

## 2.12.6 Одномерные массивы

### Пример 1.

Формирование массива в программе с помощью последовательности операторов.

Постановка задачи. Реализовать на языке VBA алгоритм поиска наибольшего элемента последовательности с определением его местонахождения.

Технология выполнения задания:

1 Набрать в стандартном модуле проекта следующую пользовательскую процедуру:

```
Sub primer()  
Const n = 10  
Dim N_max As Integer  
Dim i As Integer  
Dim Max As Single  
Dim B(1 To n) As Single  
B(1) = 0.2: B(2) = 1.4: B(3) = 0.6  
B(4) = 0.121: B(5) = 0.77: B(6) = 9.45  
B(7) = 8.21: B(8) = 0.4: B(9) = 0.3  
B(10) = 4.11  
Max = B(1)  
N_max = 1  
For i = 2 To n  
If B(i) > Max Then  
Max = B(i)  
N_max = i  
End If  
Next i  
Debug.Print "Max= " & Max & " N_max= " & N_max  
End Sub
```

2 Вычислить.

### Пример 2.

Использование функции Array (массив) для задания одномерного массива.

Постановка задачи. Прежняя.

Технология выполнения задания:

1 Набрать в стандартном модуле проекта следующую пользовательскую процедуру:

```
Option Base 1  
Sub primer()  
Dim i As Integer  
Dim Max As Single  
Dim N_Max As Integer  
Dim B As Variant  
B = Array(0.1, 1.12, 3.12, 0.45, 101.47, 12.4, 10.12, 10.11, 6.47, 13.4)  
Max = B(1)
```

```

N_Max = 1
For i = 2 To 10
If B(i) > Max Then
Max = B(i)
N_Max = i
End If
Next i
Debug.Print "Max= " & Max & " N_max= " & N_Max
End Sub
2 Вычислить.

```

### Лабораторная работа № 6 Одномерные массивы

Цель: приобретение навыков программирования обрабатывать последовательности с помощью пользовательской подпрограммы-процедуры. Индивидуальные варианты лабораторной работы № 6 представлены в таблице 22 (2 часа).

Таблица 22

№ В	Варианты индивидуальных заданий
	Разработать процедуру пользователя для обработки последовательности
1	Даны натуральные числа $n, a_1, a_2, \dots, a_n$ . Определить количество $a_k$ последовательности $a_1, a_2, \dots, a_n$ являющихся нечётными числами
2	Даны натуральные числа $n, a_1, a_2, \dots, a_n$ . Определить количество $a_k$ последовательности $a_1, a_2, \dots, a_n$ кратных 3 и не кратных 5.
3	Даны натуральные числа $n, a_1, a_2, \dots, a_n$ . Определить количество $a_k$ последовательности $a_1, a_2, \dots, a_n$ являющихся квадратами чётных чисел.
4	Даны натуральные числа $n, a_1, a_2, \dots, a_n$ . Определить количество $a_k$ последовательности $a_1, a_2, \dots, a_n$ удовлетворяющих условию $a_k < \frac{a_{k-1} + a_{k+1}}{2}.$
5	Даны натуральные числа $n, a_1, a_2, \dots, a_n$ . Определить количество $a_k$ последовательности $a_1, a_2, \dots, a_n$ удовлетворяющих условию $2^k < a_k < k!.$
6	Даны натуральные числа $n, q_1, q_2, \dots, q_n$ . найти те члены $q_i$ последовательности $q_1, q_2, \dots, q_n$ которые являются удвоенными нечётными числами.
7	Даны натуральные числа $n, q_1, q_2, \dots, q_n$ . найти те члены $q_i$ последовательности $q_1, q_2, \dots, q_n$ которые при делении на 7 дают остаток 1.2 или 5.
8	Даны натуральные числа $n, q_1, q_2, \dots, q_n$ . найти те члены $q_i$ последовательности $q_1, q_2, \dots, q_n$ которые обладают тем свойством, что корни уравнения $x^2 + 3q_i - 5$ действительны и положительны.

Продолжение таблицы 22

9	Вычислить $\sum_{i=1}^{30} (a_i)^2$ , где $a_i = \begin{cases} i, & \text{если } i - \text{нечетное} \\ i/2, & \text{в противном случае} \end{cases}$
10	Дано натуральное число $n$ . Получить сумму тех чисел вида $i^3 - 3in^2 + n, i = 1, 2, \dots, n$ , которые являются удвоенными нечётными. При отсутствии таких чисел искомая сумма равна нулю.
11	Даны целые числа $a_1, \dots, a_{50}$ . Получить сумму тех чисел данной последовательности, которые кратны 5.
12	Даны целые числа $a_1, \dots, a_{50}$ . Получить сумму тех чисел данной последовательности, которые нечётны и отрицательны.
13	Даны целые числа $a_1, \dots, a_{50}$ . Получить сумму тех чисел данной последовательности, которые удовлетворяют условию $ a_i  < i^2$ .
14	Даны натуральное число $n$ , действительные числа $a_1, a_2, \dots, a_n$ . В последовательности $a_1, a_2, \dots, a_n$ все отрицательные члены увеличить на 0.5, а все неотрицательные заменить на 0.1.
15	Вычислить $\sum_{i=1}^{30} (b_i)^2$ , где $b_i = \begin{cases} i^2, & \text{если } i - \text{нечетное} \\ i^3, & \text{в противном случае} \end{cases}$
16	Даны целые числа $a, n, x_1, \dots, x_n$ ( $n > 0$ ). Определить, каким по счёту идёт в последовательности $x_1, \dots, x_n$ член, равный $a$ . Если такого члена нет, то ответом должно быть число 0.
17	Даны натуральные числа $n, a_1, a_2, \dots, a_n$ . В последовательности $a_1, a_2, \dots, a_n$ получить сумму положительных и число отрицательных членов.
18	Даны натуральные числа $n, a_1, a_2, \dots, a_n$ . Заменить все большие семи члены последовательности $a_1, a_2, \dots, a_n$ числом 7. Вычислить количество таких членов.
19	Даны целые числа $a_1, a_2, \dots, a_{45}$ . Получить число отрицательных членов последовательности $a_1, a_2, \dots, a_{35}$ и число нулевых членов всей последовательности $a_1, a_2, \dots, a_{45}$ .
20	Запишите +1 вместо максимального элемента массива $(a_1, a_2, \dots, a_{50})$ , а -1 вместо минимального.

### 2.12.7 Двумерные массивы. Вложенные циклы.

Пример 1.

Постановка задачи. Даны натуральное число  $n$ , действительная матрица размера  $n \times 3$ . Получить сумму положительных и число отрицательных членов во всей матрице.

Технология выполнения задания:

1 Набрать в стандартном модуле проекта следующую пользовательскую процедуру:

```
Sub primer()  
Const n = 3  
Const m = 3  
Dim S As Single  
Dim i As Integer  
Dim j As Integer  
Dim k As Integer  
Dim A(1 To n, 1 To m) As Single  
A(1, 1) = -11: A(1, 2) = 0: A(1, 3) = 2  
A(2, 1) = 5: A(2, 2) = 45: A(2, 3) = 32  
A(3, 1) = -3: A(3, 2) = 14: A(3, 3) = -25  
S = 0  
k = 0  
For i = 1 To n  
For j = 1 To m  
If A(i, j) >= 0 Then S = S + A(i, j) Else k = k + 1  
Next j  
Next i  
Debug.Print "S= " & S & " k= " & k  
End Sub
```

2 Вычислить.

Пример 2.

Постановка задачи. Дана действительная матрица  $m \times n$ . Определить числа  $b_1, \dots, b_m$ , равные наименьшим значениям элементов строк.

Технология выполнения задания:

1 Набрать в стандартном модуле проекта следующую пользовательскую процедуру:

```
Option Base 1  
Sub primer()  
Dim i As Integer  
Dim j As Integer  
Dim M(1 To 3, 1 To 3) As Variant  
Randomize  
For i = 1 To 3  
For j = 1 To 3  
M(i, j) = Rnd(20)  
Debug.Print ("M(" & i & j & ")=" & M(i, j))  
Next j  
Next i
```



```

For i = 1 To 3
Min = M(i, 1)
For j = 2 To 3
If M(i, j) < Min Then Min = M(i, j)
Next j
Debug.Print ("Min(" & i & ")= " & Min)
Next i
End Sub

```

2 Вычислить.

Пример 3.

Постановка задачи. Даны натуральное число  $n$ , действительная матрица размера  $n \times 3$ . Найти среднее арифметическое, каждого из столбцов.

Технология выполнения задания:

1 Набрать в стандартном модуле проекта следующую пользовательскую процедуру:

```

Option Base 1
Sub primer()
Const m = 3: Const n = 4
Dim i As Integer
Dim j As Integer
Dim s As Single
Dim A(1 To m, 1 To n) As Integer
For i = 1 To m
For j = 1 To n
A(i, j) = i * j
Debug.Print "A(" & i & j & ")=" & A(i, j)
Next j
Next i
For j = 1 To n
s = 0
For i = 1 To m
s = s + A(i, j) / m
Next i
Debug.Print "s= " & s
Next j
End Sub

```

2 Вычислить.

### Лабораторная работа № 7

#### Двумерные массивы. Вложенные циклы

Цель: приобретение навыков программирования обработки табличных данных с помощью пользовательской подпрограммы-процедуры. Индивидуальные варианты лабораторной работы № 7 представлены в таблицах 23, 24 (4 часа)

Таблица 23

<b>№ В</b>	<b>Варианты индивидуальных заданий</b>
Построить процедуру пользователя для обработки таблиц	
1	Дана действительная матрица размера $m \times n$ . Определить числа $b_1, \dots, b_m$ равные суммам элементов строк.
2	Дана действительная матрица размера $m \times n$ . Определить числа $b_1, \dots, b_m$ равные произведениям элементов столбцов.
3	В массиве данных $c_1, \dots, c_{20}$ определить сумму квадратов отрицательных элементов и сумму положительных. Подсчитать количество положительных, отрицательных и нулевых элементов.
4	Дана действительная матрица размера $m \times n$ . Определить числа $b_1, \dots, b_m$ равные наименьшим значениям элементов строк.
5	Дана действительная матрица размера $m \times n$ . Определить числа $b_1, \dots, b_m$ равные значениям средних арифметических элементов строк.
6	Дана действительная матрица размера $m \times n$ . Определить числа $b_1, \dots, b_m$ равные разностям наибольших и наименьших значений элементов строк.
7	Даны натуральное число $n$ , действительная матрица размера $n \times 9$ . Найти среднее арифметическое: а) каждого из столбцов; б) каждого из столбцов, имеющих чётные номера.
8	Даны целые числа $a_1, a_2, a_3$ . Получить целочисленную матрицу $[b_{ij}]$ , $i, j = 1, 2, 3$ , для которой $b_{ij} = a_i - 3a_j$ .
9	Даны действительные числа $a, \dots, a$ ; $b, \dots, b$ . Получить действительную матрицу $[c_{ij}]$ , $i = 1, \dots, 20$ , $j = 1, \dots, 10$ , для которой $c_{ij} = a_j / (1 +  b_i )$ .
10	Дано натуральное число $n$ . получить действительную матрицу $[a_{ij}]$ , $i, j = 1, \dots, n$ , для которой $a_{ij} = \begin{cases} \sin(i + j), & i < j \\ 1, & i = j \\ \arcsin \frac{i + j}{2i + 3j}, & \text{в остальных случаях} \end{cases}$
11	Получить $[a_{ij}]$ , $i = 1, \dots, 10$ , $j = 1, \dots, 12$ - целочисленную матрицу для которой $a_{ij} = i + 2j$ .

Продолжение таблицы 23

12	Получить действительную матрицу $[a_{ij}] ; i, j = 1, \dots, 7$ , первая строка которой задаётся формулой $a_{1j} = 2j + 3 (j = 1, \dots, 7)$ , вторая строка задаётся формулой $a_{2j} = j - \frac{3}{2 + \frac{1}{j}}$ , $(j = 1, \dots, 7)$ , а каждая следующая строка есть сумма двух предыдущих.
13	Дано натуральное число n. Выяснить сколько положительных элементов содержит матрица $[a_{ij}] ; i, j = 1, \dots, n$ , если $a_{ij} = \sin(i + \frac{j}{2})$ .
14	Дана действительная матрица $n \times m$ , в которой не все элементы равны нулю. Получить новую матрицу путём деления всех элементов данной матрицы на её наибольший по модулю элемент.
15	Дана действительная квадратная матрица порядка 12. Заменить нулями все её элементы, расположенные на главной диагонали и выше неё.
16	Даны действительные числа $x_1, \dots, x_8$ . Получить действительную квадратную матрицу порядка 8: $\begin{bmatrix} x_1 & x_2 & \dots & x_8 \\ x_1^2 & x_2^2 & \dots & x_8^2 \\ \dots & \dots & \dots & \dots \\ x_1^8 & x_2^8 & \dots & x_8^8 \end{bmatrix}$
17	Даны натуральное число n, действительная матрица $[a_{ij}] ; i, j = 1, \dots, n$ . Получить последовательность элементов главной диагонали $a_{11}, a_{22}, \dots, a_{nn}$ .
18	Дана действительная матрица размера $6 \times 9$ . Найти среднее арифметическое наибольшего и наименьшего значений её элементов.
19	Сформировать массив p из 20 элементов, i-ый элемент которого определяется по формуле $p_i = xe^i (\sin \frac{x}{i})$ , где $x=0.225$ . Расставить в порядке возрастания элементы массива.
20	В массиве данных $c_1, \dots, c_{20}$ определить сумму квадратов отрицательных элементов и сумму положительных. Подсчитать количество положительных, отрицательных и нулевых элементов.

Таблица 24

№ В	Варианты индивидуальных заданий						
Разработать экономическую процедуру пользователя							
1	прод./цена р.	Москва	С-Петер.	Новгород	Вологда	Калуга	Сыктывкар
	молоко	1800	2500	3000	1700	2600	3800
	сметана	5000	6000	5200	4800	5500	7000
	творог	6000	6200	7000	6800	7100	6300
	масло слив.	16000	16700	16900	15000	17000	16000
	сыр	20000	22000	24000	21000	19000	19500
	По данной таблице определить максимальную свободную цену реализации каждого вида продукции молочного комбината на различных рынках сбыта за минувший месяц и вычислить максимальную прибыль от реализации единицы продукции при условии, что в среднем убыток от реализации единицы (1 кг) всех видов продукции (расходы на транспорт, потери и т.п.) составил 500 р. Прибыль от реализации единицы продукции						
	$Pr=Cp-C/ед-Уб$						
	где						
	Cp - цена реализации 1 кг; C/ед - себестоимость 1 кг продукции (вводим); Уб - убыток на ед.продукции.						
2	Вывести на экран крупнейшую компанию мира ,чистая прибыль которой превышает 1 тыс.дол.						
3	На трех торговых точках продается 4 вида товаров. Известна цена каждого товара и прибыль, полученная от реализации единицы товара, а также количество каждого товара. Необходимо найти стоимость товара и прибыль с каждой торговой точки по формуле $c[i,j]=c[i,j]+a[i,l]*b[l,j]$ .						
4	Вычислить, каким был максимальный курс акций фирмы после четырех торгов на бирже. Курс акций = Дивиденд/Ссудный процент*100 (массив - курс акций).						
5	Динамика ВВП ведущих развитых стран составила (млрд.дол.): в 1993 году: США - 4500; РОССИЯ - 4300 в 1994 году: США - 4460; РОССИЯ - 4390 Вычислить максимальный ВВП.						
6	Дан курс покупки доллара (\$) за 7 дней (неделю). Рассчитать, какова средняя цена покупки \$ за неделю.						
	1день	2день	3день	4день	5день	6день	7день
	4500	4520	4600	4750	5000	5120	5100

7	Даны данные о средней зарплате с 1990 по 1993 год (111.4; 100.0; 67.7; 64.8). Найти среднее арифметическое наименьшей и наибольшей зарплат.
8	Задан массив, содержащий разряды рабочих завода. Посчитать количество трудящихся, работающих по первому разряду.

Продолжение таблицы 24

9	Дано 6 фирм. Определить, продукция которых из них пользуется наибольшим спросом, если продажа (в единицах товара) составляет: в 1-й фирме - 35, во 2-й - 46, в 3-й - 10, в 4-й - 68, в 5-й - 30, в 6-й - 15																										
10	Вычислить сальдо (SALD) торгового баланса по формуле: $SALD = KR - DB$ Кредит (KR): нефть сырая 8370; нефтепродукты 3471; природный газ 7443; золото 1284 Дебет (DB): зерно 1576; прочее продовольствие 3912; медикаменты 269; машины и оборудование 9409; гуманитарная помощь 1000																										
11	Известны кредиты банка для разных предприятий. Посчитать сумму кредитов банка.																										
12	В каком году в период с 1945 по 1985 в США наблюдался максимальный уровень инвестирования. Данные об инвестировании поместить в одномерные массивы.																										
13	Определить, в каком месяце 1984 года в Италии темп инфляции был минимальным. <table><tr><td>месяцы</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td></tr><tr><td>инфляция</td><td>15.9</td><td>16.0</td><td>16.0</td><td>17.5</td><td>16.3</td><td>16.0</td><td>15.8</td><td>17.2</td><td>16.8</td><td>16.4</td><td>16.0</td><td>15.9</td></tr></table>	месяцы	1	2	3	4	5	6	7	8	9	10	11	12	инфляция	15.9	16.0	16.0	17.5	16.3	16.0	15.8	17.2	16.8	16.4	16.0	15.9
месяцы	1	2	3	4	5	6	7	8	9	10	11	12															
инфляция	15.9	16.0	16.0	17.5	16.3	16.0	15.8	17.2	16.8	16.4	16.0	15.9															
14	Подсчитать сумму прибыли в 3 магазинах за неделю. Для этого введите сумму прибыли, полученной в каждый день недели для каждого из 3 магазинов.																										
15	Предприниматель в течение года каждый месяц вкладывал в банк деньги под проценты. Определить, в какой месяц сумма вклада была минимальной. Вывести на печать минимальную сумму и месяц. Считать, что 1-январь, 2-февраль, 12-декабрь. Вводить данные (в тысячах): январь - 840; май - 830; сентябрь - 560; февраль - 1020; июнь - 720; октябрь - 2000; март - 950; июль - 8015; ноябрь - 3005; апрель - 640; август - 6040; декабрь - 980																										
16	В результате инвентаризации обнаружено, что у многих запчастей истек срок их эксплуатации и они подлежат списанию со счета 10 "Материалы". Определить количество списываемых запчастей. Исходные данные: S[i] - износ каждой детали SROK - предельный срок эксплуатации n-количество запчастей.																										

17	Даны дивиденды на акцию за пять лет. Найти максимальные дивиденды.
18	Даны данные о размерах дохода и расхода предприятия за каждый месяц. Какие месяцы были убыточными.

#### Продолжение таблицы 24

19	Вычислить налог на заработную плату нескольких сотрудников, учитывая, что до 48000 он составляет 7,51% , а больше 48000 - 3,755%.										
20	Дан схематичный расчетный баланс за определенный период (1995-2004). Составить программу, позволяющую найти первый отрицательный баланс (платежи превышают поступления) и вывести это число на экран дисплея. Исходными данными является баланс.										
	Расчетный баланс										
		1995	1996	1997	1998	1999	2000	2001	2002	2003	2004
	платежи	100	100	100	100	100	500	500	500	500	500
	поступления	100	225	225	225	225	225	500	500	500	500
	баланс	0	125	125	125	125	275	275	275	275	500

## 3 Объектно-ориентированное программирование на VBA

### 3.1 Разработка программы создания приложения с помощью Форм

#### Пример 1.

Постановка задачи. Требуется разработать программу вычисления значения выражения при заданных числах, вводимых пользователем с клавиатуры в два (три) поля с именами «Число1», «Число2» («Число3»), с отображением результата в поле «Результат». Результат должен отображаться после нажатия кнопки <Вычислить>. Программа должна заканчивать работу после нажатия кнопки <END>. Окно программы должно иметь вид, показанный на рисунке 22.

$$z = \ln \left| \lg x - \sqrt{|\cos x - e^x|} \cdot \arcsin \left| \operatorname{tg} \frac{|ax - b|}{\sin |x|} + b \right| \right|, \text{ при } a=0.126, b=0.842, x=0.34$$

Технология выполнения задания:

Процесс программирования приложения Вычисление предусмотрен тремя этапами, представленными в таблице 25.

Таблица 25 - Этапы программирования

Этапы программирования	Число объектов
1 Создание пользовательского интерфейса	11 объектов
2 Установка свойств объектов	11 объектов
3 Составление программного кода	2 объекта

#### 1 этап. Разработка пользовательского интерфейса

1 Запустить среду программирования VBA.

Рисунок 22 – Окно программы «Вычисление»

2 Выполнить команду Insert→UserForm. В центре экрана появляется чистая форма, как показано на рисунке 23, которая будет использоваться для создания пользовательского интерфейса.

3 Поместить указатель мыши в правый нижний угол окна формы так, чтобы он принял вид двунаправленной стрелки. Переместить указатель, увеличивая размеры формы так, чтобы на ней могли разместиться все объекты программы.

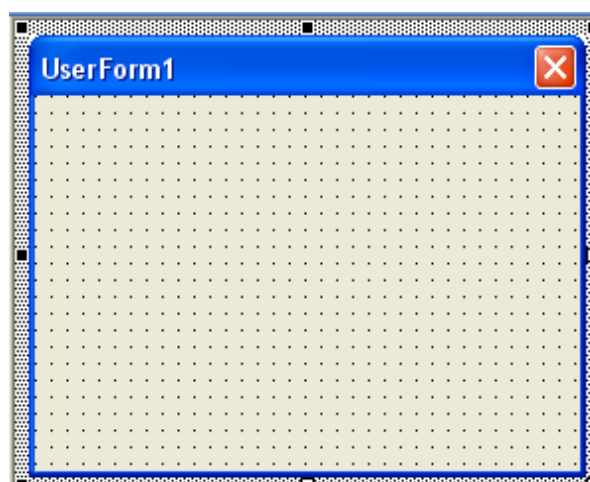


Рисунок 23 – Пользовательская форма

4 Создать на форме объект Командная кнопка.

- щелкнуть по кнопке <Command Button> (Командная кнопка) на панели инструментов (ToolBox);
- переместить указатель мыши на форму в левый нижний угол;
- удерживая левую кнопку мыши нажатой, вычертить указателем мыши прямоугольный объект. Он приобретает определенные очертания и «привязывается» к сетке формы.
- отпустить кнопку мыши. Экран будет выглядеть, как показано на рисунке 24.

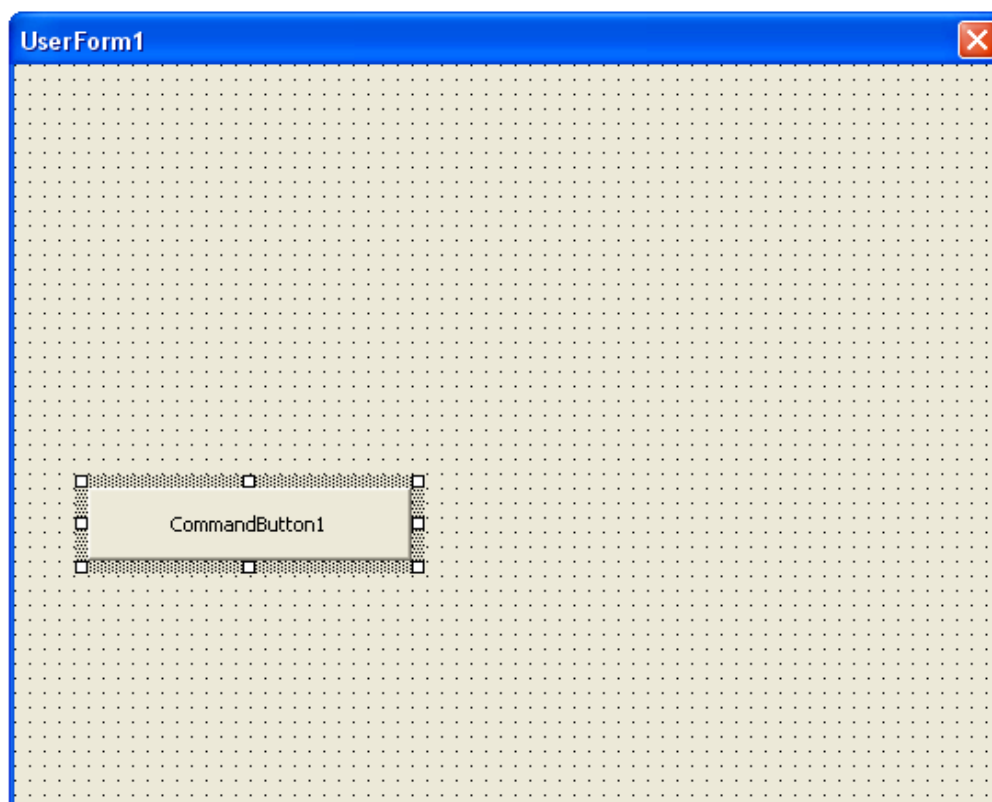


Рисунок 24 – Вид формы с объектом «Командная кнопка»

По умолчанию первая командная кнопка, имеющая набор команд управления, называется <Command1>. Ее можно перемещать и изменять ее размеры. При желании можно изменить шаг сетки (меню Tools (Настройки), команда Options (параметры), вкладка General (Общие)). Например, Grid Units: Points Width: 6, Height: 6.

5 Аналогично создать вторую командную кнопку, поместив ее ниже первой. Кнопка получит имя <Command2>.

6 Числовые и текстовые данные могут вводиться в программах VBA в текстовые поля (окна). Для того чтобы создать на форме текстовое поле, необходимо:

- щелкнуть по кнопке <TextBox> (Текстовое окно) панели инструментов;



- переместить указатель мыши в левый верхний угол формы и, удерживая нажатой левую кнопку, очертить указателем мыши прямоугольный объект. По умолчанию объект получает имя Text1.

7 Аналогично создать объекты Text2, Text3, поместив их над объектом Text1.

8 Для показа текста, чисел и символов при работе программы и формирования надписей используется специальный элемент пользовательского интерфейса, называемый этикеткой.

Добавим в форму этикетку для отображения результата. Для этого нужно:

- щелкнуть мышью по кнопке <Label> (Этикетка) на панели инструментов;
- поместить указатель мыши в верхний правый угол формы и, удерживая нажатой левую клавишу мыши, очертить прямоугольное окно.

Созданный объект получает имя Label1. После этого экран будет выглядеть, как показано на рисунке 25.

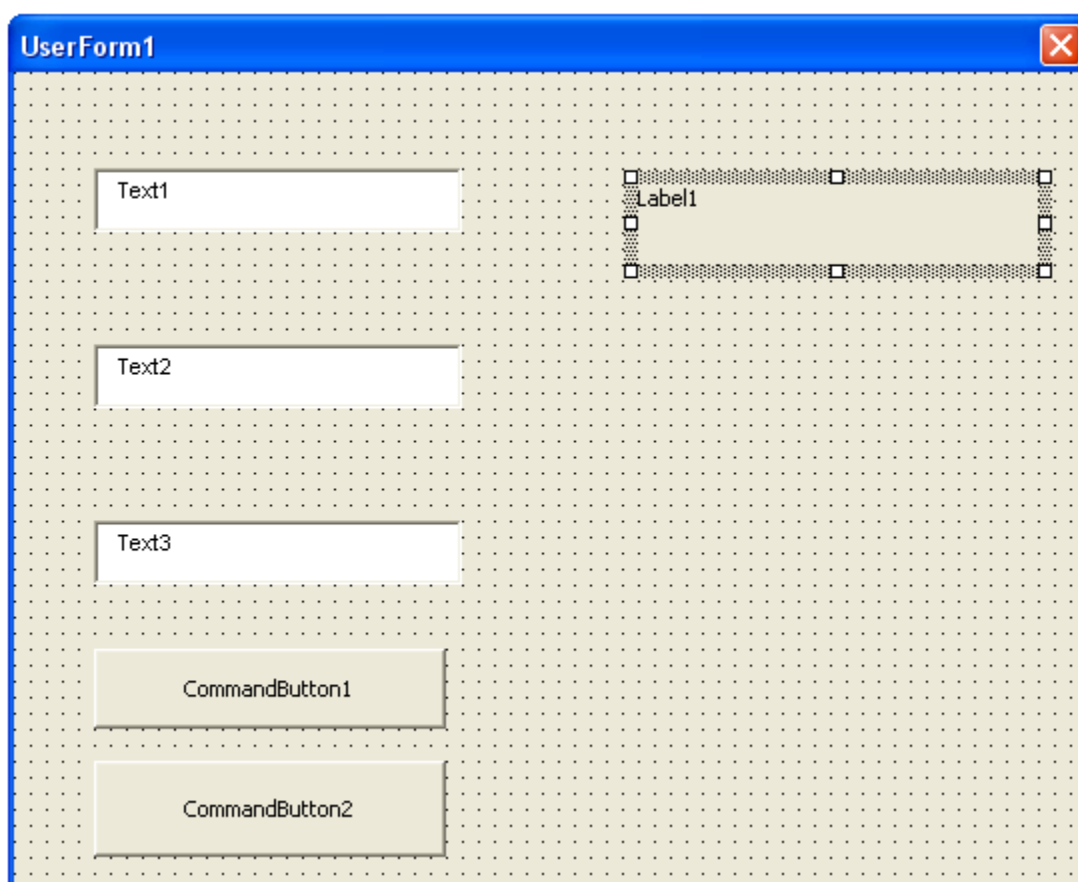


Рисунок 25 – Вид формы с объектами «Командные кнопки», «Текстовые поля», «Этикетки»

9 Добавить этикетки для надписей над окнами ввода чисел, результата и названия программы. Для этого нужно повторить пять раз операцию построения этикеток:

- щелкнуть по кнопке <Label> панели инструментов;
- переместить указатель мыши на форму и, удерживая левую клавишу мыши, построить прямоугольные окна для надписей).

Полученный на экране результат приведен на рисунке 26.

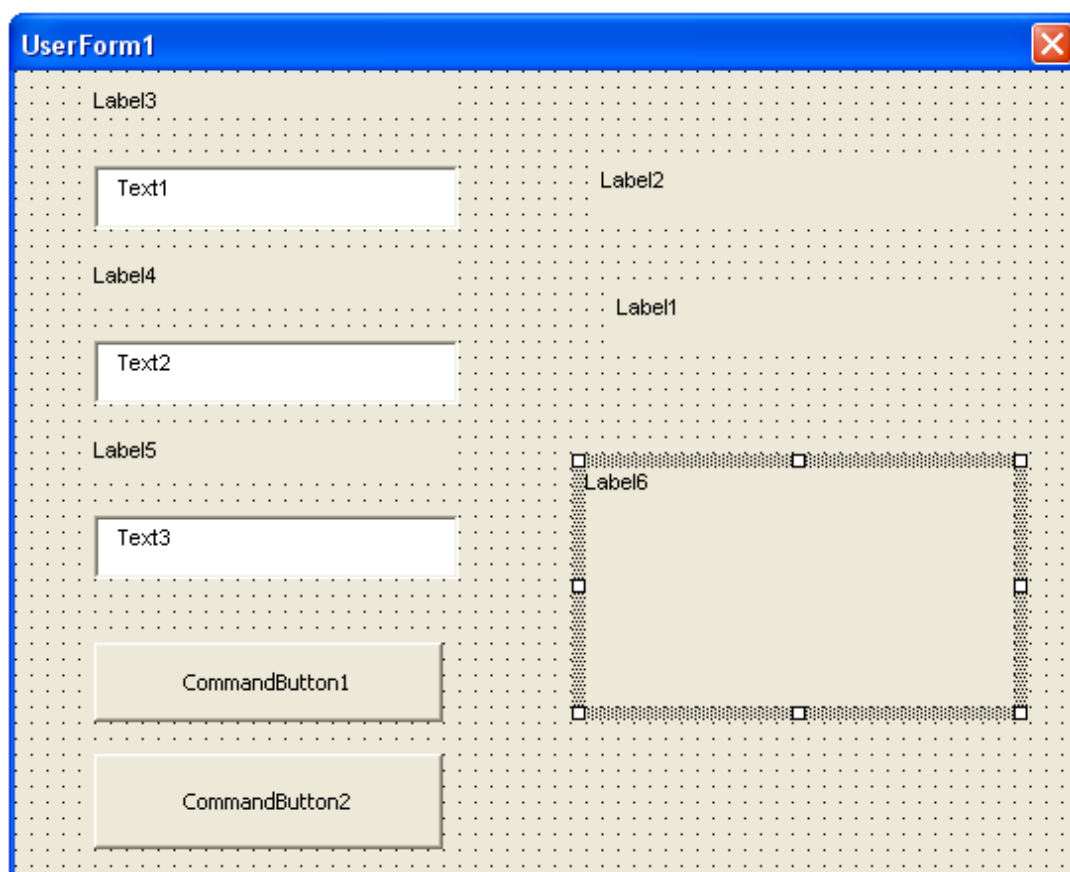


Рисунок 26 – Окончательный вид формы

## II этап. Установка свойств объектов

Изменим надпись формы с UserForm1 на значимую в соответствии с решаемой задачей. Для этого необходимо выполнить:

- щелкнуть мышью на форме;
- в окне «Properties» (Свойства) появятся свойства формы;
- дважды щелкнуть на свойстве Caption (Название) в левой колонке окна свойств для выделения установленного по умолчанию названия «UserForm1»;
- ввести название «Вычисление».

Алгоритм действий:

1) щелкнуть по первой командной кнопке (<CommandButton1>). Кнопка будет выделена маркерами;

2) в окне свойств, показанном на рисунке 27, будут перечислены установки для командной кнопки (цвет фона, название, размер шрифта, ширина кнопки и др.);

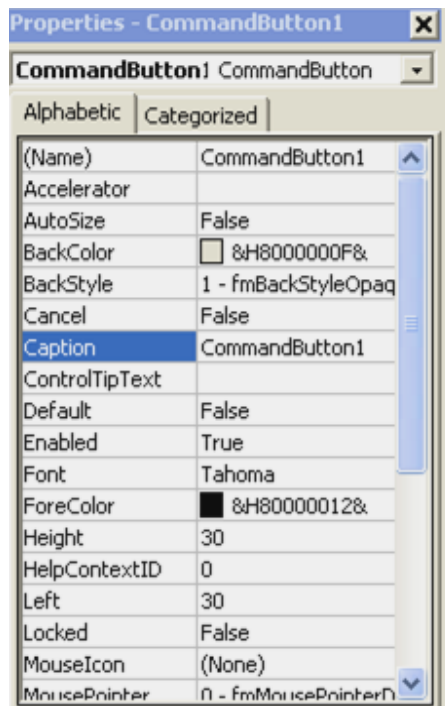


Рисунок 27 – Вид окна «Свойств»

3) дважды щелкнуть на свойстве Caption в левой колонке окна свойств для выделения установленного по умолчанию названия (CommandButton1);

4) набрать новое название для кнопки <Вычислить> и нажать клавишу Enter (можно щелкнуть мышью по форме). Название командной кнопки в окне свойств и в форме изменится на <Вычислить>.

5) аналогично надо изменить название второй командной кнопки. Это можно сделать другим способом. Для этого нужно:

- открыть ниспадающее окно «Object List» (Список объектов) щелчком мыши по кнопке в верхней части окна свойств. В списке будут перечислены все объекты программы, как показано на рисунке 28.



Рисунок 28 – Вид окна «Список объектов»

- щелкнуть на строке <CommandButton2> в окне списка. В окне свойств появятся установки для второй командной кнопки;

- дважды щелкнуть на свойстве названия (<CommandButton2>), набрать <End> и нажать клавишу Enter. Название второй кнопки изменится на <End>.

б) свойства текстовых полей 1, 2, 3 должны быть одинаковыми (они выполняют одинаковые функции – ввод чисел). Поэтому установим их как группу.

Для этого необходимо выполнить:

- щелкнуть мышью по первому объекту (Text1);
- удерживая нажатой клавишу Ctrl, щелкнуть мышью по второму объекту (Text2) и третьему (Text3).

В результате текстовые поля будут выделены, а устанавливаемые значения свойств в окне «Properties» будут общими (одними и теми же) для полей «Text1», «Text2», «Text3».

7) установить свойство Text для текстовых полей в значение Empty (Пусто).

Это означает, что нужно удалить текущую установку и оставить свойство пустым. В этом случае в поля «Text1»- «Text3» можно вводить произвольную информацию с клавиатуры.

8) перейти к установке свойств для этикетки Label1, предназначенной для отображения результата. Для этого необходимо выполнить:

- щелкнуть мышью по объекту Label1 для его выделения;
- щелкнуть на свойстве TextAlign (Выравнивание текста), а затем справа на стрелке окна ниспадающего списка;
- щелкнуть на цифре 2 - <Center> (Выравнивание по центру). Отображаемый результат будет выравниваться по центру поля.

9) изменить стиль рамки.

Для этого:

- щелкнуть мышью на свойстве Border Style (Стиль рамки), а затем – на стрелке окна ниспадающего списка, который появится справа;
- щелкнуть на значение 1 - <fmBorderStyleSingle>, чтобы добавить тонкую рамку вокруг этикетки;

10) установить шрифт, которым будут выводиться цифры в окне результата:

- дважды щелкнуть по свойству Font (Шрифт);
- появится диалоговое окно, приведенное на рисунке 29.
- в нем выбрать желаемый шрифт (например, Times New Roman), начертание (полужирный), кегль (14);
- щелкнуть по кнопке <OK>.

11) удалить название объекта Label1, так как в поле объекта будет выводиться числовой результат:

- дважды щелкнуть мышью на свойстве Caption и нажать клавишу Delete.

12) аналогично установить общие для этикеток Label2-Label5 свойства: выравнивание, стиль рамки и шрифт. Для этого необходимо:

- выделить эти объекты как группу, используя клавишу Ctrl;
- поочередно щелкая мышью по объектам;
- установить общие свойства TextAlign, Border Style, Font, как это делалось для объекта Label1.

13) установить свойство Caption для объектов Label2-Label5, используя значения Результат, Число 1, Число 2, Число 3 соответственно.

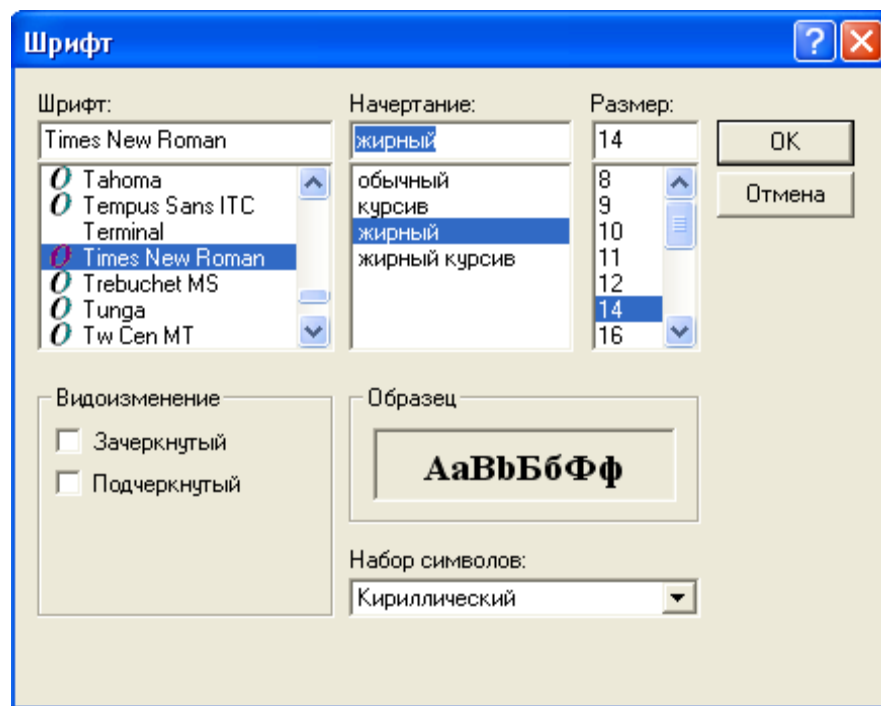


Рисунок 29 – Вид диалогового окна Шрифт

14) щелкнуть по объекту Label6 для его выделения. Эта этикетка должна иметь название программы Вычисление, поэтому значение свойства Caption заменить на Вычисление.

15) установить свойства объекта Label6:

- выравнивание по центру;
- отсутствие рамки (значение Border Style равно нулю);
- шрифт Arial;
- полужирный;
- кегль – 20.

16) изменить цвет текста.

Для этого необходимо выполнить:

- дважды щелкнуть мышью на свойстве ForeColor в окне свойств;
- в поле списка появятся вкладки System (Система) и Palette (Палитра), позволяющие изменять цвет объекта.

Вкладка System показывает цвета пользовательского интерфейса Windows.

Вкладка Palette содержит все доступные цвета.

- щелкнуть на вкладке Palette, а затем – на квадратике красного (или другого) цвета.

Название Вычисление приобрело этот цвет, в окне свойств он представляется шестнадцатеричным числом.

### III этап. Составление программного кода

Большинство объектов, которые созданы, уже «знают», как работать после запуска программы, и готовы к приему данных от пользователя. Внутренняя функциональность создаваемых объектов – одна из мощнейших особенностей VBA.

В разрабатываемой программе необходим код, который будет вычислять значение выражения по определенной формуле с помощью вводимых пользователем чисел и отображать результат вычислений.

Вычислительная логика может быть встроена в приложение только с помощью программного кода, который точно определяет, что именно должна делать программа на каждом этапе работы. Программа управляется кнопками <Вычислить> и <END> (Конец), поэтому программный код будет связан с информацией, поступающей от этих кнопок.

Для работы с программным кодом командной кнопки нужно дважды по ней щелкнуть мышью. Например, щелкнем по кнопке <END> на форме. После этого появляется окно «Cod» (Код), изображенное на рисунке 30.

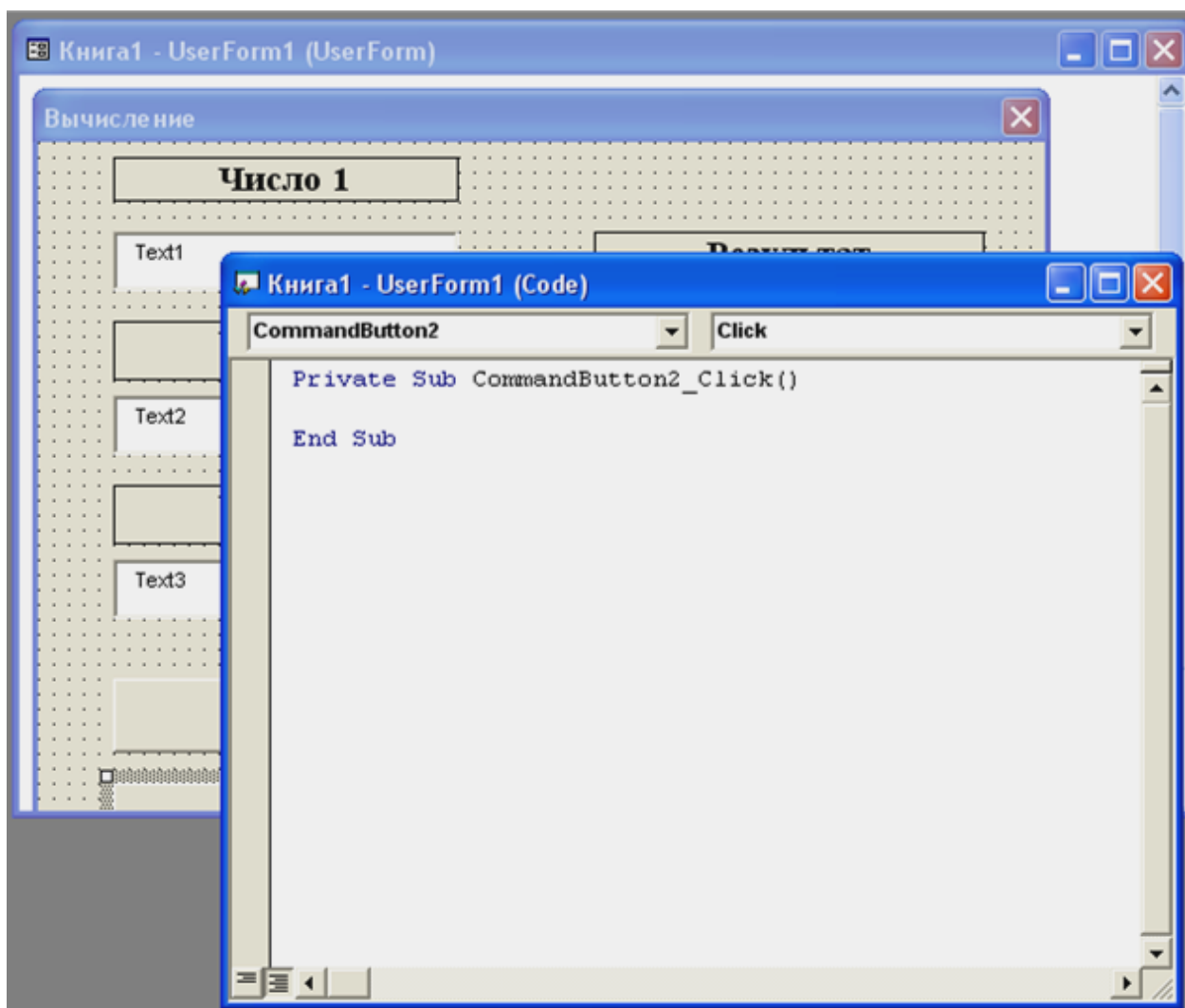


Рисунок 30 – Вид окна «Код»

Блок кода, связанный с частным объектом интерфейса, называется процедурой события VBA. Тело процедуры заключено между операторами, указывающими на начало и конец подпрограммы.

```
Private Sub CommandButton2_Click()  
End Sub
```

Операторы тела и процедуры выполняются каждый раз, когда пользователь активизирует элемент интерфейса, ассоциированный с процедурой. В данном случае событием является щелчок мыши (Click), но могут быть и события другого типа. (Их можно увидеть в ниспадающем списке, щелкнув по стрелке правее надписи Click).

Разработаем программный код для кнопок <Вычислить> и <END>.

1 Открыть список объектов в окне «Code». Все объекты интерфейса Вычислить появятся в списке, как показано на рисунке 31.

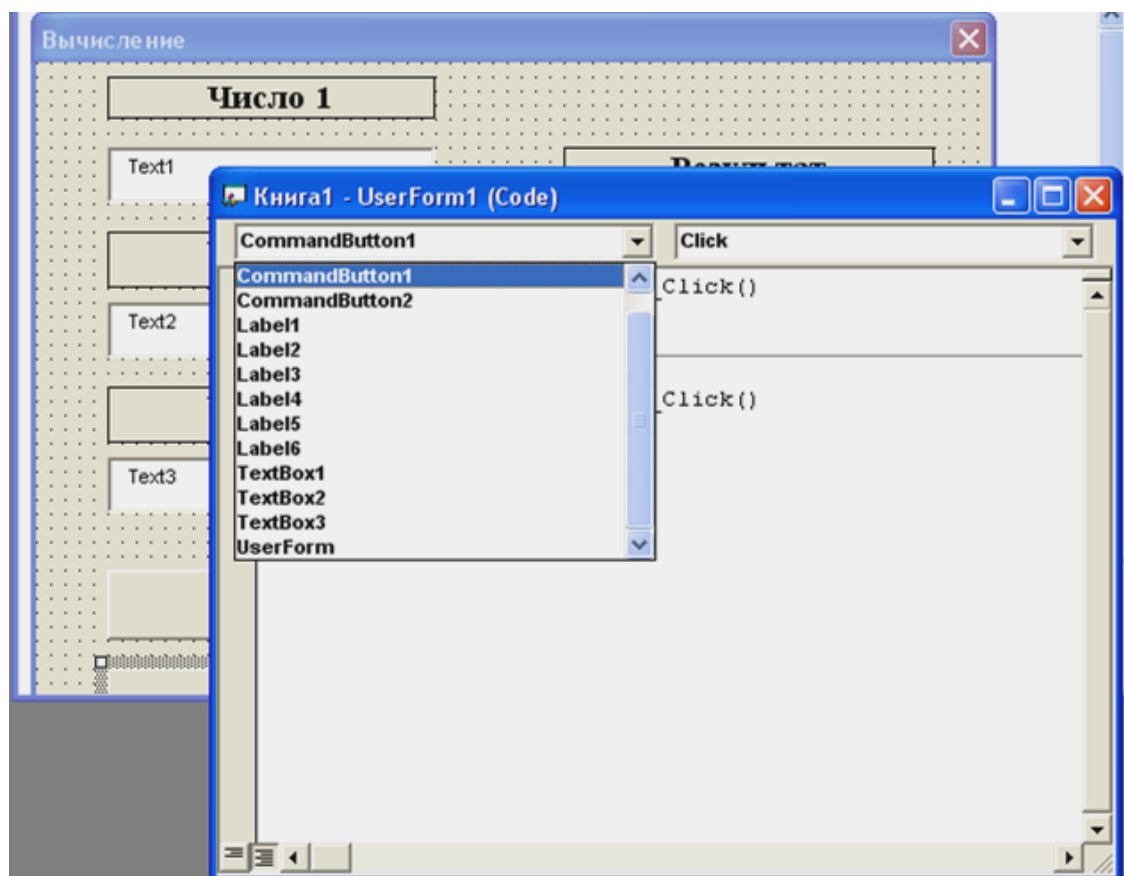


Рисунок 31 – Вид списка объектов

2 Щелкнуть по строке CommandButton1 в списке. Появится «пустая» процедура, связанная с кнопкой <CommandButton1>.

Несмотря на то, что название кнопки было изменено на <Вычислить>, в программе ее имя осталось прежним.

Следует иметь в виду, что у каждого объекта интерфейса может быть несколько ассоциированных с ним процедур – по одной для каждого связанного с этим объектом события. В данном случае для кнопки <CommandButton1> задано только одно событие – щелчок мышью.

3 Набрать следующие программные строки:

```
Private Sub CommandButton1_Click()  
Dim A, B, X, z1, z2, z3 As Single  
A = Val(TextBox1.Text)
```

```

B = Val(TextBox2.Text)
X = Val(TextBox3.Text)
z1 = Abs(Log(X) / Log(10)) - Sqr(Abs(Cos(X) - Exp(X)))
z2 = Abs(Tan(Abs(A * X - B)) / Sin(Abs(X)) + B)
z3 = Atn(z2 / Sqr(Abs(1 - z2 ^ 2)))
Label1.Caption = Log(Abs(z1 * z3))
End Sub

```

Примечание. Функция Val преобразует текстовый аргумент в числовое значение (по умолчанию данные, введенные в текстовое поле, представляются как обычный текст).

4 Дважды щелкнуть мышью по командной кнопке <END> на форме (или выбрать строку Command2 в окне «Code»).

5 Набрать программную строку процедуры, связанной с кнопкой <CommandButton2>. В данном случае это один оператор End. Он используется для останова программы и удаления ее с экрана (рисунок 32).

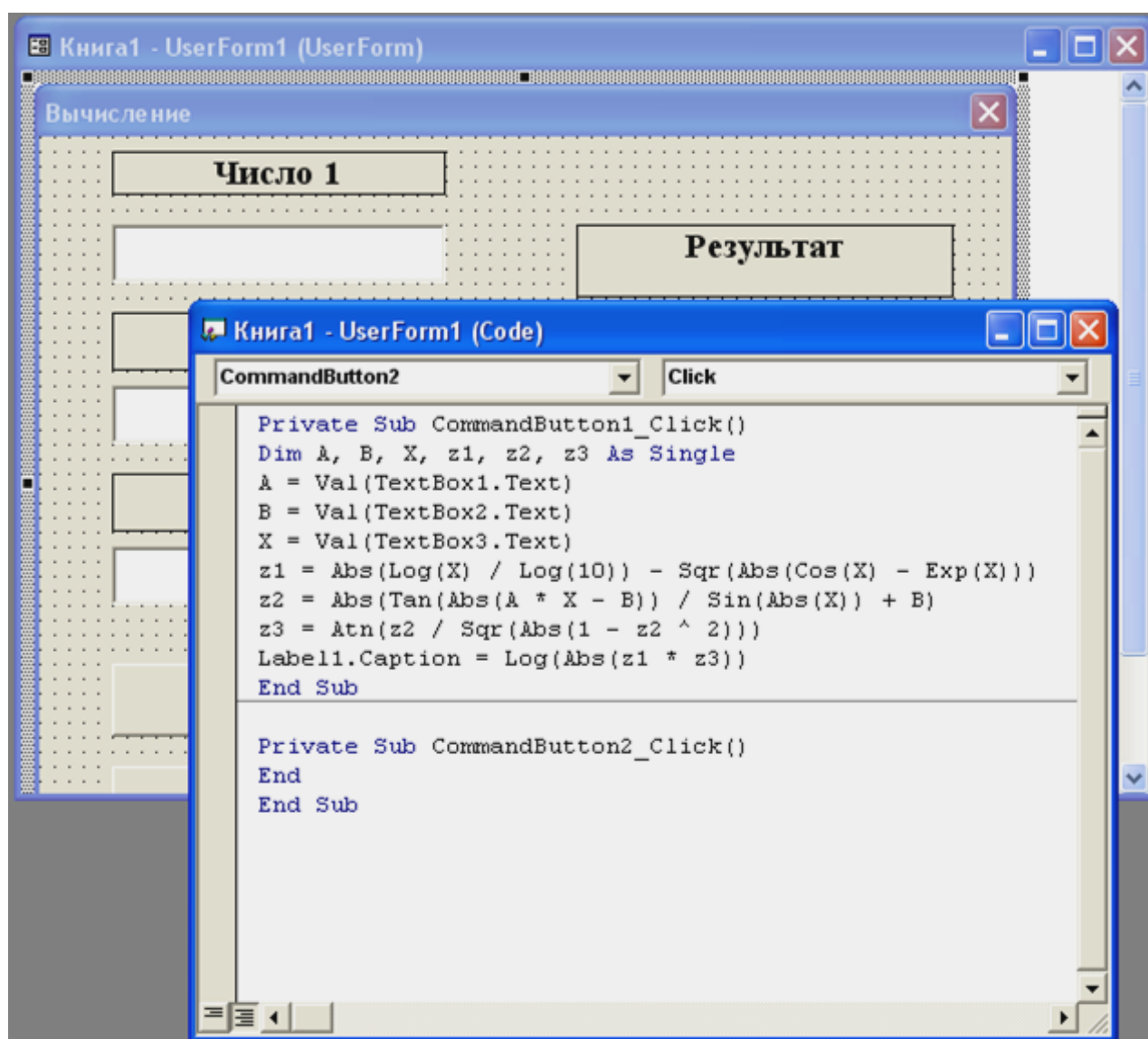


Рисунок 32 – Программный код, связанный с командной кнопкой 1 и 2



#### IV этап – Запуск программы

Запуск программы может быть осуществлен следующим образом:

1 Выбрать команду из меню Run→Run Sub/UserForm. Программа **Вычисление** запустится в среде программирования VBA. На экране появится разработанный программный интерфейс.

2 В поле ввода числа 1, где мигающий курсор, ввести произвольное число (целое или с дробной частью).

3 Нажать клавишу Tab или щелкнуть мышью на поле ввода числа 2. Мигающий курсор появится в поле числа 2.

4 Ввести второе произвольное число в поле 2.

5 Аналогично ввести третье число в поле 3.

6 Щелкнуть на кнопке <Вычислить>. В поле результата будет показан окончательный результат вычисления выражения.

7 Для продолжения работы программы, то есть для вычисления выражения при других входных данных, надо щелкнуть мышью на поле ввода числа 1 и ввести его.

8 Выполнить действия п.3-7 и т.д. Работа программы в подобном режиме диалога будет продолжаться до тех пор, пока не будет произведен щелчок мыши по кнопке <End>. В этом случае программа завершит работу, и среда программирования VBA вновь появится на экране.

#### V этап

Сохранить созданную программу создания приложения

Пример 2. Расчет стоимости с учетом НДС.

Постановка задачи. Создать диалоговое окно, в котором производится расчет стоимости товара с учетом НДС по его стоимости без НДС и значению ставки НДС.

Технология выполнения задания:

1 В редакторе Visual Basic в проект добавьте форму. На ней расположить три надписи, три поля ввода и две командные кнопки, как показано на рисунке 33.

Рисунок 33 – Диалоговое окно «Расчет стоимости»

2 При помощи окна Properties задать элементам управления значения следующим образом, как показано в таблице 26.

Таблица 26

Элемент управления	Свойство	Значение
Форма	Caption	Расчет стоимости
Надпись	Caption	Стоимость без учета НДС
Поле ввода	Name	TextBox1
Надпись	Caption	НДС
Поле ввода	Name	TextBox2
Надпись	Caption	Стоимость с учетом НДС
Поле ввода	Name	TextBox3
Кнопка	Name	CommandButton1
	Caption	OK
Кнопка	Name	CommandButton2
	Caption	Отмена

3 В модуле проекта набрать следующий код, изображенный на рисунке 34:

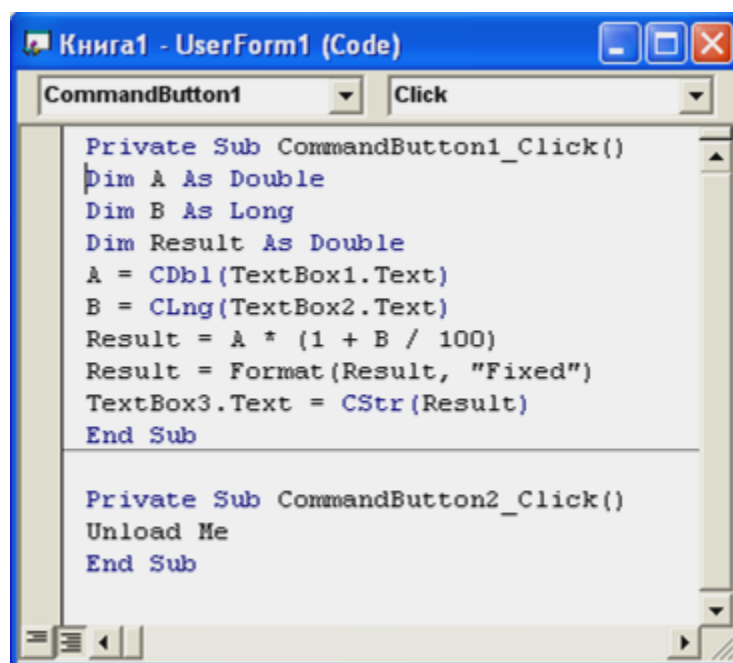


Рисунок 34 – Программный код

4 Запустить проект.

- нажать клавишу <F5>;
- в поле **Стоимость без учета НДС** ввести 1000;
- в поле **НДС** ввести 25;
- нажать кнопку ОК.

В поле **Стоимость с учетом НДС** будет выведена искомая стоимость. Для закрытия окна нажать кнопку Отмена.

## Лабораторная работа № 8

Разработка пользовательского приложения с помощью объектов Форм

Цель: приобретение навыков разработки пользовательского приложения с помощью объектов Формы, вычисляющее значение выражения в заданных точках, вводимых с клавиатуры. Индивидуальные варианты лабораторной работы № 8 выполняются в соответствии с индивидуальными вариантами лабораторной работы № 1 (4 часа).

### 3.2 Встроенные диалоговые окна в VBA

В проектах VBA часто встречаются две разновидности встроенных диалоговых окон: окна сообщений и окна ввода.

#### 3.2.1 Окна сообщений (MsgBox)

Выводит на экран диалоговое окно, содержащее сообщение, устанавливает режим ожидания нажатия кнопки пользователем, а затем возвращает значение типа Integer, указывающее, какая кнопка была нажата.

Окно сообщения является модальным по отношению к приложению. Это означает, что приложение не сможет продолжить работу, пока окно сообщения не будет закрыто. Иногда окно сообщения отображает пиктограмму или содержит несколько кнопок, с помощью которых пользователь может сделать выбор. В этом случае окно сообщения позволяет получить информацию о принимаемом пользователем решении.

Отобразить окно сообщения можно с помощью оператора MsgBox или функции MsgBox (). Разница между ними заключается в том, что функция отображает окно сообщения и возвращает некоторое значение, а оператор только отображает окно сообщения.

Синтаксис оператора MsgBox:

MsgBox <Строковое выражение> [ , целая константа] [ , строковое выражение]

Синтаксис функции MsgBox ():

Переменная = MsgBox (Строковое выражение [ , целая константа]  
[ , строковое выражение])

где

переменная – переменная целого типа, которой присваивается значение, возвращаемое функцией MsgBox ();

MsgBox – имя оператора/функции;

строковое выражение - строковое выражение, которое будет отображаться в поле сообщения окна сообщения (заключается в кавычки);

целая константа – или группа констант, определяющая, какие кнопки (комбинации кнопок) и пиктограммы будут отображаться в окне сообщения;

строковое выражение – строковое выражение, которое будет отображаться в строке заголовка окна сообщения (заключается в кавычки);

В окне сообщения можно выбрать одну из четырех пиктограмм. Эти пиктограммы и их свойства представлены в таблице 27.

Таблица 27 - Константы процедуры MsgBox, определяющие отображаемые информационные значки в диалоговом окне

Пиктограмма	Имя пиктограммы	Константа	Назначение
	vbCritical	Critical message (Критическое сообщение)	Указывает, что произошла серьезная ошибка. Часто после такого сообщения программа прекращает работу
	vbExclamation	Warning message (Предупреждение)	Сообщает о возникновении ошибки, которая требует исправления или может привести к нежелательным последствиям
	vbQuestion	Query (Запрос)	Указывает, что для продолжения работы программе требуется дополнительная информация от пользователя
	vbInformation	Information message (Информационное сообщение)	Информирует пользователя о состоянии программы. Чаще всего используется для сообщения о завершении выполнения некоторой задачи.

В функции MsgBox () можно использовать шесть наборов кнопок, задаваемых константами в соответствии с таблицей 28. Значение константы в соответствии с выбранным набором кнопок принимает аргумент Options.

Таблица 28 - Константы процедуры MsgBox, определяющие отображаемые кнопки в диалоговом окне

Набор кнопок	Значение	Константа
<ОК>	0	vbOKOnly
<ОК>, <Отмена>	1	vbOKCancel
<Стоп>, <Повтор>, <Пропустить>	2	vbAbortRetryIgnore
<Да>, <Нет>, <Отмена>	3	vbYesNoCancel
<Да>, <Нет>	4	vbYesNo
<Повтор>, <Отмена>	5	vbRetryCancel

Если требуется поместить в окно пиктограмму и набор кнопок, то аргумент Options набирается, как две константы, соединенные знаком «+». Первая константа выбирается из таблицы 27, вторая – из таблицы 28. Если используется несколько кнопок, то можно указать, какая из них будет кнопкой команды по умолчанию, т.е. кнопкой, которая сработает при нажатии клавиши [Enter] (кнопка, получающая фокус при отображении окна сообщения). В этом случае в аргумент Options добавляется третья константа в соответствии с таблицей 29.

Каждая из кнопок диалогового окна после щелчка на ней мышью во время выполнения программы возвращает соответствующее ей значение переменной (таблица 30).

Таким образом, имеется возможность программного определения, какая из кнопок диалогового окна нажата, и в соответствии с этим выполнить то или иное действие.

Таблица 29 - Константы процедуры MsgBox, определяющие основную кнопку в диалоговом окне

Кнопка команды по умолчанию	Значение	Константа
Первая	0	vbDefaultButton1
Вторая	256	vbDefaultButton2
Третья	512	vbDefaultButton3
Четвертая	768	vbDefaultButton4

Таблица 30 - Значения, возвращаемые функцией MsgBox

Кнопка	Значение	Константа
<ОК>	1	vbOK
<Отмена>	2	vbCancel
<Стоп>	3	vbAbort
<Повтор>	4	vbRetry
<Пропустить>	5	vbIgnore
<Да>	6	vbYes
<Нет>	7	vbNo

#### Пример 1.

Постановка задачи. Создать окно информационного сообщения с заголовком «Окно» и текстом «Нажмите кнопку» с пиктограммой информационного сообщения и с двумя кнопками <Да> и <Нет>. При нажатии кнопки <Да> надпись на командной кнопке <Command1> формы требуется заменить на значение Да, а при нажатой кнопки <Нет> на значение Нет.

Технология выполнения задания:

1 Запустить VBA.

2 В проект добавить форму. На панели инструментов выбрать элемент CommandButton и поместить на поле формы. Появится командная кнопка <CommandButton1>.

3 Дважды щелкнуть на командной кнопке <CommandButton1> и в окне программного кода между строками Private Sub и End Sub набрать следующие утверждения:

```
Private Sub CommandButton1_Click()  
Dim A As Long  
A = MsgBox ("Нажмите кнопку", vbInformation + vbYesNo, "Окно")  
If A = 6 Then  
    CommandButton1.Caption = "Да"  
ElseIf A = 7 Then  
    CommandButton1.Caption = "Нет"  
End If  
End Sub
```

Примечание.

В первой строке функция MsgBox выводит на экран диалоговое окно с текстом «Нажмите кнопку», двумя кнопками <Yes> и <No> и заголовком окна «Окно».

Вторая строка проверяет возвращаемое значение переменной A на его равенство числу 6.

Если условие выполняется, то это означает, что нажата кнопка <Yes>, и тогда в третьей строке свойству Caption (Надпись) командной кнопки <CommandButton1> присваивается значение Да.

Если возвращаемое значение не равно 6, то четвертая строка проверяет его на значение, равное 7, поскольку при нажатии кнопки <No>, возвращаемое значение A равно 7.

Если A действительно равно 7, то в пятой строке надписи на командной кнопке присваивается значение Нет.

Шестая строка закрывает условный оператор.

4 Запустить программу и проверить ее работоспособность, как показано на рисунке 35.

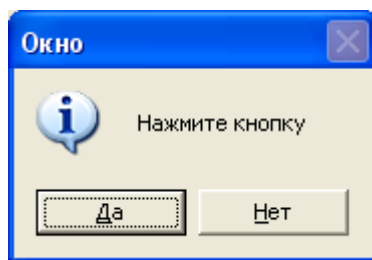


Рисунок 35 – Проверка работоспособности программы

### 3.2.2 Окна ввода

Если в программе необходимо получить какую-либо информацию от пользователя (ввести свое имя, имя файла, входные данные для расчетов и т.п.), можно воспользоваться функцией ввода.

Она выводит на экран диалоговое окно, содержащее сообщение, после ввода и две кнопки ОК и Cancel.

Устанавливает режим ожидания ввода текста пользователем и нажатия кнопки <ОК> или <Отмена>.

Если пользователь выбрал <ОК>, окно ввода возвращает значение типа String, находящийся в поле ввода.

Если же пользователь выбрал <Отмена>, окно ввода возвращает пустую строку (Empty) независимо от того, что напечатал пользователь.

Синтаксис функции:

Строковая переменная = InputBox (Строковое выражение [, константа заголовка] [, константа поля ввода] [, xPos] [, yPos])

где

Строковая переменная – переменная, в которой хранится значение, возвращаемое функцией InputBox ();

InputBox – имя функции;

строковое выражение – выражение, которое будет отображаться в области сообщения окна ввода;

константа заголовка – строковая константа или переменная, значение которой будет отображаться в заголовке окна ввода;

константа поля ввода – строковая константа или переменная, значение которой будет отображаться в поле ввода;

xPos, yPos – положение окна ввода соответственно по горизонтали и вертикали (в твипах).

Пример 2.

Постановка задачи. Иллюстрация использования окна ввода.

Технология выполнения задания:

1 Запустить VBA.

2 В проект добавить форму. На панели инструментов выбрать элемент CommandButton и поместить на поле формы. Появится командная кнопка <CommandButton2>.

3 Щелкнуть мышью дважды на командной кнопке <CommandButton2> и в окне программного кода между строками Private Sub и End Sub набрать следующие утверждения:

```
Private Sub CommandButton2_Click()  
A = InputBox ("Введите информацию", "Окно ввода")  
If A <> " " Then  
CommandButton2.Caption = A  
Else  
CommandButton2.Caption = "Строка ввода пуста"  
End If  
End Sub
```

Примечание.

Первая строка отображает окно ввода и присваивает возвращаемое значение переменной A.

Вторая строка проверяет содержимое поля ввода.

Если в поле набраны какие-либо данные, т.е. строка ввода непустая, то утверждение третьей строки после нажатия кнопки <ОК> меняет название кнопки <CommandButton2> на содержимое введенных данных.

В противном случае (строка 4) в пятой строке <CommandButton2> присваивается значение *Строка ввода пуста*.

4 Запустить программу и проверить ее работоспособность, как показано на рисунке 36.

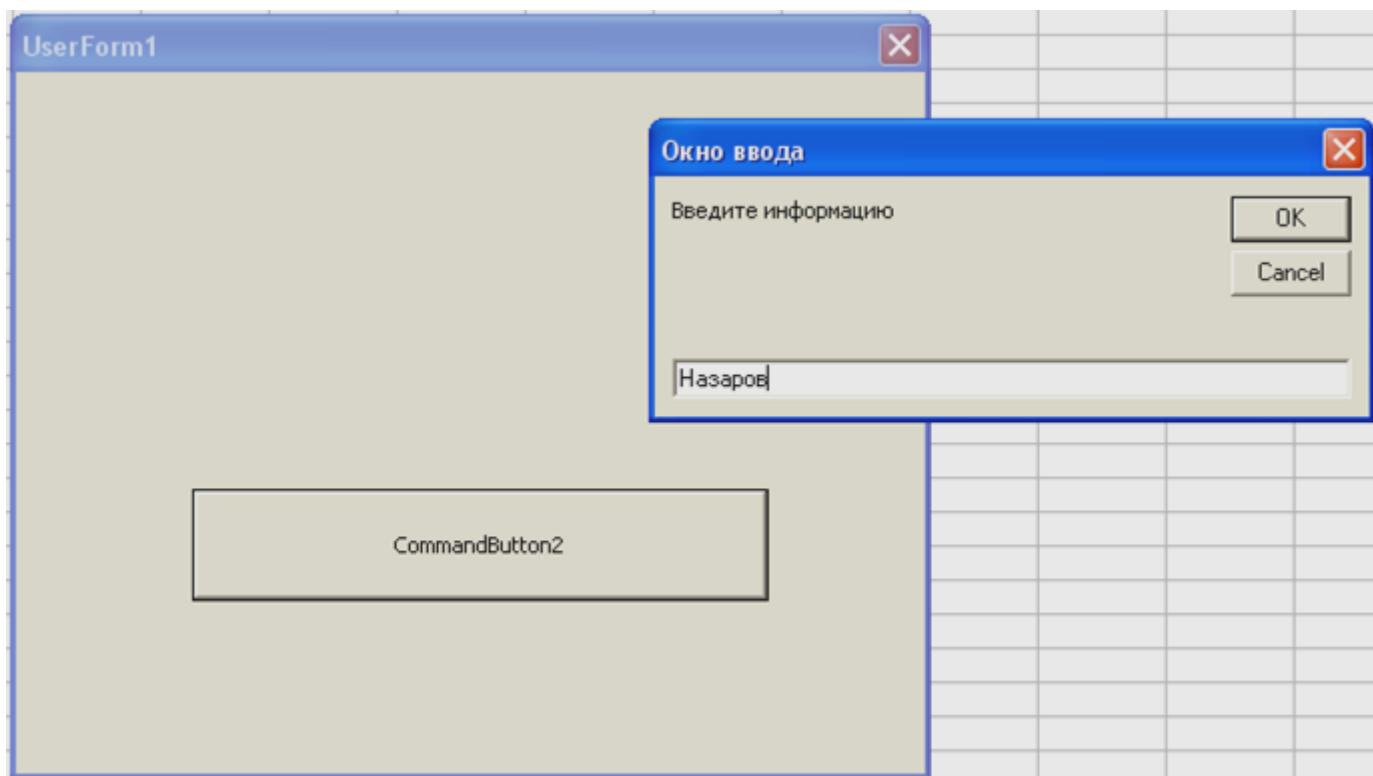


Рисунок 36 – Запуск созданной программы  
Лабораторная работа № 9

#### Встроенные диалоговые окна в VBA

Цель: приобретение навыков разработки проекта встроенных диалоговых окон в VBA. Индивидуальные варианты лабораторной работы № 9 представлены в таблице 31 (2 часа).

Таблица 31

№ В	<b>Варианты индивидуальных заданий</b>		
	Разработать проект диалоговых окон в VBA		
	Элемент управления	Свойство	Значение
1	2	3	4
1	Форма	Caption	Диалоговое окно1
	Кнопка (CommandButton1)	Caption	Критическое сообщение с тремя кнопками Стоп, Повтор, Пропустить



	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, Имя, Отчество, Факультет, Группу» в «Окно ввода»
	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint

Продолжение таблицы 31

1	2	3	4
2	Форма	Caption	Диалоговое окно2
	Кнопка (CommandButton1)	Caption	Предупреждение с одной кнопкой ОК
	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, Имя, Отчество, дату рождения» в «Окно ввода»
	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint
3	Форма	Caption	Диалоговое окно3
	Кнопка (CommandButton1)	Caption	Запрос с двумя кнопками ОК, Отмена
	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, адрес местожительства» в «Окно ввода»
	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint
4	Форма	Caption	Диалоговое окно4
	Кнопка (CommandButton1)	Caption	Информационное сообщение с двумя кнопками Повтор, Отмена
	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, паспортные данные» в «Окно ввода»
	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint
5	Форма	Caption	Диалоговое окно5
	Кнопка (CommandButton1)	Caption	Критическое сообщение с двумя кнопками Да, Нет
	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, домашний телефон» в «Окно ввода»

	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint
--	----------------------	---------	-----------------------------------------

Продолжение таблицы 31

1	2	3	4
6	Форма	Caption	Диалоговое окно6
	Кнопка (CommandButton1)	Caption	Предупреждение с двумя кнопками ОК, Отмена
	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, адрес местожительства» в «Окно ввода»
	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint
7	Форма	Caption	Диалоговое окно7
	Кнопка (CommandButton1)	Caption	Запрос с одной кнопкой ОК
	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, расписание занятий на понедельник В\Н» в «Окно ввода»
	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint
8	Форма	Caption	Диалоговое окно8
	Кнопка (CommandButton1)	Caption	Информационное сообщение с двумя кнопками Да, Нет
	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, перечень дисциплин, изучаемых в текущем семестре» в «Окно ввода»
	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint
9	Форма	Caption	Диалоговое окно9
	Кнопка (CommandButton1)	Caption	Предупреждение с тремя кнопками Стоп, Повтор, Пропустить
	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, Имя, Отчество, год рождения» в «Окно ввода»

	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint
10	Форма	Caption	Диалоговое окно10
	Кнопка (CommandButton1)	Caption	Запрос с тремя кнопками Да, Нет, Отмена

Продолжение таблицы 31

1	2	3	4
	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, Группу» в «Окно ввода»
	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint
11	Форма	Caption	Диалоговое окно11
	Кнопка (CommandButton1)	Caption	Информационное сообщение с одной кнопкой ОК
	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, адрес местожительства» в «Окно ввода»
	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint
12	Форма	Caption	Диалоговое окно12
	Кнопка (CommandButton1)	Caption	Критическое сообщение с двумя кнопками ОК, Отмена
	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, Факультет» в «Окно ввода»
	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint
13	Форма	Caption	Диалоговое окно13
	Кнопка (CommandButton1)	Caption	Информационное сообщение с тремя кнопками Да, Нет, Отмена
	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, адрес прописки» в «Окно ввода»
	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint
14	Форма	Caption	Диалоговое окно14
	Кнопка (CommandButton1)	Caption	Запрос с тремя кнопками Стоп, повтор, Пропустить

	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, Имя, расписание занятий на среду по Н/Н» в «Окно ввода»
	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint

Продолжение таблицы 31

1	2	3	4
15	Форма	Caption	Диалоговое окно15
	Кнопка (CommandButton1)	Caption	Предупреждение с двумя кнопками Повтор, Отмена
	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, Пол, группу крови» в «Окно ввода»
	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint
16	Форма	Caption	Диалоговое окно16
	Кнопка (CommandButton1)	Caption	Критическое сообщение с тремя кнопками Да, Нет, Отмена
	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, Имя, специальность» в «Окно ввода»
	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint
17	Форма	Caption	Диалоговое окно17
	Кнопка (CommandButton1)	Caption	Предупреждение с двумя кнопками Да, Нет
	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, Имя, № зачетной книжки» в «Окно ввода»
	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint
18	Форма	Caption	Диалоговое окно18
	Кнопка (CommandButton1)	Caption	Запрос с двумя кнопками Повтор, Отмена
	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, Имя, Отчество, паспортные данные» в «Окно ввода»

	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint
19	Форма	Caption	Диалоговое окно19
	Кнопка (CommandButton1)	Caption	Информационное сообщение с тремя кнопками Стоп, Повтор, Пропустить

Продолжение таблицы 31

1	2	3	4
	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, Имя, почтовый индекс» в «Окно ввода»
	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint
20	Форма	Caption	Диалоговое окно20
	Кнопка (CommandButton1)	Caption	Критическое сообщение с одной кнопкой ОК
	Кнопка (CommandButton2)	Caption	Ввод сообщения «Введите, пожалуйста свою Фамилию, Имя, Отчество, текущую дату и время» в «Окно ввода»
	Изображение (Image1)	Picture	Произвольный рисунок, созданный в Paint

### 3.3 Объекты формы в VBA: Кнопки-переключатели, Контрольные индикаторы, Рамки

Кнопки-переключатели. Элемент управления OptionButtons (Переключатель) позволяют пользователю выбрать одну или несколько взаимоисключающих альтернатив. Эти элементы управления обычно появляются группами по две или более, и все они взаимосвязаны (если нет рамок, в противном случае кнопки взаимосвязаны внутри рамок). Эти элементы управления состоят из кружка и текста, находящегося рядом. Свойство Alignment может быть установлено в 0 и 1 в зависимости от того, справа или слева от кружка должен находиться текст. Свойство Value может иметь установку True или False в зависимости от того, может быть выбрана кнопка или нет. В каждый момент времени может быть выбрана только одна кнопка, внутри такой кнопки появляется точка. Для выбранной кнопки свойство Value устанавливается в значение True, для всех других кнопок – в значение False.

Контрольные индикаторы (Флажок). Элемент управления CheckBox (Флажок) предоставляет пользователю возможность выбрать одну или несколько альтернатив, но независимо друг от друга. Флажок имеет два состояния: установлен и сброшен, но может настраиваться на выбор из трех альтернатив. В

результате пользователь может выбрать несколько контрольных индикаторов одновременно. Изображаются контрольные индикаторы маленькими квадратиками. Выбранный контрольный индикатор имеет внутри знак «☒» («галочку»).

Свойства контрольных индикаторов аналогичны, за исключением свойства Value, для которого возможными установками являются: 0 (не выбрано), 1 (выбрано), 2 (недоступно). При установке 2 контрольный индикатор и отметка обесцвечиваются.

Флажок обладает уникальным свойством TripleState, позволяющим производить выбор из трех альтернатив. Допустимыми значениями свойства TripleState являются:

- False (выбор из двух альтернатив True и False, т.е. флажок может находиться только в двух состояниях – установлен и сброшен);
- True (выбор из трех альтернатив True, False, Null, т.е. флажок может находиться в трех состояниях – установлен, сброшен и нейтрален).

Рамки. Элемент управления Frame (Рамка) предназначена для объединения элементов управления в группы. С функциональной точки зрения рамки необходимы в том случае, если на форме надо установить более одного набора кнопок-переключателей. Внутри рамки набор кнопок-переключателей работает как группа, независимо от других кнопок. Следует помнить, что сначала на форму добавляется рамка, а после этого внутри рамки добавляются кнопки-переключатели. Рамки также могут быть использованы для улучшения внешнего вида окна. Свойство рамки Caption хранит текст, размещенный в левом верхнем углу рамки.

Пример 1.

Постановка задачи. Разработать приложение нахождения минимального или максимального значения среди трех чисел, заданных формульными зависимостями  $\{a^2; b * a; a + b\}$  и среднеарифметического среди этих же трех чисел. Исходные данные вводятся в два поля. Результат должен отображаться после нажатия кнопки <ОК>. При выборе первого переключателя после нажатия кнопки ОК будет находиться минимальное число среди трех чисел, а при выборе второго – максимальное среди этих же трех чисел. С помощью контрольного индикатора (флажка) будет вычисляться среднеарифметическое среди полученных трех чисел. Программа должна заканчивать работу после нажатия кнопки <END>. Окно программы должно иметь вид, показанный на рисунке 37.

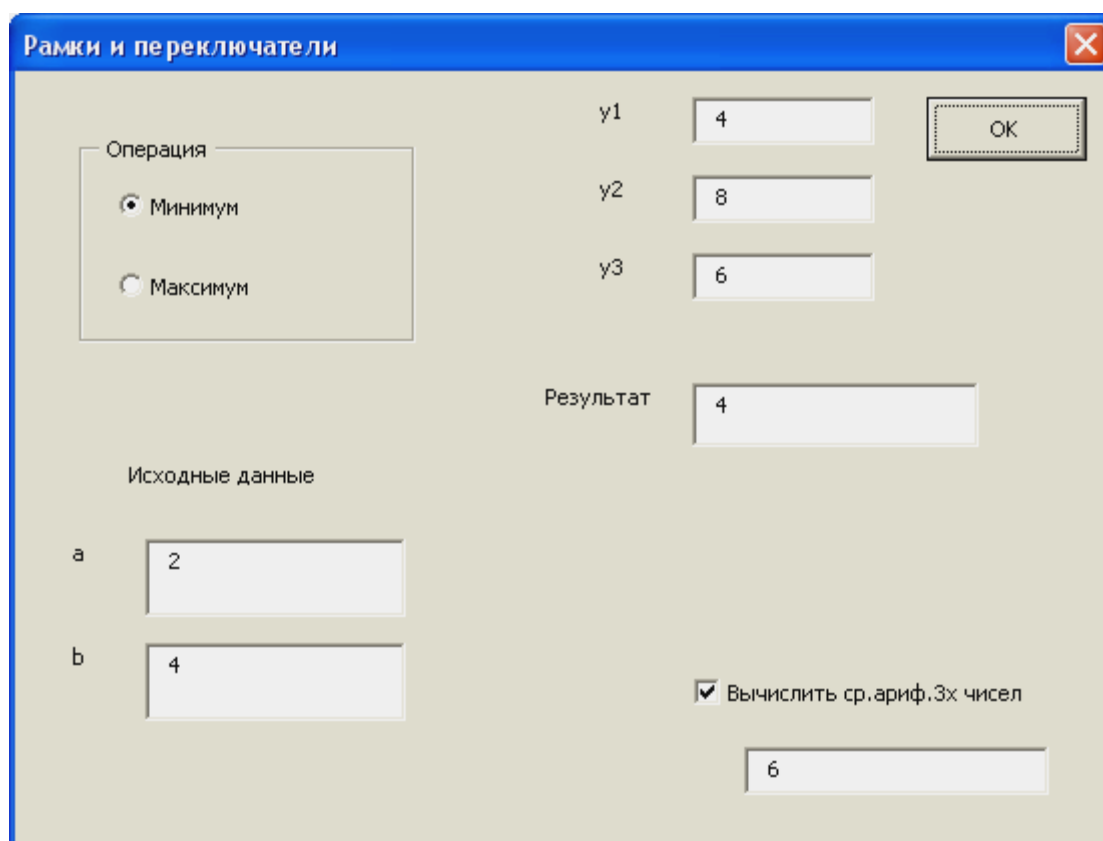


Рисунок 37 – Диалоговое окно программы, вычисляющее минимальное (максимальное) число среди трех полученных чисел

Технология выполнения задания:

1 Исходные данные:

$a, b \in \mathbb{Z}$

$a = 2; b = 4$

Результат:  $\text{Min (Max)} \in \mathbb{Z}$

2 Конструирование проекта.

Процесс программирования приложения **Рамки и переключатели** предусмотрен тремя этапами, как показано в таблице 32.

Таблица 32 - Этапы программирования

Этапы программирования	Число объектов
1. Создание пользовательского интерфейса	18 объектов
2. Установка свойств объектов	18 объектов
3. Составление программного кода	2 объекта

Создать форму с семью полями ввода, семью надписями, командной кнопкой, рамкой, внутри которой, как в контейнере, расположите два переключателя, с одним контрольным индикатором (флажком).

При помощи окна Properties установить значения свойств Name и Caption полей элементов управления следующим образом, как показано в таблице 33.

Таблица 33

Элемент управления	Свойство	Значение	Примечание
Поле ввода	Name	TextBox1	Число a
Поле ввода	Name	TextBox2	Число b
Поле ввода	Name	TextBox3	Число y1
Поле ввода	Name	TextBox4	Число y2
Поле ввода	Name	TextBox5	Число y3
Поле ввода	Name	TextBox6	Результат
Поле ввода	Name	TextBox7	Результат
Надпись	Caption	Label1	Исходные данные
Надпись	Caption	Label2	a
Надпись	Caption	Label3	b
Надпись	Caption	Label4	Y1
Надпись	Caption	Label5	Y2
Надпись	Caption	Label6	Y3
Надпись	Caption	Label7	Результат
Командная кнопка	Name Caption	CommandButton1	OK
Командная кнопка	Name Caption	CommandButton2	End
Рамка	Caption	Frame	Операция
Переключатель	Name Caption	OptionButton1	Минимум
Переключатель	Name Caption	OptionButton2	Максимум
Контрольный индикатор		CheckBox1	Вычислить среднеарифметическое среди 3-х чисел

3 В модуле формы набрать следующий код, как изображено на рисунке 38.



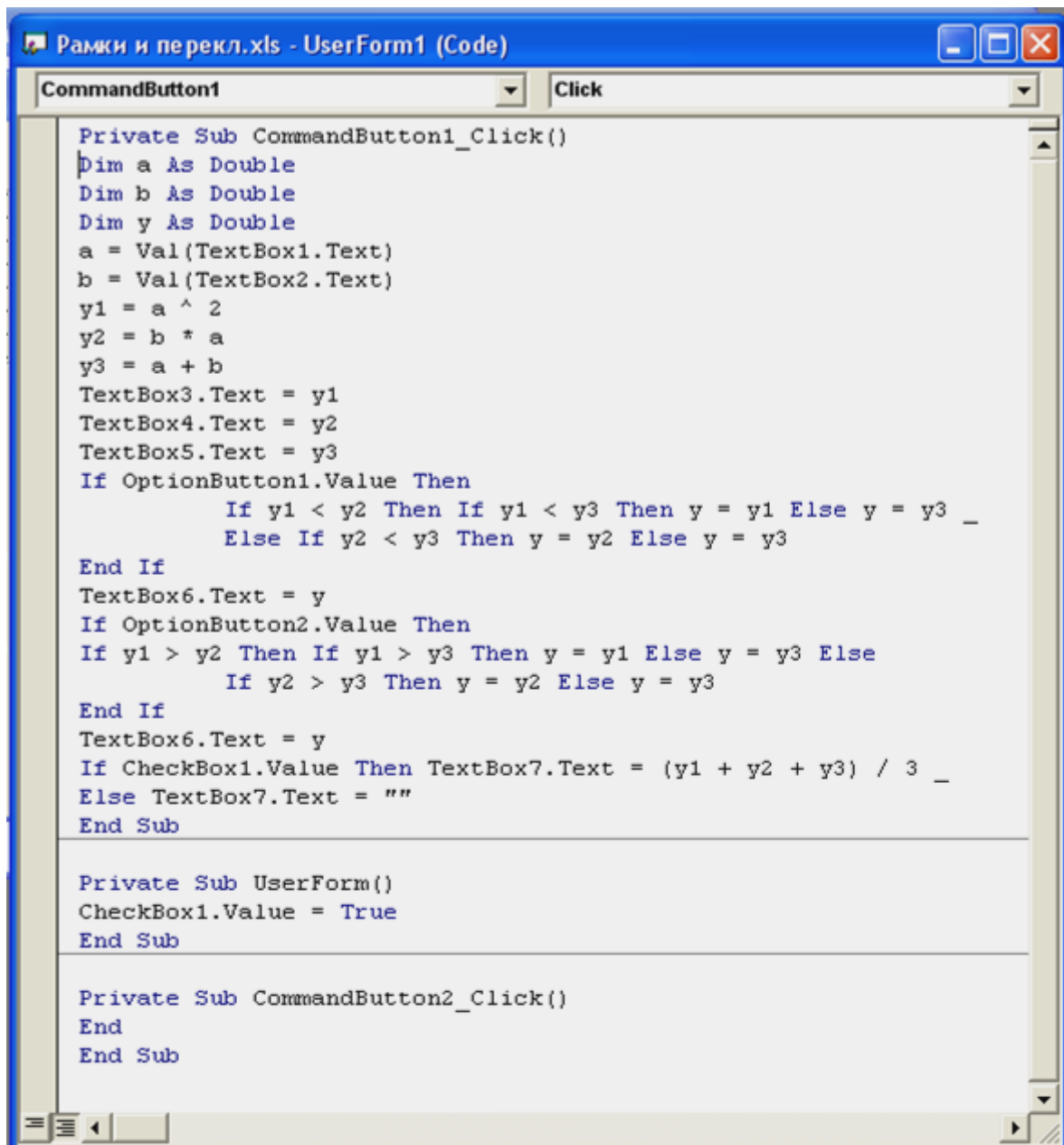


Рисунок 38 – Программный код

4 Запустить приложение.

Пример 2. Расчет ренты.

Постановка задачи. Создать проект для решения следующего типа задач по расчету ренты. Выкупается страховка, по которой выплачивается по 500 р. в конце каждого месяца в течение 20 последующих лет. Стоимость ренты составляет 60000 р. и выплачиваемые деньги принесут 8% годовых. Необходимо определить, будет ли выгодным инвестирование капитала.

Рисунок 39 – Окно «Рента». Флажок установлен

Таблица 34

Элемент управления	Свойство	Значение
Форма	Caption	Рента
Надпись	Caption	Стоимость ренты
Поле ввода	Name	TextBox1
Надпись	Caption	Процентная ставка, годовых
Поле ввода	Name	TextBox2
Надпись	Caption	Разовая выплата
Поле ввода	Name	TextBox3
Надпись	Caption	Срок, лет
Поле ввода	Name	TextBox4
Надпись	Caption	Настоящий объем вклада
Поле ввода	Name	TextBox5
Кнопка	Name	CommandButton1
	Caption	OK
Флажок	Name	CheckBox1
	Caption	Выплата в конце каждого месяца

Технология выполнения задания:

1 Создать форму, на которой расположить пять надписей и пять полей ввода, одну кнопку и флажок, как показано на рисунке 39.

2 При помощи окна Properties установить значения свойств Name и Caption элементов управления следующим образом, как показано в таблице 34.

3 Настоящий объем вклада находится финансовой функцией PV: PV (0.08/12, 12\*20, 500, , 0), которая возвращает -59 777,15 р.

Результат получает отрицательный, поскольку он представляет сумму, которую необходимо выплатить. Настоящий объем вклада (59777,15 р.) меньше, чем запрашиваемая цена (60000 р.). Следовательно, можно сделать вывод, что это не самый лучший способ инвестирования денег.

Если же выплаты производились в начале каждого периода, то настоящий объем PV (0.08/12, 12\*20, 500, , 1) возвращает 60 175,66 р. Это немного больше запрашиваемой цены и, поэтому, стоит подумать о предложении.

4 В модуле формы набрать следующий код, изображенный на рисунке 40.

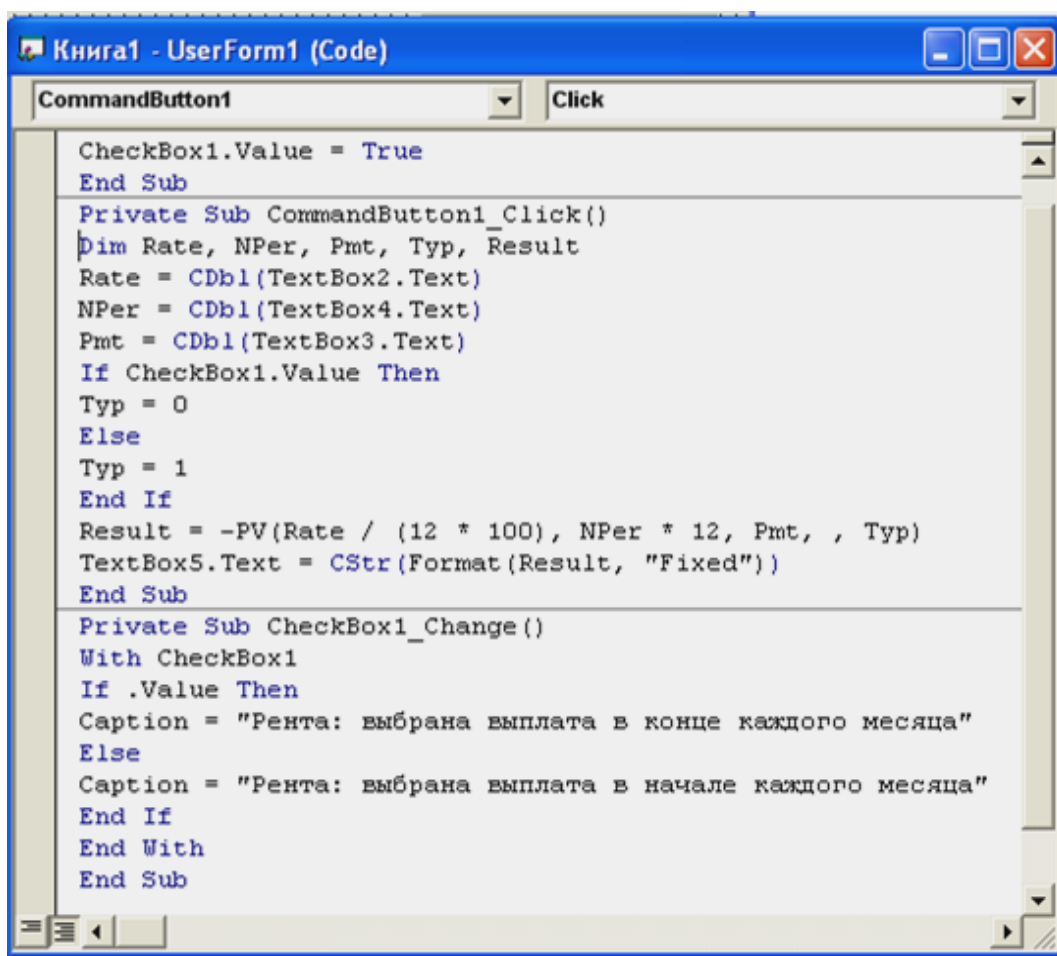


Рисунок 40 – Программный код

Примечание. Если флажок установлен, то в заголовке окна появится текст **Рента: выбрана выплата в конце каждого месяца**, а если сброшен, то **Рента: выбрана выплата в начале каждого месяца**. Таким образом, флажок в данном проекте не только фиксирует тот или иной выбор, но и управляет заголовком окна.

5 Запустить проект.

Разработка пользовательского приложения с помощью объектов: кнопки-переключателя, контрольного индикатора, рамки

Цель: приобретение навыков разработки проекта VBA с помощью объектов Формы - кнопка-переключатель, контрольный индикатор, рамка. Индивидуальные варианты лабораторной работы представлены в таблице 35 (2 часа).

Таблица 35

<b>№ В</b>	<b>Варианты индивидуальных заданий</b>
	Разработать проект VBA с помощью объектов: кнопки-переключателя, контрольного индикатора, рамки
1	$y = \max\left\{\sqrt{ x+1 } \cdot [2x^2 - 6(\ln x-1.83 ) + 0.18]; e^{ x -4}; 2\sin x\right\};$ x=-1.23
2	Даны действительные числа x, y, z. Вычислить: $\max\{x + y + z; x \cdot y \cdot z; x^2 + y^2 - z\};$
3	Даны действительные числа x, y, z. Вычислить: $\min^2\left\{x + y + \frac{z}{2}; x \cdot y \cdot z\right\} + 1$
4	$a = \max\left\{\frac{x^2 + b}{0.15}; be^x * \ln a + x ; b + x^3 - \frac{\cos 2x}{1 + x^2}\right\};$ b=0.251, x=0.004
5	$c = \max\{a, b, z\}$ где $a = x^3; b = e^x; z = \begin{cases} \frac{a-b}{0.2} + \frac{x^2 - (a-b)}{\sqrt{x} + b} + \frac{b-a}{x}, & 1 > x > 0 \\ \ln(e^x + 2.36), & \text{в остальных случаях} \end{cases};$ x=1.14
6	$y = \max\{a, b, x\}; \text{ где } a = \sin(2\pi), b = \cos(2\pi), x = \frac{e^{\ln(a)}}{\operatorname{tg}(a)}$
7	$y = \max\left\{\frac{\cos(2x + \pi) - \sin \frac{2x^2 - 3}{5x + 2}}{\ln \left  \operatorname{arctg} \frac{2x}{3} \right }; e^{\frac{\cos 2x}{8}} - \sqrt[3]{\cos 2x - \ln x-1 }; e^x\right\};$ x=0.15394

Продолжение таблицы 35

8	$t = \min \left\{ \frac{x}{\ln 3-2x +b^2}; e^{a^2-1.2} - x; e^{b^2+0.1} - x \right\};$ $a=1.618, b=0.992, x=0.018$
9	$z = \min \{ x^{0.64} - y^{0.13}; x + \ln y ; e^{x+y} \};$ $y=0.1465, x=2.3864$
10	$y = \frac{az^2 + bz - e^z}{ z+a  \cdot \ln(z+b) -} \text{ где } z = \min \left\{ ax^2; b^2x; \frac{\cos^2(2b)}{1+\ln^2(x)} \right\};$ $a=0.647, b=1.546, x=1.667$
11	$y = \min \{ \cos(a+x); \sin(ax); \operatorname{tg}(a-x) \};$ $a=0.361, x=0.852$
12	$a = \min \left\{ e^{-cx}; \frac{c}{2 + \sin^2 cx}; \frac{2x+3}{c} \right\};$ $c=0.983, x=0.89$
13	$c = \max \left\{ ax^2; bx^2 - a; \frac{e^{x^2+ab} + e^{\sin a}}{\sqrt{\left( \ln \left  \frac{b^3-2}{a+b} \right  + 18 \right)^3 + 1}} \right\};$ $a=0.185, b=0.256, x=0.246$
14	$l = \max \left\{ \frac{ax^2+b}{2a+1}; \frac{bx^2+a}{2b+1}; \frac{\cos^2(a)}{2 * \operatorname{tg}(b)} \right\}, \text{ где } b = \{ ae^{-x} \}, a = \left\{ \frac{x^2+1}{2.56} \right\};$ $x=0.29$
15	$y = \min \left\{ \frac{1}{\sin \ln x - b }; \ln \left  \operatorname{tg} \left( \frac{ab}{2} + \frac{\pi}{4} \right) \right ; \sqrt{\ln^3 x - \cos x} \right\};$ $a=0.357, b=0.128, x=0.634$
16	$z = \max \left\{ \frac{x^3+b}{\cos x-a }; \sin \left( \ln \sqrt{ x - \sqrt{ax-b} } \right); \arccos(\ln x ) \right\};$ $x=0.324, a=0.421, b=0.722$

Продолжение таблицы 35

17	$t = \min \left\{ \frac{\cos^2(x-b)}{\ln x }; \arccos(\operatorname{tg}(x)); \sqrt[3]{\frac{b^5 + 2 * \ln b }{x}} \right\};$ <p style="text-align: center;">где <math>x = 0.256, b = 0.147</math></p>
18	$y = \max \left\{ \frac{\cos^3(2a + \ln x) - \sin \frac{5x^2}{2}}{\ln \arctg 2a }; e^{\cos 2\pi} - \sqrt[4]{ \ln a-1  }; e^x \right\};$ <p style="text-align: center;">где <math>a = 0.504, x = 0.112</math></p>
19	$t = \min \left\{ \frac{x^{\cos(4\pi)}}{\operatorname{tg}^2 3-2x +b^2}; e^{\ln a-1.2} - x; e^{b^2+0.1} - x \right\};$ <p style="text-align: center;">где <math>a = 0.125, b = 1.320, x = 1.257</math></p>
20	$y = \max \left\{ \arccos^2(\ln a+x ); \arcsin^2\left(\frac{a}{x}\right); \operatorname{tg}^3(a-x) \right\};$ <p style="text-align: center;">где <math>a = 0.849, x = 0.560</math></p>

### 3.4 Объекты формы в VBA: Полоса прокрутки и Счетчик

Элемент управления ScrollBar (Полоса прокрутки) предлагает установить числовое значение, основываясь на положение ползунка, причем этот элемент может устанавливать только целые значения. Полоса прокрутки часто находится в текстовых полях, но ее можно добавлять и в другие места. Как правило, она может быть присоединена к краям окон, например, при выводе на экран части растрового рисунка.

Основным событием элемента управления Полоса прокрутки является событие Change, генерируемое при изменении текущего значения полосы. В таблице 36 перечислены основные свойства, которыми может обладать объект «Полоса прокрутки».

Таблица 36 - Основные свойства полосы прокрутки

Свойство	Описание
Value	Возвращает или устанавливает текущее значение ползунка на полосе прокрутки
Min	Минимальное значение полосы прокрутки
Max	Максимальное значение полосы прокрутки
SmallChange	Устанавливает шаг изменения значения при щелчке по одной из стрелок полосы прокрутки. Это свойство определяет величину, на которую изменится Value при щелчке по стрелке на конце линейки прокрутки.

Продолжение таблицы 36

LargeChange	Устанавливает шаг изменения при щелчке между ползунком и одной из стрелок полосы прокрутки
Orientation	Устанавливает ориентацию полосы прокрутки. Допустимые значения: - fmOrientationAuto или -1 (ориентация зависит от размера элемента управления. Используется по умолчанию); - fmOrientationVertical или 0 (вертикальное расположение); - fmOrientationHorizontal или 1 (горизонтальное расположение)

Элемент управления SpinButton (Счетчик) можно рассматривать, как полосу прокрутки без ползунка, и он выполняет те же функции, что и полоса прокрутки. Он имеет те же свойства Value, Min, Max, SmallChange, что и полоса прокрутки.

Пример 1.

Постановка задачи. Разработать программу вычисления значения выражения, состоящее из вычисления произведения конечного ряда  $P = \prod_{i=1}^{10} \frac{i * (i + a)}{i^2 + a^2} + \prod_{k=2}^4 \frac{k^3}{k + a}$ . Вводить пользователем с клавиатуры в одно (два) поле с именем «Исходное число А» («Исходное число В» и т.д.) при установке начального и конечного значения параметра цикла с помощью счетчика и полосы прокрутки, с отображением результата в поле «Результат».

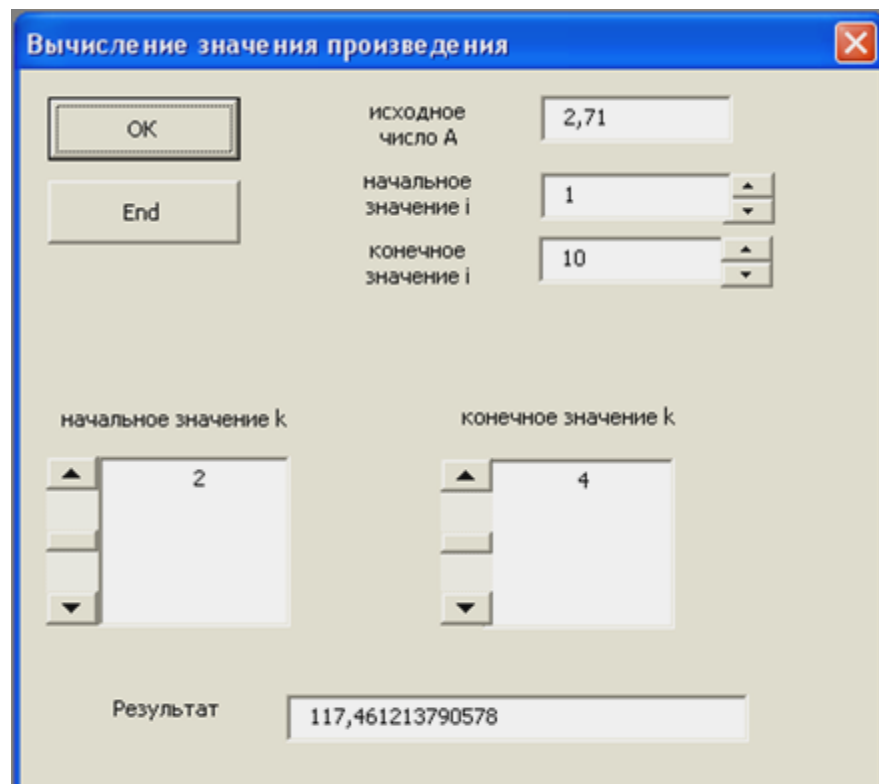


Рисунок 41 – Окно программы «Вычисление значения произведения»

Результат должен отображаться после нажатия кнопки <OK>. Программа должна заканчивать работу после нажатия кнопки <END>. Окно программы должно иметь вид, показанный на рисунке 41.

Технология выполнения задания:

1 Исходные данные:

$i, k \in \mathbb{Z}$

$a \in \mathbb{R}, a = 2,71$

Результат:  $P \in \mathbb{R}$ .

2 Процесс программирования приложения Вычисление значения произведения предусмотрен тремя этапами, как показано в таблице 37.

Таблица 37 - Этапы программирования

Этапы программирования	Число объектов
1. Создание пользовательского интерфейса	18 объектов
2. Установка свойств объектов	12 объектов
3. Составление программного кода	2 объекта

3 При помощи окна Properties установить значения свойств Name и Caption элементов управления, как указано в таблице 38.

Таблица 38 – Свойства элементов управления

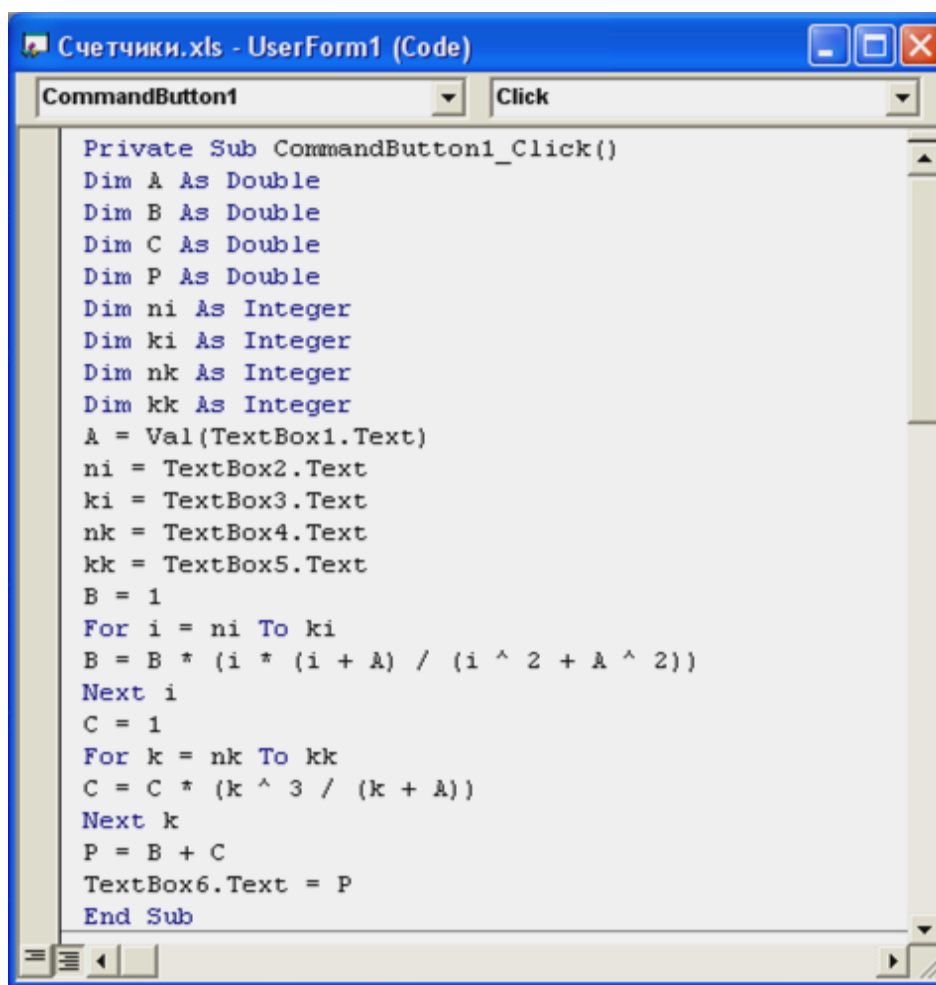
Элемент управления	Свойство	Значение
Форма	Caption	Вычисление значения произведения
Надпись	Caption	Исходное число A
Поле ввода	Name	TextBox1
Надпись	Caption	Начальное значение i
Поле ввода	Name	TextBox2
Счетчик	Name	SpinButton1
Надпись	Caption	Конечное значение i
Поле ввода	Name	TextBox3
Счетчик	Name	SpinButton2
Надпись	Caption	Начальное значение k
Поле ввода	Name	TextBox4
Полоса прокрутки	Name	ScrollBar1
Надпись	Caption	Конечное значение k
Поле ввода	Name	TextBox5



Продолжение таблицы 38

Полоса прокрутки	Name	ScrollBar2
Надпись	Caption	Результат
Поле ввода	Name	TextBox6
Кнопка	OK	CommandButton1
Кнопка	END	CommandButton2

4 В модуле формы набрать следующий код, изображенный на рисунках 42а, 42б, 42в.



```

Private Sub CommandButton1_Click()
    Dim A As Double
    Dim B As Double
    Dim C As Double
    Dim P As Double
    Dim ni As Integer
    Dim ki As Integer
    Dim nk As Integer
    Dim kk As Integer
    A = Val(TextBox1.Text)
    ni = TextBox2.Text
    ki = TextBox3.Text
    nk = TextBox4.Text
    kk = TextBox5.Text
    B = 1
    For i = ni To ki
        B = B * (i * (i + A) / (i ^ 2 + A ^ 2))
    Next i
    C = 1
    For k = nk To kk
        C = C * (k ^ 3 / (k + A))
    Next k
    P = B + C
    TextBox6.Text = P
End Sub

```

Рисунок 42а – Программный код кнопки «ОК»

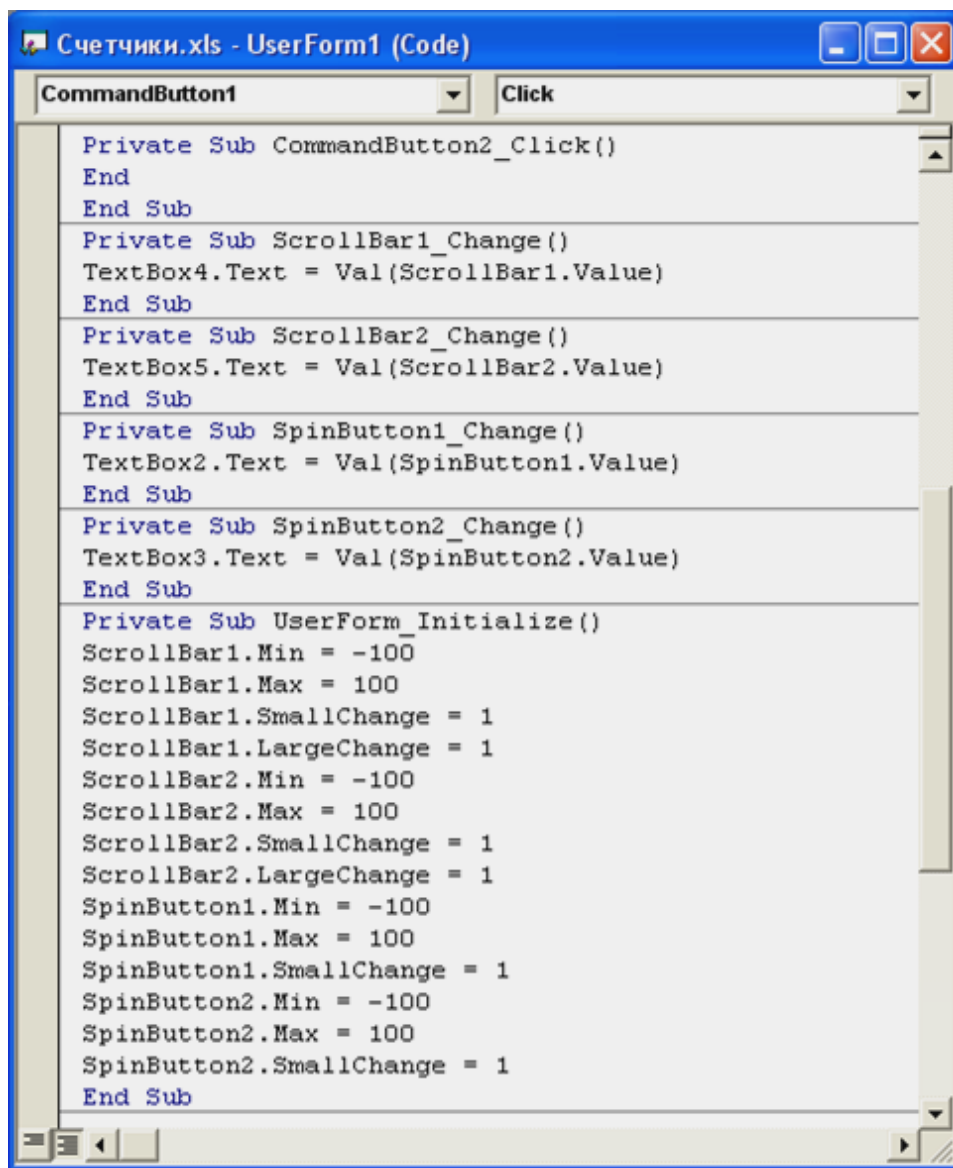


Рисунок 426 – Программный код кнопки «End», «Счетчика1» («Счетчика2»), «Полосы прокрутки1», («Полосы прокрутки2»), инициализации формы

Обратить внимание на две группы процедур:

1 ScrollBar1\_Change () (ScrollBar2\_Change ()), SpinButton1\_Change () (SpinButton2\_Change ()).

2 TextBox2\_Change () (TextBox3\_Change (), TextBox4\_Change (), TextBox5\_Change ()).

Первая из них обрабатывает событие Change счетчика (полосы прокрутки), при изменении значения которого его текущее значение отображается в поле ввода **Начальное значение к, Конечное значение к, Начальное значение i, Конечное значение i**.

Пользователь может ввести значение в поле ввода **Начальное значение к, Конечное значение к, Начальное значение i, Конечное значение i** с клавиатуры. В этом случае, текущее значение счетчика должно быть автоматически установлено тому значению, которое введено в это поле ввода. Как раз это и делает вторая группа процедур, обрабатывая событие Change поля ввода **Начальное**

значение **k**, Конечное значение **k**, Начальное значение **i**, Конечное значение **i**. Таким образом, обе процедуры синхронизируют работу поля ввода и счетчика.

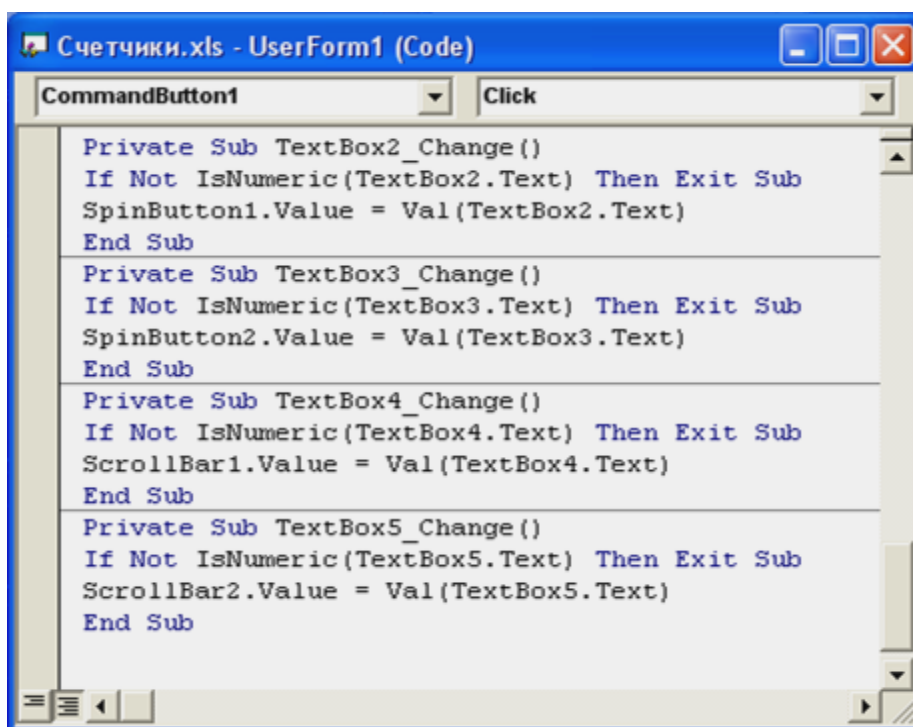


Рисунок 42в – Программный код процедур полей ввода «Начальное значение **i**», «Конечное значение **i**», «Начальное значение **k**», «Конечное значение **k**»

Пример 2. Определение размера платежей с целью накопления искомой суммы.

Постановка задачи. Предположим, что за 18 лет необходимо иметь в распоряжении 50000 р. С этой целью решено откладывать постоянную сумму каждый месяц. Если предположить, что удастся обеспечить 6% годовых, сколько денег надо откладывать каждый месяц?

Технология выполнения задания:

1 Создать форму, на которой расположить четыре надписи, четыре поля ввода, одну кнопку, счетчик, как показано на рисунке 43.

2 При помощи окна Properties установить значения свойств Name и Caption элементов управления следующим образом, как показано в таблице 39.

Рисунок 43 – Диалоговое окно «Размер платежей»

Таблица 39 – Свойства элементов управления

Элемент управления	Свойство	Значение
Форма	Caption	Размер платежей
Надпись	Caption	Искомая сумма
Поле ввода	Name	TextBox1
Надпись	Caption	Процентная ставка, годовых
Поле ввода	Name	TextBox2
Надпись	Caption	Срок, лет
Поле ввода	Name	TextBox3
Надпись	Caption	Размер ежемесячных выплат
Поле ввода	Name	TextBox4
Кнопка	Name	CommandButton1
	Caption	OK
Счетчик	Name	SpinButton1

3 Определение размера платежей с целью накопления искомой суммы находится при помощи финансовой функции Pmt.

Pmt (0.06/12, 18\*12, 0, 50000), которая возвращает -129,08 р.

Таким образом, каждый месяц надо откладывать по 129,08 р.

4 В модуле формы набрать следующий код, показанный на рисунке 44.

Примечание. Обратите внимание на две процедуры: SpinButton1\_Change и TextBox2\_Change. Первая из них обрабатывает событие Change счетчика, при изменении значения которого его текущее значение отображается в поле ввода **Процентная ставка, годовых**.

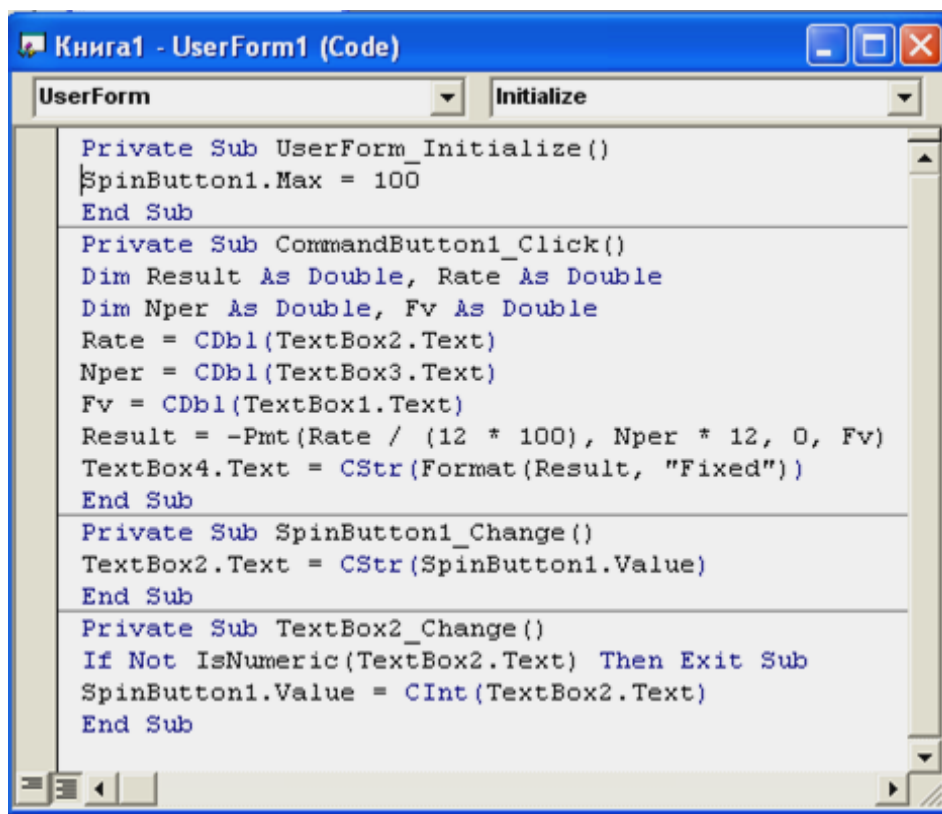


Рисунок 44 – Программный код

Можно ввести значение в поле ввода **Процентная ставка, годовых** с клавиатуры. В этом случае, текущее значение счетчика должно быть автоматически установлено тому значению, которое введено в это поле ввода. Как раз это и делает вторая процедура, обрабатывая событие Change поля ввода **Процентная ставка, годовых**.

Таким образом, обе процедуры синхронизируют работу поля ввода и счетчика.

5 Запустить проект.

#### Лабораторная работа № 11

Разработка пользовательского приложения с помощью объектов: полоса прокрутки, счетчик

Цель: приобретение навыков разработки пользовательского приложения с помощью объектов формы - полоса прокрутки, счетчик. Индивидуальные варианты лабораторной работы № 11 выполняются в соответствии с индивидуальными вариантами лабораторной работы № 4 (2 часа).

### 3.5 Объекты формы в VBA: Список и Поле со списком

Элемент управления ListBox (Список) предназначен для отображения и хранения списка значений. В списке пользователь может выбрать одно или несколько значений, которые в последующем используются в тексте программы. Обычно выбор элемента из списка производится щелчком по элементу. Двойной щелчок по элементу применяется с целью выполнения каких-либо действий в

программе, связанных с этим элементов. В таблице 40 приведены основные свойства объекта «Список».

Таблица 40 - Свойства объекта «Список»

Свойство	Описание
ListIndex	Возвращает номер выбранного элемента списка. Нумерация элементов списка начинается с нуля. Если ни один элемент списка не выбран, то возвращает -1
ListCount	Возвращает число элементов списка
TopIndex	Возвращает элемент списка с наибольшим номером
ColumnCount	Устанавливает число столбцов в списке
TextColumn	Устанавливает столбец в списке, элементы из которого возвращаются в качестве значения свойства Text
Text	Возвращает выбранной в списке элемент
List	Возвращает элемент списка, стоящий на пересечении указанной строки и столбца. Синтаксис: List (row, column) Row – строка Column - столбец
RowSource	Устанавливает диапазон, содержащий элементы списка
ControlSource	Устанавливает диапазон (ячейку), куда возвращается выбранный элемент из списка
MultiSelect	Устанавливает способ выбора элементов списка. Допустимые значения: fmMultiSelectSingle или 0 (выбор только одного элемента); fmMultiSelectMulti или 1 (разрешен выбор нескольких элементов, выбор осуществляется либо щелчком, либо нажатием клавиши <Пробел>); fmMultiSelectExtended или 2 (разрешено использование клавиши <Shift> при выборе ряда последовательных элементов списка)
Selected	Логическое свойство, которое возвращает значение True, если элемент списка выбран, и False – в противном случае. Используется для определения выбранного элемента, когда значение свойства MultiSelect установлено равным fmMultiSelectMulti или fmMultiSelectExtended
ColumnWidths	Устанавливает ширину столбцов списка. Синтаксис: ColumnWidths = String где String – строка, устанавливающая ширину столбцов в пунктах. В следующем примере устанавливается ширина каждого из трех столбцов списка: With ListBox1 .ColumnCount = 3 .ColumnWidths = "20:30:30" End With

Продолжение таблицы 40

ColumnHeads	Логическое свойство, определяющее, выводить ли в списке заголовки столбцов
ListStyle	Устанавливает способ выделения элементов. Допустимые значения: fmListStylePlain или 0 (выбранный элемент из списка выделяется цветом); fmListStyleOption или 1 (перед каждым элементом в списке располагается флажок, и выбор элемента из списка соответствует установке флажка)
MathEntry	Выводит первый подходящий элемент из списка при выборе его имени с клавиатуры. Допустимые значения: fmMathEntryFirstLetter или 0 (выводит подходящий элемент по набранной первой букве. В этом случае, предпочтительно, чтобы элементы списка были упорядочены в алфавитном порядке); fmMathEntryComplete или 1 (выводит подходящий элемент по полному набранному имени); fmMathEntryNone или 2 (режим вывода подходящего элемента в списке отключен)
BoundColumn	Устанавливает данные, возвращаемые свойством Value. Допустимые значения: 0 (свойством Value возвращается индекс выбранной строки, т.е. в этом случае оно действует как свойство ListIndex); От 1 до количества столбцов в списке (свойством Value возвращается элемент из выбранной строки, стоящий в столбце, заданным значением свойства BoundColumn)

### 3.5.1 Заполнение списка и удаление его элементов

Для поэлементного заполнения списка и удаления из него выбранного элемента применяются методы Clear, RemoveItem и AddItem.

Метод Clear удаляет все элементы из списка.

Метод RemoveItem удаляет из списка элемент с указанным номером.

Синтаксис:

RemoveItem (index)

где

index – номер удаляемого элемента из списка.

Примечание. Этот метод не работает, если список заполнен элементами из диапазона при помощи свойства RowSource.

Метод AddItem добавляет элемент в список.

Синтаксис:

AddItem ([item [, varIndex]])

где

item – элемент (строковое выражение) добавляемый в список;

varIndex – номер добавляемого элемента.

Метод `AddItem` позволяет заполнять список как в цикле, так и поэлементно.  
Пример 1.

Постановка задачи. Создать форму, которая добавляет и удаляет список. Расположить в ней три командные кнопки и один список.

Технология выполнения задания:

1 При помощи окна `Properties` установить значения свойств `Name` и `Caption` элементов управления следующим образом, как показано в таблице 41.

Таблица 41 – Свойства элементов управления

Элемент управления	Свойство	Значение
Форма	<code>Caption</code>	Список
Кнопка	<code>Name</code> <code>Caption</code>	Добавить
Кнопка	<code>Name</code> <code>Caption</code>	Удалять из списка
Кнопка	<code>Name</code> <code>Caption</code>	Удалить все
Список	<code>Name</code>	<code>ListBox1</code>

2 В модуле формы набрать следующий код, изображенном на рисунке 45.

3 Проект готов (рисунок 46).

Примечание. В этом проекте при нажатии на кнопку `Add` в список добавляется новый элемент. При нажатии кнопки `Remove` удаляется выделенный элемент, а при нажатии кнопки `Clear` – весь список очищается.

### 3.5.2 Список с несколькими столбцами

Для создания многоколоночного списка следует установить значение свойства `ColumnCount`, которое и задает, сколько столбцов имеется в списке. Кроме того, при работе с подобным списком полезно определить значение свойства `ColumnWidths`, которое устанавливает ширину каждого столбца. А затем список надо заполнить либо посредством массива, либо поэлементно.

В качестве примера создания списка с несколькими столбцами рассмотрим проект, в котором производится табулирование функции  $\sin(x)$  на промежутке от  $a$  до  $b$  с шагом  $h$ , а результат табуляции (значения аргумента и функции) выводится в список.

Пример 1.

Постановка задачи. Создать форму, позволяющую табулировать функцию  $\sin(x)$ . Расположить три надписи, три поля ввода, одну кнопку и список, как показано на рисунке 47.



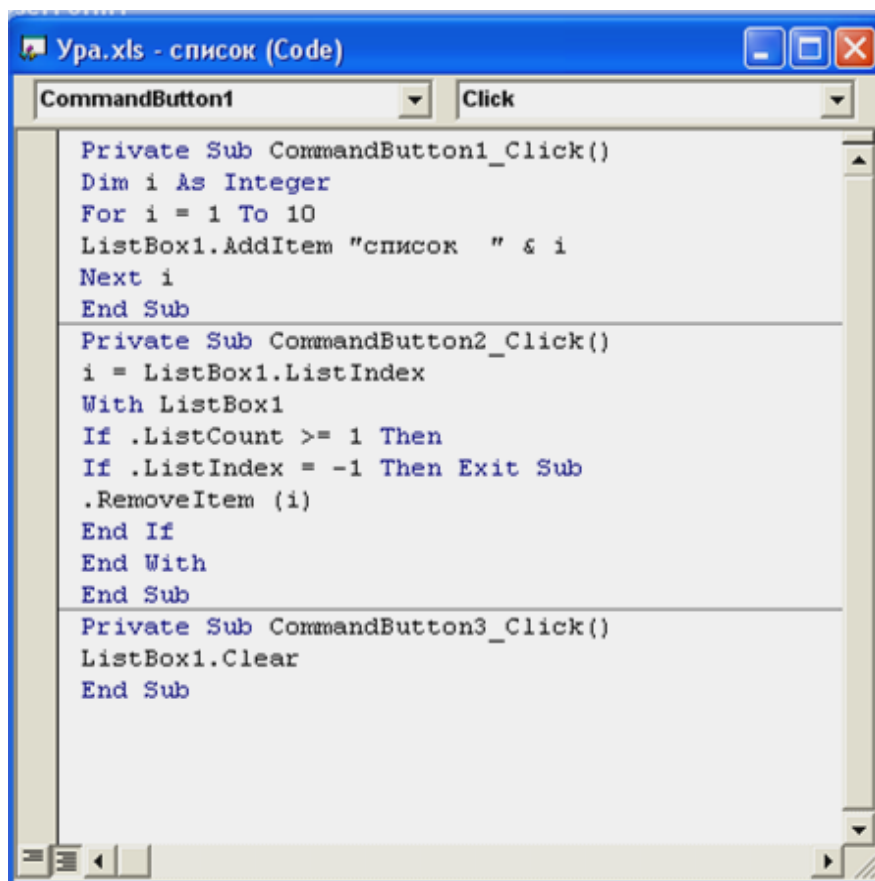


Рисунок 45 – Код программы

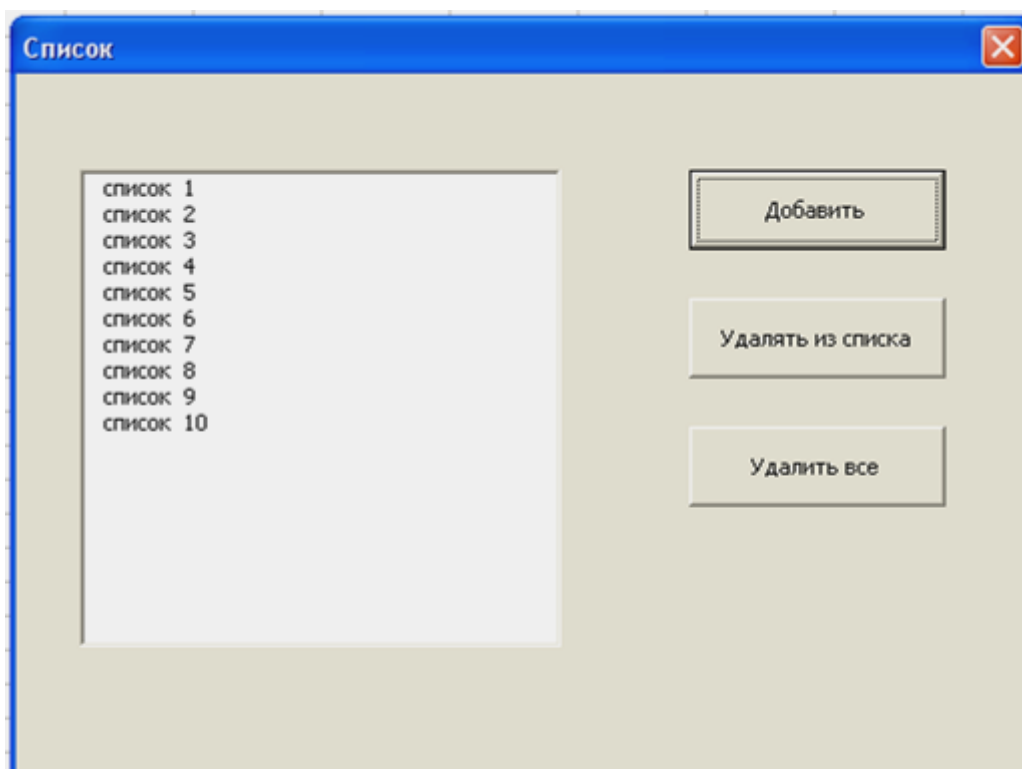


Рисунок 46 – Приложение Список

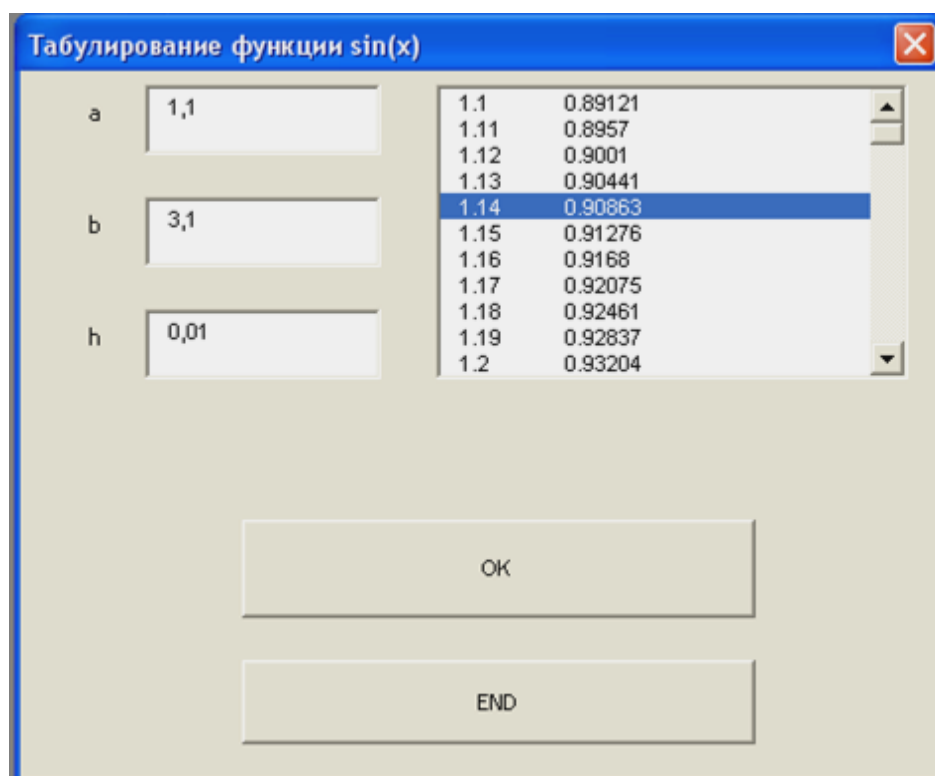


Рисунок 47 – Диалоговое окно Табулирование функции  $\sin(x)$

Технология выполнения задания:

1 При помощи окна Properties установить значения свойств Name и Caption элементов управления следующим образом, как показано в таблице 42.

Таблица 42 – Свойства элементов управления

Элемент управления	Свойство	Значение
Форма	Caption	Табулирование функции $\sin(x)$
Надпись	Caption	a
Поле ввода	Name	TextBox2
Надпись	Caption	b
Поле ввода	Name	TextBox3
Надпись	Caption	h
Поле ввода	Name	TextBox4
Кнопка	Name	OK
	Caption	CommandButton1
Список	Name	ListBox1

2 В модуле формы набрать следующий код, указанный на рисунке 48. Проект готов.

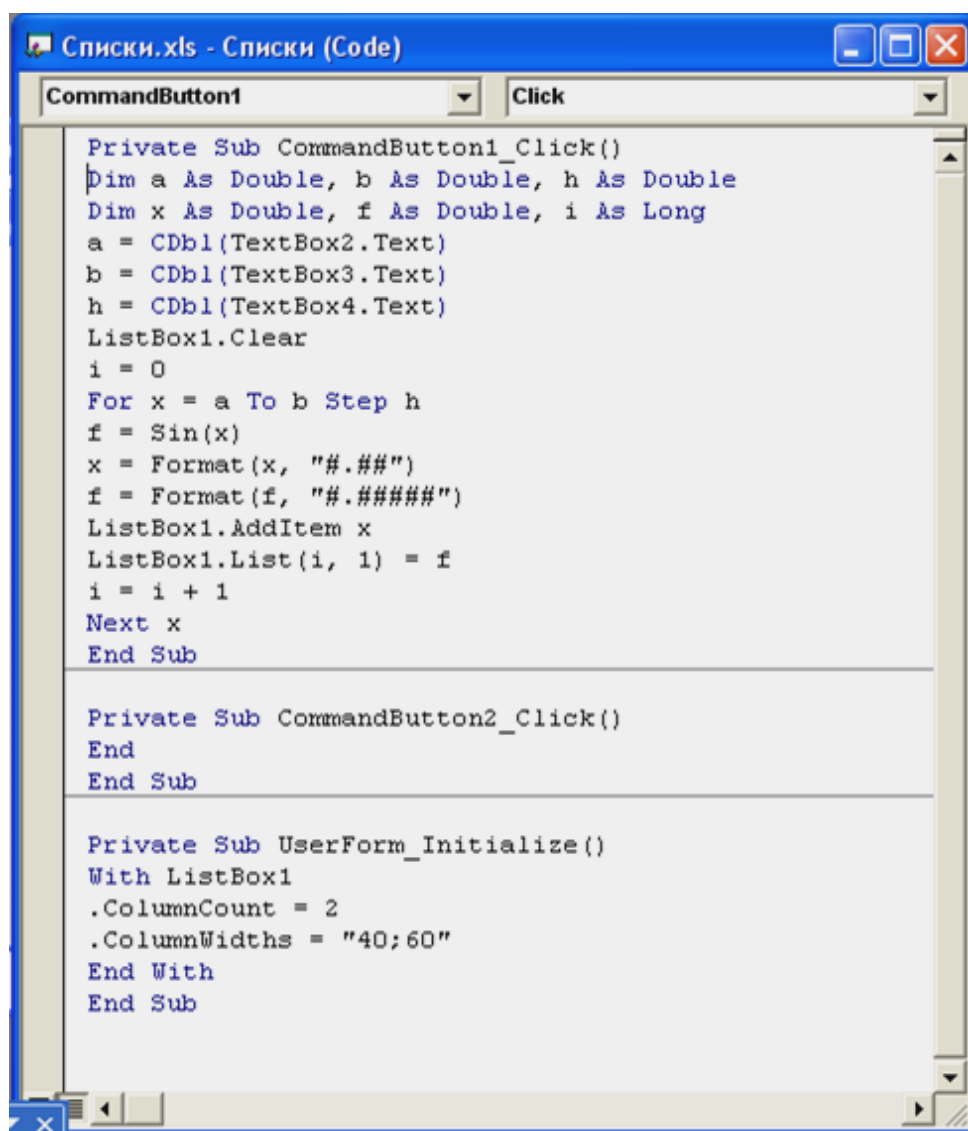


Рисунок 48 – Программный код

### 3.5.3 Многоэлементный выбор из списка

Свойство списка MultiSelect позволяет устанавливать режим, при котором допустим выбор нескольких элементов из списка.

Устанавливает способ выбора элементов списка. Возможные значения:

fmMultiSelectSingle или 0 (выбор только одного элемента);

fmMultiSelectMulti или 1 (разрешен выбор нескольких элементов, выбор осуществляется либо щелчком, либо нажатием клавиши <Пробел>);

fmMultiSelectExtended или 2 (разрешено использование клавиши <Shift> при выборе ряда последовательных элементов списка).

Свойство Selected предоставляет возможность проверить, выбран ли элемент с указанным индексом.

Пример 1.

Постановка задачи. Создать форму, позволяющую вычислить среднее значения выбранных в списке элементов (рисунок 49). Расположить список, кнопку и поле ввода.

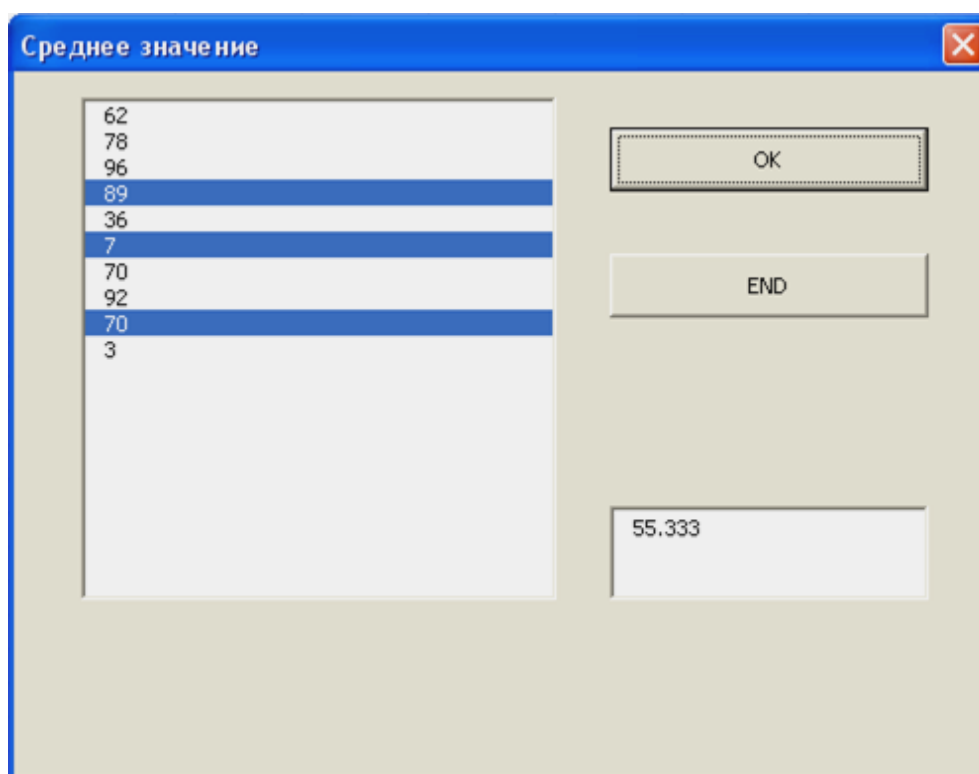


Рисунок 49 – Диалоговое окно «Среднее значение»

Технология выполнения задания:

1 При помощи окна Properties установить значения свойств Name и Caption элементов управления следующим образом, как показано в таблице 43.

Таблица 43 – Свойства элементов управления

Элемент управления	Свойство	Значение
Форма	Caption	Среднее значение
Кнопка	Name	CommandButton1
	Caption	OK
Кнопка	Name	CommandButton2
	Caption	END
Поле ввода	Name	TextBox1
Список	Name	ListBox1

2 В модуле формы набрать следующий код (рисунки 50, 51, 52). Проект готов.

Примечание. При инициализации формы список заполняется последовательностью случайных чисел из интервала от 0 до 100. Щелчком мыши с помощью клавиатуры <Ctrl> (выбор несмежных элементов списка) или <Shift> (выбор смежных элементов списка) выбрать группу элементов в списке. Нажать кнопку OK, и в поле ввода отобразится среднее арифметическое значение выбранных из списка чисел.

```

Private Sub UserForm_Initialize()
Dim i As Integer
ListBox1.MultiSelect = fmMultiSelectExtended
Randomize
For i = 1 To 10
ListBox1.AddItem Int(101 * Rnd())
Next i
End Sub

```

Рисунок 50 – Программный код формирования списка

Среднее значение равно сумме значений, деленной на их количество. Если в списке не выбран ни один элемент, то данная формула приведет к генерации ошибки деления на нуль. Поэтому, во избежание этой ошибки в коде сначала определяется количество выбранных элементов, и если оно положительно, то лишь тогда находится среднее значение.

```

Private Sub CommandButton1_Click()
Dim s As Double, i As Integer, k As Integer
k = 0
For i = 1 To ListBox1.ListCount - 1
If ListBox1.Selected(i) Then
k = k + 1
End If
Next i
If k = 0 Then Exit Sub
With ListBox1
s = 0
For i = 1 To ListCount - 1
If .Selected(i) Then
s = s + .List(i)
End If
Next i
s = s / k
End With
TextBox1 = CDb1(Format(s, "#.###"))
End Sub

```

Рисунок 51 – Программный код, с помощью которого определяется количество выбранных элементов и их среднее значение

Элемент управления ComboBox (Поле со списком). Применяется для хранения списка значений. Он сочетает в себе функциональные возможности списка и поля ввода. В отличие от списка в Поле со списком отображается только один элемент. У него отсутствует режим выделения нескольких элементов списка. Поле со списком позволяет пользователю вводить значение через поле ввода, как это делает элемент управления Поле.

```
Private Sub CommandButton2_Click()
End
End Sub
```

Рисунок 52 – Программный код командной кнопки END

Свойства элемента управления Поле со списком, такие как ListIndex, ListCount, List и методы Clear, RemoveItem и AddItem, аналогичны соответствующим свойствам и методам элемента управления Список.

Кроме того, у него есть ряд уникальных свойств, указанных в таблице 44.

Таблица 44 - Свойства объекта «Поле со списком»

Свойство	Описание
DropButtonStyle	Устанавливает вид поля со списком. Допустимые значения: - fmDropButtonStylePlain или 0 (кнопка без символов) - fmDropButtonStyleArrow или 1 (кнопка со стрелкой) - fmDropButtonStyleEllipsis или 2 (кнопка с эллипсом) - fmDropButtonStyleReduce или 3 (кнопка с линией)
ListRows	Устанавливает количество элементов, отображаемых в поле со списком
MathRequired	Допустимые значения: True (в поле ввода нельзя ввести значения, отличные от перечисленных в списке, т.е. в Поле со списком отключается функция поля ввода) и False (в противном случае)
MathFound	Допустимые значения: True (среди элементов поля со списком имеется элемент, который совпадает с элементом, вводимым в поле ввода) и False (в противном случае)

### 3.5.4 Добавление и удаление данных

#### Пример 1.

Постановка задачи. Создать приложение, в котором через поле ввода поля со списком в список возможно добавление новых элементов. Кроме того, в проекте будет предусмотрено удаление выбранного элемента из списка, а также очистка всего списка (рисунок 53).

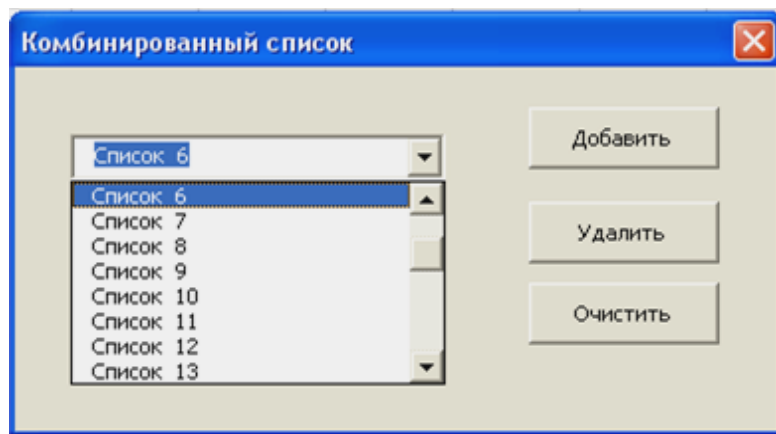


Рисунок 53 – Приложение Поле со списком

Технология выполнения задания:

- 1 Создать форму, разместив на ней поле со списком и три командные кнопки.
- 2 В модуле формы набрать программный код, показанный на рисунке 54.

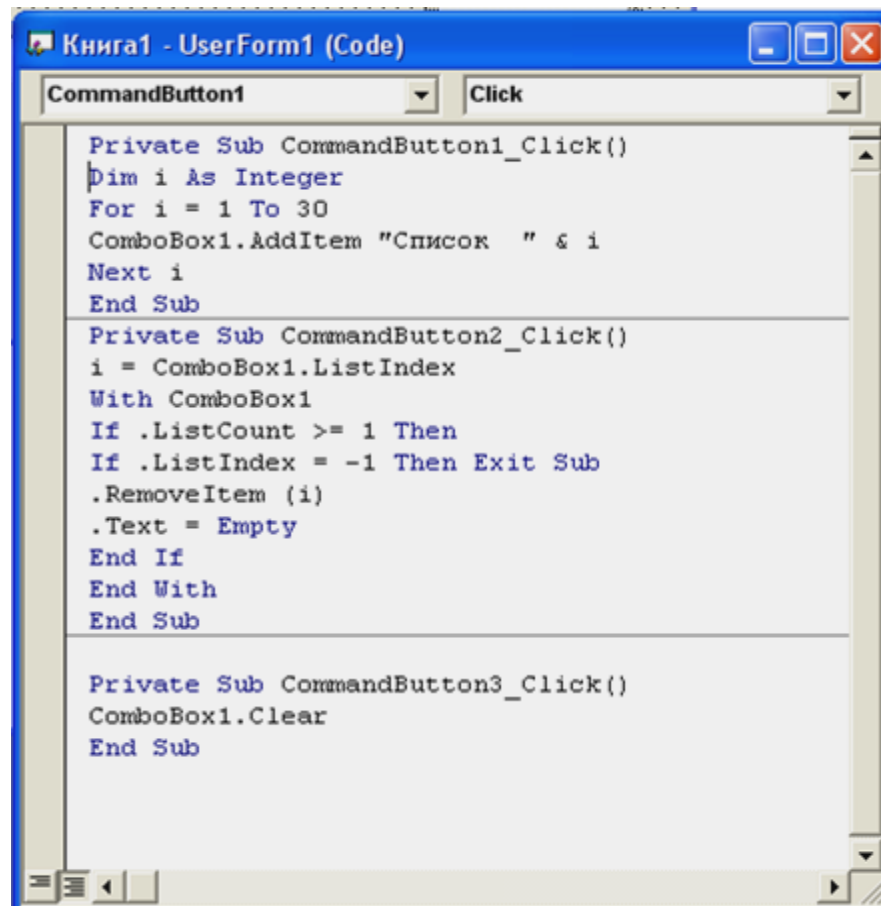


Рисунок 54 – Программный код

Примечание. Метод RemoveItem удаляет элемент из списка, но не очищает поле ввода. Поэтому в коде предусмотрена после удаления элемента очистка поля ввода.

Разработка пользовательского приложения с помощью объектов: список и поле со списком

Цель: приобретение навыков разработки пользовательского приложения с помощью объектов формы - список и поле со списком. Индивидуальные варианты лабораторной работы № 12 выполняются в соответствии с индивидуальными вариантами лабораторной работы № 5 (2 часа).

### 3.6 Объект формы в VBA: Рисунок

Элемент управления Image (Рисунок). Предназначен для вывода растровых изображений, имеющих один из следующих форматов: \*.bmp, \*.cur, \*.gif, \*.ico, \*.jpg, \*.wmf. Основные свойства элемента управления Image представлены в таблице 45.

Таблица 45 - Основные свойства данного элемента управления

Свойство	Описание
AutoSize	Принимает логические значения и устанавливает, должен ли объект автоматически изменять размер, чтобы разместить изображение целиком
Picture	Задаёт отображаемый графический файл. Используется с функцией LoadPicture Синтаксис: Picture = LoadPicture (FileName) где FileName – полное имя отображаемого графического файла
PictureSizeMode	Устанавливает масштабирование изображения. Допустимые значения: - fmPictureSizeModeClip или 0 (непомещающиеся в границах объекта части рисунка обрезаются); - fmPictureSizeModeStretch или 1 (рисунок масштабируется так, чтобы он занимал всю поверхность объекта); - fmPictureSizeModeZoom или 2 (рисунок масштабируется с сохранением относительных размеров так, чтобы он помещался целиком внутри объекта)
PictureAlignment	Устанавливает расположение рисунка внутри объекта. Допустимые значения: - fmPictureAlignmentTopLeft или 0 (в верхнем левом углу); - fmPictureAlignmentTopRight или 1 (в верхнем правом углу); - fmPictureAlignmentCenter или 2 (в центре); - fmPictureAlignmentBottomLeft или 3 (в нижнем левом углу); - fmPictureAlignmentBottomRight или 4 (в нижнем правом углу)
PictureTiling	Принимает логические значения и устанавливает, следует ли заполнять объект мозаикой из изображения



### 3.6.1 Добавление растровых изображений в форму

Добавить растровое изображение можно непосредственно в форму или в элементы управления Image. Для добавления изображения в форму нужно выполнить следующие действия:

- 1) создать новый проект и присвоить ему имя. Для заголовка формы задать название, например «Растровая графика» (свойство Caption);
- 2) добавить в форму элемент управления Image;
- 3) установить размеры и положение элементов управления Image, как показано на рисунке 55;

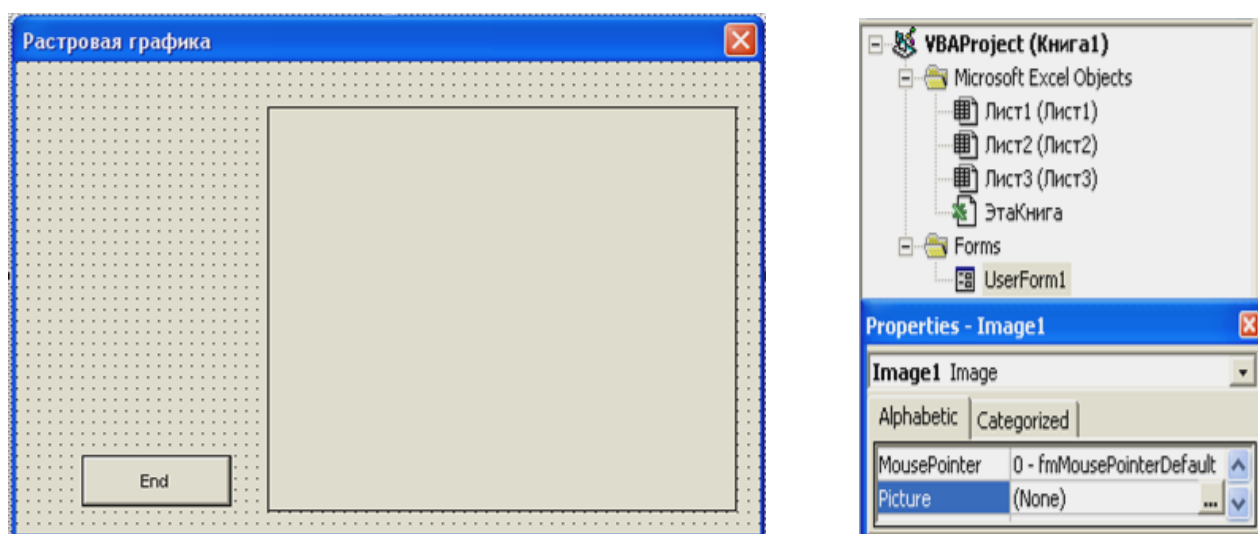


Рисунок 55 – Проект формы «Растровая графика»

- 4) выделить форму. Обратиться к свойству Picture в окне “Properties” (Свойства). Открыть диалоговое окно “LoadPicture” (Загрузка рисунка), показанное на рисунке 56

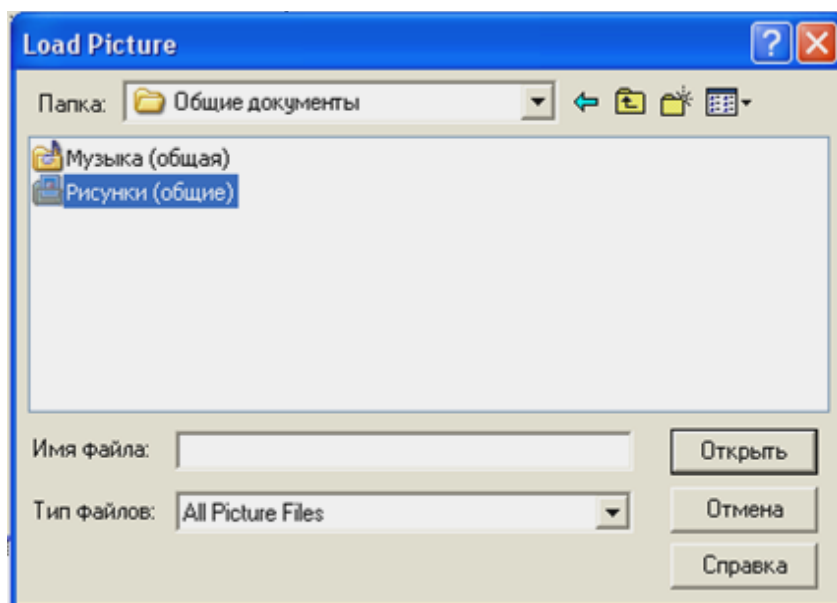


Рисунок 56 – Диалоговое окно «Загрузка рисунка»

5) в диалоговом окне “LoadPicture” перейти в каталог Windows (или в другой каталог) и отыскать файл с расширением \*.bmp, а затем щелкнуть мышью по кнопке <Открыть>. На месте вставки рисунка в форме появится изображение;

6) для свойства AutoSize объекта Image1 установить значение True. Размеры окна Image1 будут приведены к размерам рисунка;

7) сохранить проект и запустить программу. Внешний вид формы после запуска программы будет выглядеть подобно изображению на рисунке 57.

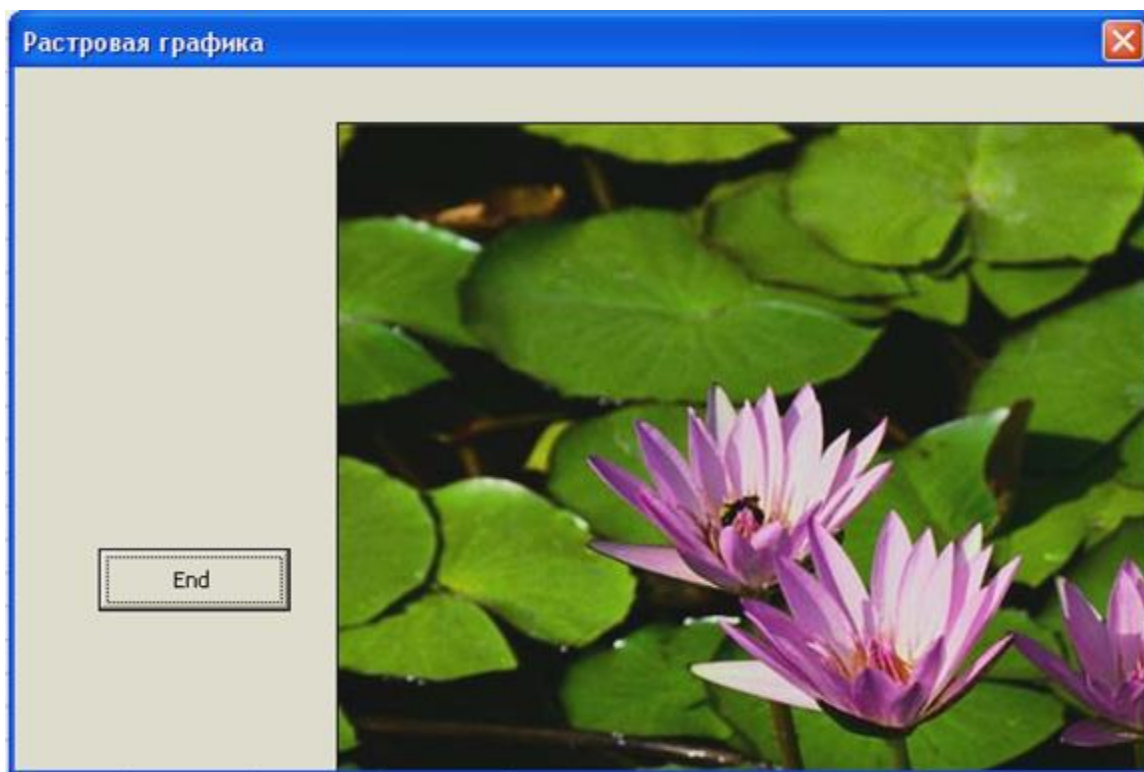


Рисунок 57 – Внешний вид формы

### 3.6.2 Вставка изображений в форму

Ранее рассматривался пример добавления растровых графических изображений в форму во время проектирования. Функция LoadPicture позволяет делать добавление изображений в форму во время выполнения программы.

Синтаксис:

NamePicture = LoadPicture (FilePath)

где

NamePicture – имя элемента управления типа Image;

LoadPicture – имя функции;

FilePath – строковая переменная или константа, которая содержит путь к файлу с графическим изображением.

Пример 1.

Постановка задачи. Создать приложение, в котором, щелкая кнопкой мыши на элементах управления OptionButton (Переключатели), можно менять графические изображения в окне Image (рисунок 58).

Технология выполнения задания:

1 Конструирование проекта.

Процесс программирования приложения Рисунок предусмотрен тремя этапами, как показано в таблице 46.

Таблица 46 - Этапы программирования

Этапы программирования	Число объектов
1 Создание пользовательского интерфейса	5 объектов
2 Установка свойств объектов	5 объектов
3 Составление программного кода	4 объекта

При помощи окна Properties установить значения свойств Name и Caption полей элементов управления следующим образом, как показано в таблице 47.

Таблица 47 – Свойства элементов управления

Элемент управления	Свойство	Значение	Примечание
Переключатель	Name Caption	OptionButton1	Рисунок 1
Переключатель	Name Caption	OptionButton2	Рисунок 2
Переключатель	Name Caption	OptionButton3	Рисунок 3
Командная кнопка	Name Caption	CommandButton1	End
Рисунок	Name	Image1	Рисунок

2 В модуле формы набрать следующий код, изображенный на рисунке 59.

### Лабораторная работа № 13

Разработка пользовательского приложения с помощью объекта: Рисунок

Цель: приобретение навыков разработки пользовательского приложения с помощью объекта формы VBA – рисунок. Лабораторная работа № 13 выполняется согласно вышеописанной технологии. Рисунок выбирается студентом самостоятельно (2 часа).

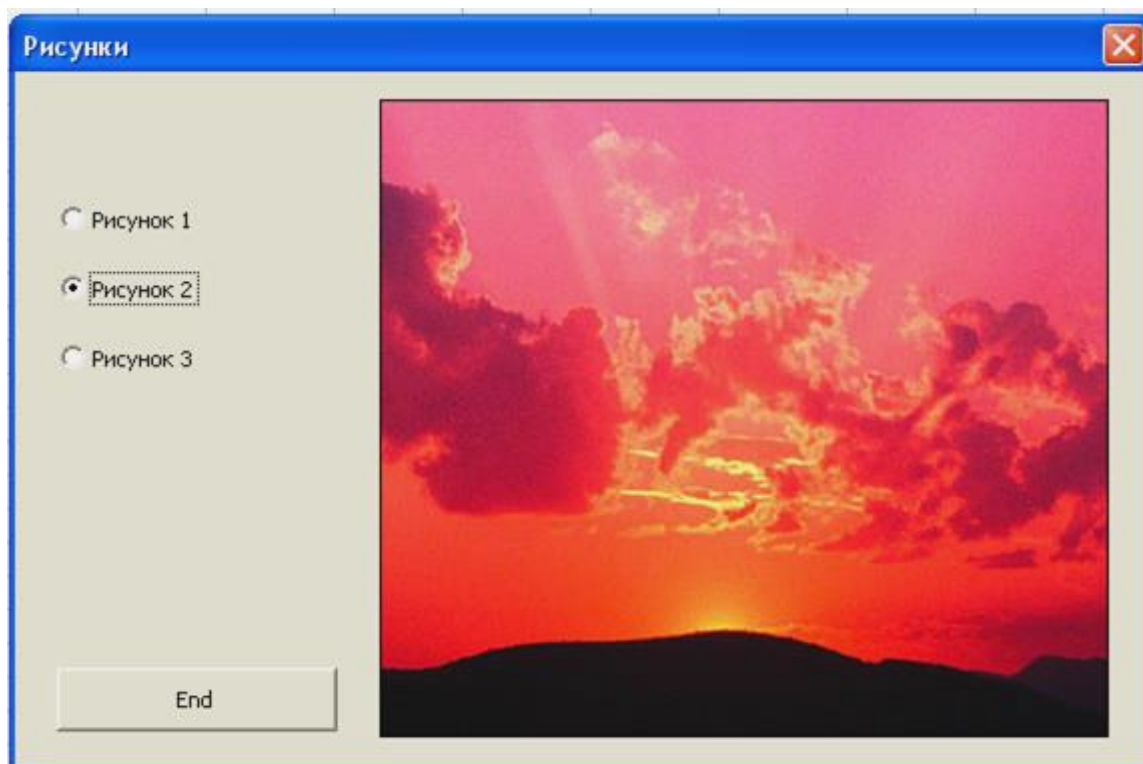


Рисунок 58 – Окно приложения Рисунок

```
Private Sub OptionButton1_Click()
Dim Path As String
Path = "C:\Documents and Settings\All Users\Документы\Мои рисунки\Образцы рисунков\" & "Голубые холмы.jpg"
Image1.Picture = LoadPicture(Path)
Image1.PictureSizeMode = fmPictureSizeModeStretch
End Sub

Private Sub OptionButton2_Click()
Dim Path As String
Path = "C:\Documents and Settings\All Users\Документы\Мои рисунки\Образцы рисунков\" & "Закат.jpg"
Image1.Picture = LoadPicture(Path)
Image1.PictureSizeMode = fmPictureSizeModeStretch
End Sub

Private Sub OptionButton3_Click()
Dim Path As String
Path = "C:\Documents and Settings\All Users\Документы\Мои рисунки\Образцы рисунков\" & "Зима.jpg"
Image1.Picture = LoadPicture(Path)
Image1.PictureSizeMode = fmPictureSizeModeStretch
End Sub

Private Sub CommandButton1_Click()
End
End Sub
```

Рисунок 59 – Программный код

### 3.7 Интеграция приложений: MS Excel и MS Word

#### 3.7.1 Открытие документа MS Word функцией CreateObject

1 Для реализации данного демонстрационного приложения создать форму, в которой расположить командную кнопку, установить значение ее свойства Name, а в модуле формы набрать код программы, изображенный на рисунке 60.

2 Создать документ MS Word с названием Пример.doc и расположить в рабочем каталоге проекта.

3 Перед выполнением программы надо установить ссылку на библиотеку объектов MS Word 9.0 Object Library в окне References – VBAProject, отображаемом на экране выбором команды Tools/References в редакторе VB.

4 С помощью программного кода первоначально проверяется наличие файла Пример.doc в рабочем каталоге приложения. Если файл присутствует, то с помощью функции CreateObject запускается еще один экземпляр MS Word, даже если MS Word уже был запущен, и в нем открывается файл.

5 После открытия файла на экране отображается диалоговое окно с сообщением «Закреть документ?». Нажатие кнопки ОК на этом окне приведет как к закрытию окна, так и созданного экземпляра Word с загруженным файлом. Нажатие кнопки Нет оставит документ открытым.

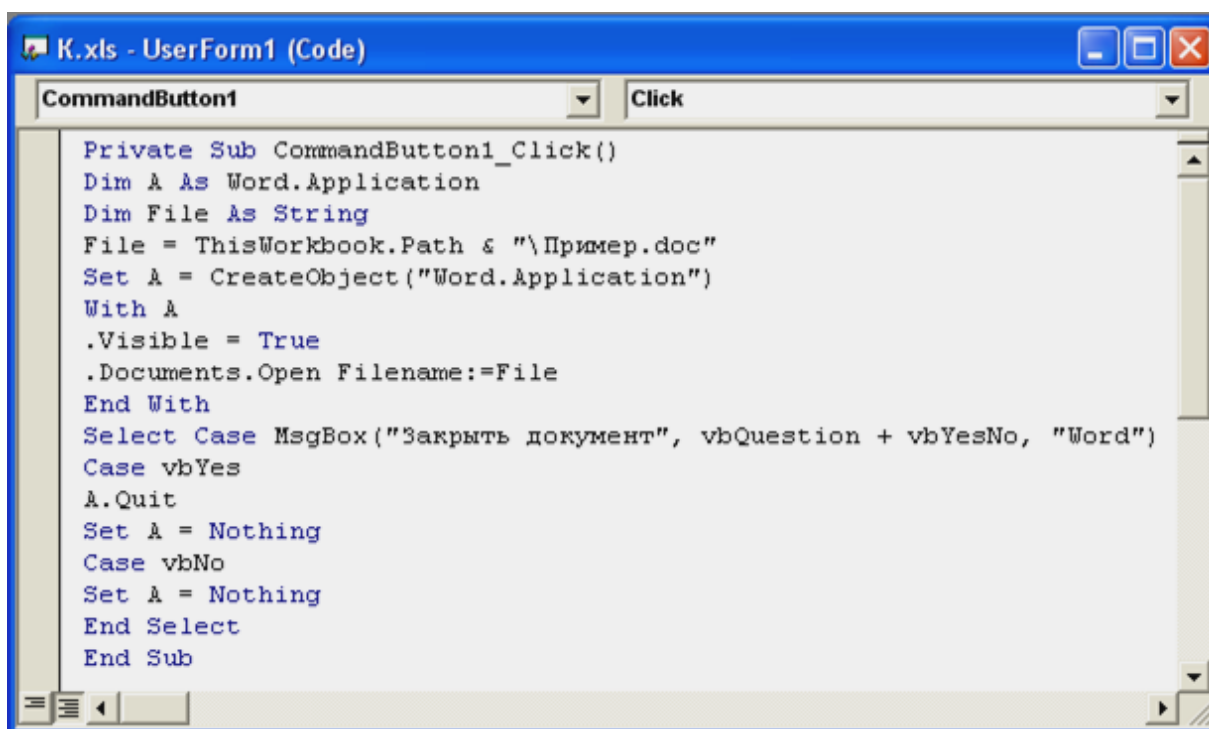


Рисунок 60 – Программный код

### 3.7.2 Открытие документа MS Word функцией GetObject

Использование функции GetObject, позволяющей открывать существующий документ MS Word, подобно использованию функции CreateObject (рисунок 61).

Функцию GetObject следует использовать для доступа к существующим документам, хранящимся в файлах.

Функцию GetObject можно применить для доступа к объекту Application, предварительно запущенного приложения MS Office MS Word. Для этого надо вызвать функцию GetObject без первого параметра.

Этот способ доступа к объекту Application любого уже запущенного приложения MS Office применяется, когда нет необходимости в запуске еще одного экземпляра приложения, что происходит при работе функции CreateObject.

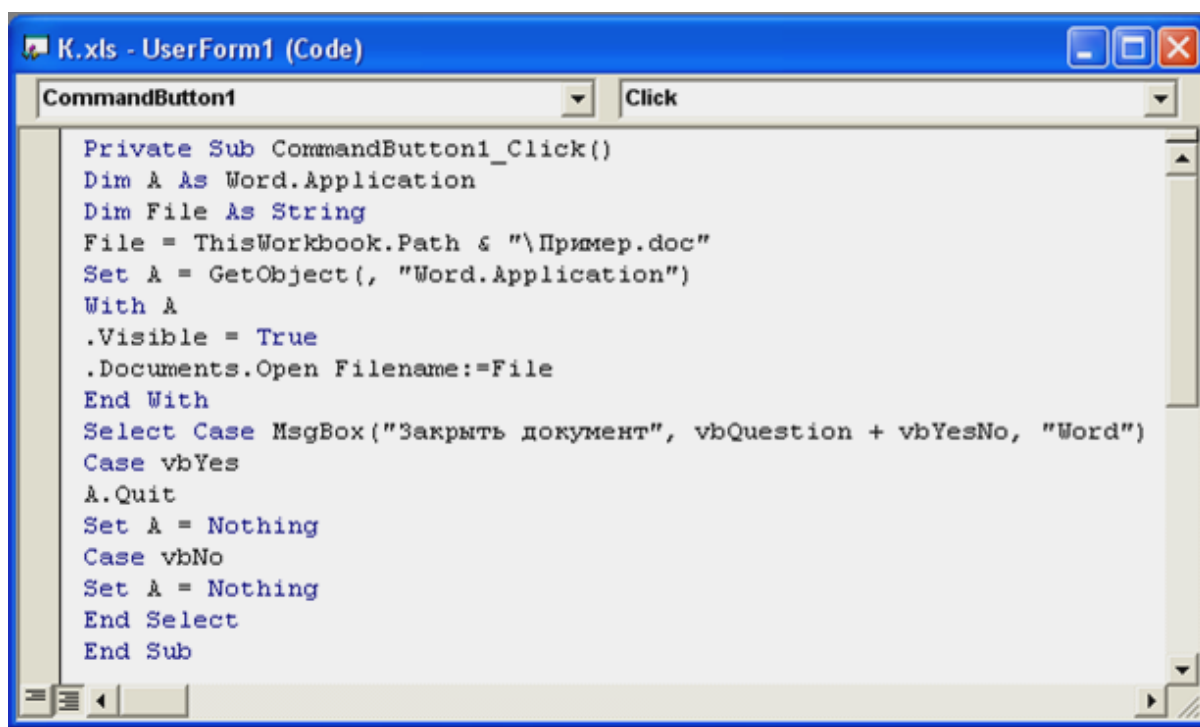


Рисунок 61 – Программный код

#### Пример 1.

Постановка задачи. Создать квитанцию об оплате услуг за использование Internet, которая автоматизировано будет заполняться с помощью приложения MS Excel.


Технология выполнения задания:

1 Создать новый документ MS Word с названием «Квитанция.doc», как показано на рисунке 62, и сохранить в рабочем каталоге будущего проекта.

2 В нем расположить три поля ввода для ввода UID, фамилии пользователя, суммы предоплаты по следующему алгоритму:

- расположить курсор в ячейке таблицы рядом со словом UID пользователя;
- в управляющем меню выбрать команду Вид→Панели инструментов→Элементы управления;



- активизировать на панели инструментов «Элементы управления» кнопку Поле ввода ;
- установить соответствующий размер этого поля.

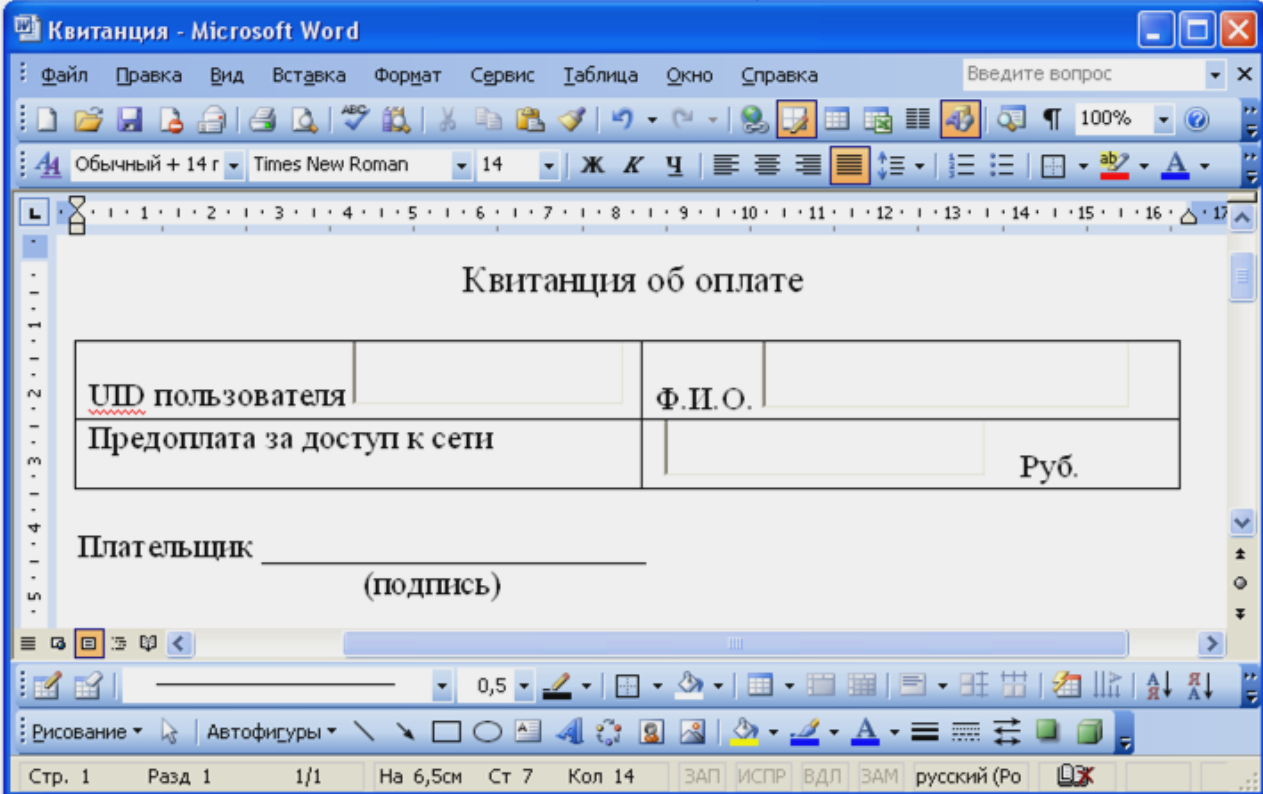


Рисунок 62 – Квитанция об оплате услуг в Internet

Аналогично проделать с двумя остальными полями ввода.

3 Конструирование проекта. Создать форму, как показано на рисунке 63. Расположить в форме три надписи, три поля ввода и две командные кнопки.

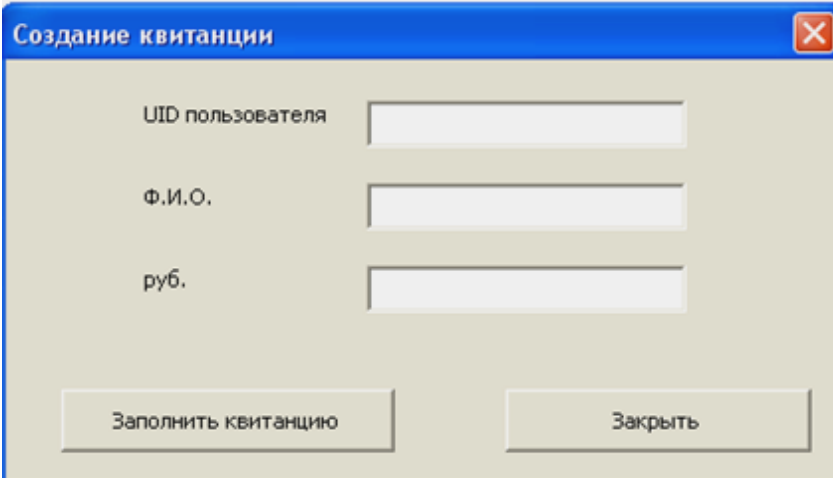


Рисунок 63– Окно приложения «Создание квитанции»

4 Установить при помощи окна Properties следующие значения свойств, как показано в таблице 48.

Таблица 48 – Свойства элементов управления

Элемент управления	Свойство	Значение
Форма	Caption	Создание квитанции
Надпись	Caption	UID
Надпись	Caption	Ф.И.О.
Надпись	Caption	р.
Поле ввода	Name	TextBox1
Поле ввода	Name	TextBox2
Поле ввода	Name	TextBox3
Кнопка	Name	CommandButton1
	Caption	Заполнить квитанцию
Кнопка	Name	CommandButton1
	Caption	Заккрыть

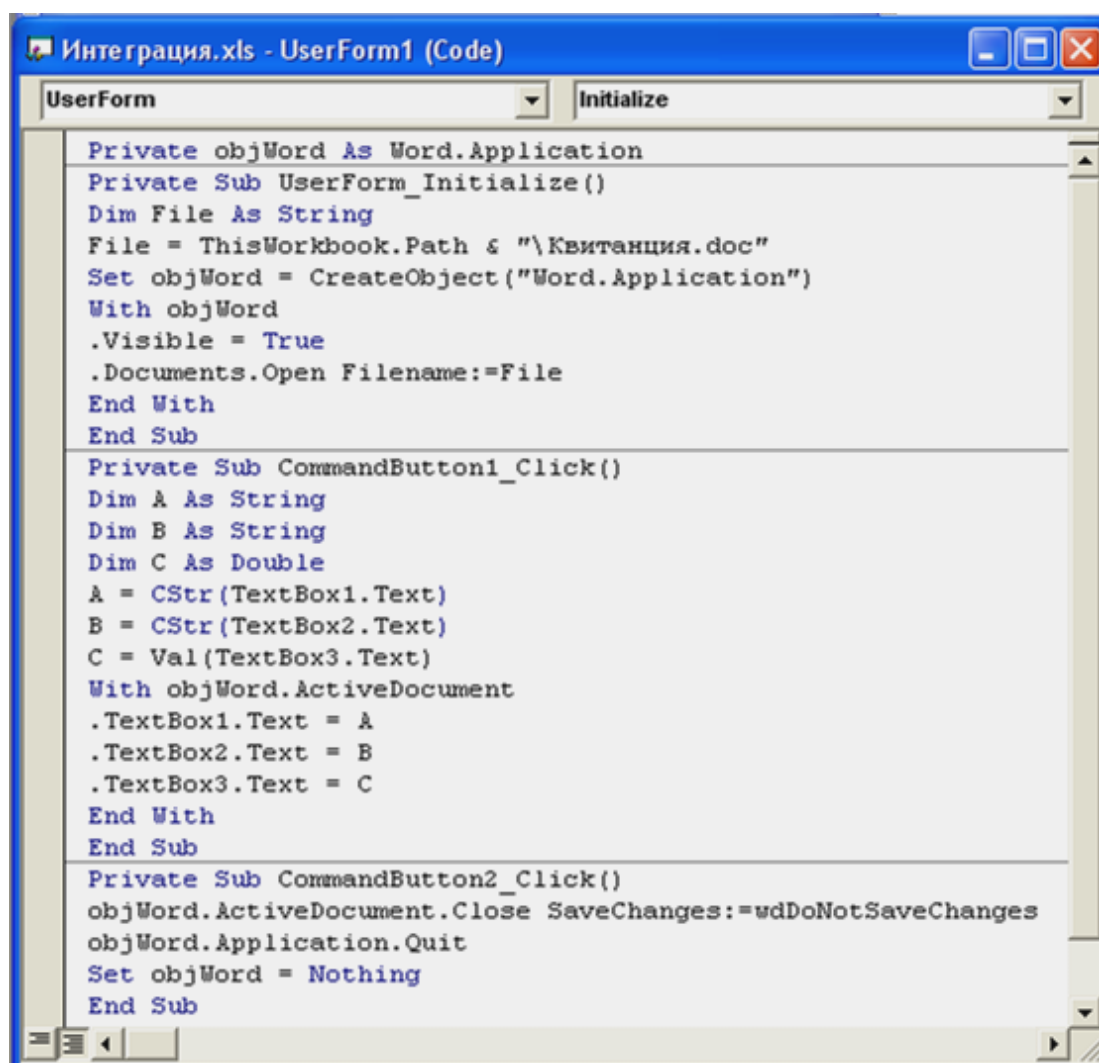


Рисунок 64 – Программный код к приложению «Создание квитанции»



- 5 Установить ссылку на библиотеку объектов MS Word 9.0 Object Library.
- 6 В модуле формы набрать следующий код, представленный на рисунке 64.
- 7 Запустить проект.

Примечание. На экране появится документ MS Word Квитанция.doc. Его необходимо свернуть на панель задач и в окне приложения «Создание квитанции» заполнить соответствующие поля, выполнить команду «Заполнить квитанцию». Затем осуществить переход в документ MS Word Квитанция.doc и просмотреть результат выполненных действий. В приложении выполнить команду «Заккрыть».

#### Лабораторная работа № 14

Цель: приобретение навыков осуществлять интеграцию популярных приложений, входящих в пакет MS Office операционной системы Windows'XP, MS Excel и MS Word. Индивидуальные варианты лабораторной работы № 14 представлены в таблице 49 (2 часа).

Таблица 49

№ В	Варианты индивидуальных заданий																				
Создать документ MS Word. Заполнить его автоматизировано через MS Excel																					
1	<div>ОренбургРегионГаз</div> <div>СЧЕТ-КВИТАНЦИЯ за май 2006 г.</div> <table><tr><td></td><td>кол-во</td><td>ед.изм.</td><td>тариф</td><td>сумма</td></tr><tr><td>Плита</td><td></td><td></td><td></td><td></td></tr><tr><td>Льготы</td><td></td><td></td><td></td><td></td></tr><tr><td>Итого к оплате:</td><td colspan="4"></td></tr></table> <div>Поля «Сумма» и «Итого к оплате» выполнить вычисляемыми в зависимости от вводимых данных.</div>		кол-во	ед.изм.	тариф	сумма	Плита					Льготы					Итого к оплате:				
	кол-во	ед.изм.	тариф	сумма																	
Плита																					
Льготы																					
Итого к оплате:																					
2	<div>Акционерное общество «Телесот»</div> <div>КВИТАНЦИЯ ОБ АБОНЕНТСКОЙ ОПЛАТЕ ЗА УСЛУГИ СВЯЗИ</div> <table><tr><td>Телефон №</td><td></td></tr><tr><td>Оплата за</td><td></td></tr><tr><td>Тариф</td><td></td></tr><tr><td>Кол-во времени</td><td></td></tr><tr><td>Сумма, р</td><td></td></tr></table> <div>Поле «Сумма» сделать вычисляемым в зависимости от вводимых данных.</div>	Телефон №		Оплата за		Тариф		Кол-во времени		Сумма, р											
Телефон №																					
Оплата за																					
Тариф																					
Кол-во времени																					
Сумма, р																					
3	<div>КВИТАНЦИЯ ЗА ЖИЛИЩНЫЕ И КОММУНАЛЬНЫЕ УСЛУГИ</div> <table><tr><td>Получатель платежа</td><td></td></tr><tr><td>Учреждение банка</td><td></td></tr><tr><td>Адрес:</td><td></td></tr><tr><td>Ф.И.О.</td><td></td></tr><tr><td>Месяц</td><td></td></tr><tr><td>Сумма</td><td></td></tr><tr><td>Пеня</td><td></td></tr><tr><td>Всего:</td><td></td></tr></table> <div>Поле «Всего» сделать вычисляемым в зависимости от вводимых данных.</div>	Получатель платежа		Учреждение банка		Адрес:		Ф.И.О.		Месяц		Сумма		Пеня		Всего:					
Получатель платежа																					
Учреждение банка																					
Адрес:																					
Ф.И.О.																					
Месяц																					
Сумма																					
Пеня																					
Всего:																					

Продолжение таблицы 49

4	<b>ГАРАНТИЙНЫЙ ТАЛОН НА СОТОВЫЙ ТЕЛЕФОН</b>														
	Изделие														
	Модель														
	Серийный номер														
	Дата продажи														
	Бесплатное сервисное обслуживание														
	Адрес фирмы-продавца														
	Телефон фирмы-продавца														
Подпись покупателя _____															
5	<b>НАЛОГОВОЕ УВЕДОМЛЕНИЕ № _____</b> на уплату налога на строения за _____ год Ф.И.О. налогоплательщика _____ ИНН _____														
	<table border="1" style="width: 100%;"> <tr> <th data-bbox="285 822 608 902">Объект налогообложения</th> <th data-bbox="608 822 842 902">Стоимость имущества, р.</th> <th data-bbox="842 822 1010 902">Ставка налога</th> <th data-bbox="1010 822 1177 902">Сумма налога к уплате</th> <th data-bbox="1177 822 1469 902">Место нахождения объекта</th> </tr> <tr> <td data-bbox="285 902 608 992"> </td> <td data-bbox="608 902 842 992"> </td> <td data-bbox="842 902 1010 992"> </td> <td data-bbox="1010 902 1177 992"> </td> <td data-bbox="1177 902 1469 992"> </td> </tr> </table>	Объект налогообложения	Стоимость имущества, р.	Ставка налога	Сумма налога к уплате	Место нахождения объекта									
	Объект налогообложения	Стоимость имущества, р.	Ставка налога	Сумма налога к уплате	Место нахождения объекта										
Руководитель инспекции Министерства РФ по налогам и сборам Подпись _____ Поле «Сумма налога к уплате» сделать вычисляемым в зависимости от вводимых данных.															
6	<b>СВИДЕТЕЛЬСТВО</b> о постановке на учет в налоговом органе физического лица по месту жительства на территории РФ Настоящее Свидетельство выдано в соответствии с положениями ч.1 Налогового кодекса РФ, принятого Федеральным законом от 31 июля 1998 г. № 146-ФЗ Физическому лицу _____ Пол _____ Дата рождения _____ Место рождения _____ и подтверждает постановку физического лица на учет (число, месяц, год постановки на учет _____ в Инспекции МНС России (наименование налогового органа) _____ ИНН _____ Дата выдачи Свидетельства _____														

Продолжение таблицы 49

7	<p><b>ДОГОВОР ЛИЧНОГО БАНКОВСКОГО СЧЕТА</b></p> <p>Банк «Центральное ОВК», именуемый в дальнейшем «Банк», в лице Директора Оренбургского регионального филиала Банка, _____, действующего на основе Устава, Положения о филиале, Доверенности № _____ от _____, с одной стороны, и</p> <p>Ф.И.О. _____</p> <p>Гражданство _____</p> <p>Адрес _____</p> <p>Дата рождения _____</p> <p>Паспортные данные: серия _____, номер _____, выдан _____</p> <p>Клиенту открыт счет:</p> <p>Номер личного банковского счета _____</p> <p>Дата открытия счета _____</p>		
8	<p><b>РАСХОДНЫЙ КАССОВЫЙ ОРДЕР № _____</b></p> <p>Число _____ Месяц _____ Год _____</p> <p>Выдать (Ф.И.О.) _____</p> <p>Основание: _____</p> <p>_____ р. _____ к.</p> <p>(прописью)</p> <p>Руководитель _____</p> <p>Главный бухгалтер _____</p> <p>Получил _____ р. _____ к.</p> <p>(прописью)</p> <p>« _____ » _____ 20 ____ г.      Подпись _____</p> <p>Выдал кассир _____</p>		
9	<p><b>ОБЪЯВЛЕНИЕ № _____</b></p> <p>на взнос наличными</p> <p>« _____ » _____ 20 ____ г.</p> <p>для зачисления на счет № _____</p> <table border="1" data-bbox="933 1400 1348 1568"> <tr> <td>_____</td> </tr> <tr> <td>сумма</td> </tr> </table> <p>От кого _____</p> <p>Банк получателя _____</p> <p>Получатель _____</p> <p>_____ р. _____ к.</p> <p>(сумма прописью)</p> <p>Источник вноса _____</p> <p>Подпись вносителя: _____ Бухгалтер _____</p> <p>Деньги принял кассир _____</p>	_____	сумма
_____			
сумма			

Продолжение таблицы 49

10	<p>СЧЕТ-ФАКТУРА № _____ от _____</p> <p>Продавец _____ Покупатель _____</p> <p>Адрес _____ Адрес _____</p> <p>ИНН/КПП продавца _____</p> <p>Грузоотправитель и его адрес _____</p> <p>Грузополучатель и его адрес _____</p>							
	Наименование товара	Ед. изм.	Кол-во	Цена	Стоимость	Сумма налога	Всего	Страна происхождения
	<p>Руководитель организации _____</p> <p>Главный бухгалтер _____</p> <p>Индивидуальный предприниматель _____</p> <p>Сделать поля «Стоимость» и «Всего» вычисляемыми в зависимости от вводимых данных.</p>							
11	<p>Наименование предприятия, организации _____</p> <p>Адрес _____</p> <p>ИНН _____</p> <p>Получатель _____</p> <p>Банк получателя _____</p> <p>СЧЕТ № _____ от «__» _____ 20__ г.</p> <p>Заказчик _____</p> <p>Платательщик _____</p>							
	№	Наименование товара	Ед. изм.	Кол-во	Цена	Сумма		
	<p>Руководитель предприятия _____</p> <p>Главный бухгалтер _____</p> <p>Сделать поле «Сумма» вычисляемым в зависимости от вводимых данных.</p>							
12	<p>РАСХОДНАЯ НАКЛАДНАЯ № _____ от _____</p> <p>Покупатель: _____</p>							
	Наименование товара		Цена	Кол-во	Ед.	Сумма		
	Итого:							
	<p>Отпустил _____ Получил _____</p> <p>Сделать поля «Сумма» и «Итого» вычисляемыми в зависимости от вводимых данных.</p>							

Продолжение таблицы 49

13	<p>Договор № _____ от _____</p> <p><b>О ВКЛАДЕ «ПЕНСИОННЫЙ ДЕПОЗИТ СБЕРБАНКА РОССИИ»</b></p> <p>Вкладчик вносит во вклад наличными деньгами, а Банк принимает денежные средства в сумме _____ р.</p> <p>Срок вклада: _____</p> <p>Дата окончания срока вклада: _____</p> <p>Дата возврата вклада: _____</p> <p>Процентная ставка составляет: _____ % годовых</p> <p>Выплата дохода _____ р.</p>
14	<p><b>КВИТАНЦИЯ</b></p> <p>к приходному кассовому ордеру № _____</p> <p>Принято от _____</p> <p>Основание: _____</p> <p>_____</p> <p>(прописью)</p> <p>_____ р. _____ к.</p> <p>« ____ » _____ 20 ____ г.</p> <p>Главный бухгалтер _____</p> <p>Кассир _____</p>
15	<p><b>ПРИХОДНЫЙ КАССОВЫЙ ОРДЕР № _____</b></p> <p>Число ____ Месяц ____ 20 ____ г.</p> <p>Корреспондирующий счет, субсчет _____</p> <p>Сумма _____</p> <p>Принято от _____</p> <p>Основание: _____</p> <p>_____ р. _____ к.</p> <p>(прописью)</p> <p>Главный бухгалтер _____</p> <p>Получил кассир _____</p>
16	<p><b>КВИТАНЦИЯ № _____</b></p> <p>За парковку автомобилей</p> <p>Платная автостоянка № _____</p> <p>Адрес _____</p> <p>Принят на хранение автотранспорт марки _____</p> <p>номерной знак _____ Владелец _____</p> <p>(фамилия, имя, отчество)</p> <p>Адрес _____ Телефон _____</p> <p>Дата приема _____ 20 ____ г.</p> <p>Сдал владелец _____</p> <p>(подпись)</p> <p>Принял кладовщик _____</p> <p>(подпись)</p> <p>Сумма _____</p> <p>(прописью)</p>

17	<p>ДОВЕРЕННОСТЬ № _____</p> <p>Дата выдачи «___» _____ 20__ г.</p> <p>Доверенность действительна по «___» _____ 20__ г.</p> <p>_____</p> <p style="text-align: center;">(наименование потребителя и его адрес)</p> <p>Счет № _____ в банке _____</p> <p>Доверенность выдана _____</p> <p style="text-align: center;">(фамилия, имя, отчество)</p> <p>Паспорт: серия _____ № _____ Кем выдан _____</p> <p>На получение от _____</p> <p style="text-align: center;">(наименование поставщика)</p> <p>Материальных ценностей по _____</p> <p style="text-align: center;">(наименование, номер, вид документа)</p>																	
18	<p>АВАНСОВЫЙ ОТЧЕТ № _____ от _____</p> <p>Структурное подразделение _____</p> <p>Подотчетное лицо (Ф.И.О.) _____</p> <p>Профессия (должность) _____</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <tr> <th colspan="2" style="padding: 5px;">Наименование показателя</th> <th style="padding: 5px;">Сумма (р., к.)</th> </tr> <tr> <td rowspan="2" style="padding: 5px; text-align: center;">Предыдущий аванс</td> <td style="padding: 5px; text-align: center;">остаток</td> <td style="padding: 5px;"></td> </tr> <tr> <td style="padding: 5px; text-align: center;">перерасход</td> <td style="padding: 5px;"></td> </tr> <tr> <td colspan="2" style="padding: 5px;">Получен аванс: из кассы</td> <td style="padding: 5px;"></td> </tr> <tr> <td colspan="2" style="padding: 5px;">Итого получено</td> <td style="padding: 5px;"></td> </tr> <tr> <td colspan="2" style="padding: 5px;">Израсходовано</td> <td style="padding: 5px;"></td> </tr> </table> <p style="margin-top: 5px;">Отчет проверен к утверждению в сумме _____</p> <p style="text-align: right;">(сумма прописью)</p> <p>Главный бухгалтер _____</p> <p>Бухгалтер _____</p> <p>Кассир _____</p> <p>Сделать поле «Итого получено» вычисляемым в зависимости от вводимых данных.</p>	Наименование показателя		Сумма (р., к.)	Предыдущий аванс	остаток		перерасход		Получен аванс: из кассы			Итого получено			Израсходовано		
Наименование показателя		Сумма (р., к.)																
Предыдущий аванс	остаток																	
	перерасход																	
Получен аванс: из кассы																		
Итого получено																		
Израсходовано																		
19	<p>ГАРАНТИЙНЫЙ ТАЛОН № _____ от _____</p> <p>Поставщик _____</p> <p>Покупатель _____</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <tr> <th style="padding: 5px;">Товар</th> <th style="padding: 5px;">кол-во</th> <th style="padding: 5px;">ед.</th> <th style="padding: 5px;">цена</th> <th style="padding: 5px;">сумма</th> <th style="padding: 5px;">гарантия (месяц)</th> <th style="padding: 5px;">серийный номер</th> </tr> <tr> <td style="height: 30px;"></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table> <p style="margin-top: 5px;">Итого: _____</p> <p>Поля «Сумма» и «Итого» сделать вычисляемыми в зависимости от вводимых данных.</p>	Товар	кол-во	ед.	цена	сумма	гарантия (месяц)	серийный номер										
Товар	кол-во	ед.	цена	сумма	гарантия (месяц)	серийный номер												

20	РЕГИСТРАЦИОННАЯ КАРТОЧКА владельца продукции Samsung	
	Фамилия	
	Имя	
	Отчество	
	Страна	
	Город	
	Индекс	
	Улица, дом, корпус, квартира	
	Электронная почта	
	Пол	
	Образование	
	Род занятий	
	Общий доход на семью в месяц (доллары США)	

#### 4 Литература, рекомендуемая для изучения

4.1 **Гарнаев А.Ю.** VBA./А.Ю.Гарнаев. –СПб.: БХВ-Петербург, 2005. -848 с.

4.2 **Гарнаев А.Ю.** Excel, VBA, Internet в экономике и финансах /А.Ю.Гарнаев. –СПб.: БХВ-Петербург, 2003. -423 с.

4.3 **Король В.И.** Visual Basic 6.0. Visual Basic for Applications 6.0. Язык программирования: справочник с примерами /В.И.Король. –М.: КУДИЦ-ОБРАЗ, 2000. -448 с.

4.4 **Назаров С.В.** Программирование на MS Visual Basic: учебное пособие /С.В.Назаров, П.П.Мельников; под ред. С.В.Назарова. –М.: Финансы и статистика, 2002. – 320 с.

4.5 **Экономическая информатика:** учебник /Под ред. В.П.Косарева. – 2-е изд., перераб. и доп. – М.: Финансы и статистика, 2004. – 592 с.