

Министерство образования и науки Российской Федерации

**Орский гуманитарно-технологический институт (филиал)
федерального государственного бюджетного образовательного учреждения
высшего образования «Оренбургский государственный университет»
(Орский гуманитарно-технологический институт (филиал) ОГУ)**

Р.Р. Абдулвелеева

**ФОРМИРОВАНИЕ ПРОФЕССИОНАЛЬНОЙ КОМПЕТЕНТНОСТИ
БАКАЛАВРОВ В ПРОЦЕССЕ ОСВОЕНИЯ
ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ**

Электронное учебное пособие

Орск 2016

© Абдулвелеева Р. Р., 2016

© Орский гуманитарно-технологический институт
(филиал) ОГУ, 2016

© Издательство Орского гуманитарно-
технологического института (филиала) ОГУ, 2016

ISBN 978-5-8424-0840-5

УДК 004.4
ББК 32.973-018р
А13

Научный редактор

*Уткина Т. И., доктор педагогических наук, профессор,
заведующий кафедрой математики, информатики и физики
Орского гуманитарно-технологического института (филиала)
ОГУ*

Рецензенты:

*Маркова А. Н., кандидат педагогических наук, преподаватель
ГАПОУ «Педагогический колледж г. Орск»;*

*Пергунов В. В., кандидат физико-математических наук,
доцент Орского филиала АОЧУ ВО «Московский финансово-
юридический университет МФЮА»*

А13 Абдулвелеева, Р. Р. **Формирование профессиональной компетентности бакалавров в процессе освоения объектно-ориентированного программирования** : электронное учебное пособие / Р. Р. Абдулвелеева. – Орск: Издательство Орского гуманитарно-технологического института (филиала) ОГУ, 2016 - ISBN 978-5-8424-0840-5. Режим доступа: http://library.ogti.ru/global/metod/metod2016_11_07.pdf

Подключение к сети Интернет на скорости не ниже 64 кбит/сек, современный браузер с надстройкой для чтения формата PDF (Portable Document Format) и/или возможностью скачивания файла для последующего просмотра с помощью программ для просмотра файлов в формате PDF

Программные средства, использованные при подготовке электронного издания: Microsoft Word 2013, Adobe Acrobat pro 10.

Редактор
Е. В. Кондаева

Редактор 2 категории
Г. А. Чумак

Технический редактор
Ю. И. Базлин

Утверждено редакционно-издательским советом Орского гуманитарно-технологического института (филиала) ОГУ в качестве электронного учебного пособия.

Объем издания – 1,05 мБ.

**Издательство Орского гуманитарно-технологического института
(филиала) федерального государственного бюджетного
образовательного учреждения высшего образования
«Оренбургский государственный университет»**

462403, г. Орск Оренбургской обл., пр. Мира, 15А

Содержание

Введение

1. Объектно-ориентированное программирование в среде VBA

1.1. Основные понятия объектно-ориентированного программирования

1.2. Технология объектно-ориентированного программирования

1.3. Практикум по объектно-ориентированному программированию в среде

VBA

1.3.1. Создание приложений в VBA с использованием MacroRecorder

(2 ч.)

1.3.2. Диалоговые окна приложений VBA. Использование функций (4 ч.)

1.3.3. Работа с модулями формы в VBA (4 ч.)

1.3.4. Создание объектно-ориентированного приложения в VBA (4 ч.)

1.3.5. Создание элементов базы данных объектно-ориентированного приложения в VBA (4 ч.)

1.3.6. Разработка диалоговых окон базы данных

2. Альтернативное программирование

2.1. Программное обеспечение на базе ОС Linux

2.2. Программирование с использованием LibreOffice

Библиографический список

Приложение 1

Приложение 2

Приложение 3

Приложение 4

Приложение 5

Приложение 6

Введение

В учебном пособии предлагается описание процесса использования принципов и методов объектно-ориентированного программирования на примере разработки макросов, форм, диалоговых окон и приложений. Предложены коды и скриншоты программных продуктов в качестве образцов для выполнения лабораторных работ, даны пошаговые инструкции по выполнению алгоритма программ с применением для этой цели языка программирования VBA, табличного процессора Microsoft Excel, текстового процессора Microsoft Word, реляционной системы управления базами данных Microsoft Access. Представлены формулировки заданий с кодами и скриншотами по реализации лабораторных работ с использованием ОС GNU/Linux и LibreOffice в школьном курсе информатики.

Материалы учебного пособия могут быть использованы в процессе подготовки бакалавров по дисциплинам «Объектно-ориентированное программирование», «Визуальное программирование», «Офисное программирование», при итоговой аттестации студентов направления подготовки 44.03.01 – Педагогическое образование (профиль Информатика и ИКТ), при подготовке курсовых и выпускных квалификационных работ, а также могут быть полезными для слушателей курсов переподготовки, курсов повышения квалификации учителей информатики, преподавателей информатики учебных заведений.

1. Объектно-ориентированное программирование в среде VBA

1.1. Основные понятия объектно-ориентированного программирования

Метод программирования, реализующий объекты в виде главных элементов программы, называется объектно-ориентированным программированием (ООП). Объект – основное понятие ООП. Объект – совокупность свойств (данных данного объекта), методов их обработки (подпрограмм) и событий (реакций объекта), приводящих к изменению свойств объекта.

Класс – фундаментальное понятие ООП. Класс является описанием поведения его представителя. Класс – это шаблон, программного объекта, в котором описаны свойства и методы, задающие поведение объектов этого класса. Экземпляр класса – объект – имеет структуру этого класса. Каждый объект является представителем (экземпляром) определенного класса.

Иногда под термином «класс» понимают тип структурированных данных, в этом случае объект будет являться структурированной переменной типа класс. В ней содержится информация о предмете или понятии, реализуемом в данной программе.

В описании класса нельзя определять конкретные значения полей (элементов), входящих в этот класс. Это вызовет мгновенную ошибку компилятора объектно-ориентированного языка. Класс определяет только общий тип объекта. Далее в программе необходимы выполнения действий над значениями, в том числе и над полями описанного класса. Помимо всех методов, входящих в класс кроме полей, функцию (set-функцию) инициализации (присвоения) значений класса должен выполнить специальный метод данного класса – конструктор. Эта функция создаёт конкретный экземпляр класса – объект. Конструктор – метод, имеющий имя класса и запускается столько раз, сколько разных объектов одного и того же класса нам необходимо для работы программы. А если при этом конструктором был динамически выделен участок памяти под данный экземпляр класса и его необходимо очистить, применяют специальный

метод – деструктор. Деструктор запускается только один раз. Объекты одного и того же класса имеют поля и методы, схожие по своим свойствам и правилам поведения. Говорят, что объекты имеют одинаковый интерфейс. Интерфейс класса иногда называют особенностями класса.

Объект – это частный случай обобщённой структуры ООП – класса (участок памяти), который содержит данные и правила (методы с кодом алгоритма), определяющие:

- поля объекта, то есть атрибуты исходных данных, своего рода переменная – посредник между самим обобщённым классом и пользователем, который определяет текущее свойство данного объекта через общение с этой самой переменной;

- методы объекта, представляющие собой определённое количество функций и процедур, которые в ответ на их вызов (переданное сообщение члену объекта) выполняют тот или иной алгоритм действий над этими членами экземпляра класса;

- свойства того или иного объекта, которые объявляются для получения и передачи атрибутов данных описания индивидуальности объекта посредством специальных переменных (property).

«Объявить класс» означает ввести необходимые свойства объектов: поля, методы и свойства объектов с определением предка данного класса в том числе.

Известно, что у объекта поля и функции по интерфейсу определяются как `public` – открытые, и как `private` – закрытые или приватные.

Выполнение объектно-ориентированной программы представляется последовательным вызовом сообщений (методов) соответствующих объектов. Объект при этом начинает выполнение кода определённой функции (процедуры) с полученными значениями переменных. Послать сообщения объекту в коде программы означает записать имя метода по его имени и с заданными аргументами непосредственно после указания имени вызываемого объекта. В этом и заключается такое взаимодействие объектов, как передача сообщений между этими объектами.

ООП предполагает, что любая функция (процедура) со структурированными переменными в программе успешно определяет объект некоторого класса как тип данных либо уже имеющегося в компиляторе, либо полученного из уже существующих, для определения любого задуманного физического предмета через его абстрагирование в виде одного класса и с использованием уже существующих классов (встроенных или пользовательских) – программирование «от класса к классу».

В основу ООП положены принципы программирования: наследование, инкапсуляция, полиморфизм.

Наследование позволяет новому классу-потомку, созданному на базе уже существующего, иметь (наследовать) все свойства класса-родителя.

Инкапсуляция – данные и свойственные им процедуры обработки, объединенные в одном объекте.

Полиморфизм (греч., poly – много, morphos – форма) – много форм. Полиморфизм – возможность различных реализаций одного и того же класса с одинаковой спецификацией (общими свойствами) в зависимости от типа обрабатываемых данных, то есть полиморфизм определяет разнообразные формы реализации действий с одинаковыми именами. Полиморфизм означает информированность объектов об используемых методах в зависимости от месторасположения.

Модульность – свойство программ, которое позволяет осуществлять свободное копирование и внедрение одного объекта в другие. Модуль включает в себя объекты с полным определением их характеристик, методов и свойств.

[Вернуться к содержанию](#)

1.2. Технология объектно-ориентированного программирования

Основы объектного программирования рассмотрим на примере реализации приложений с помощью языка программирования Visual Basic for Applications (VBA), который является подмножеством мощного языка объектно-ориентированного программирования Visual Basic.

В VBA можно использовать имеющиеся объекты и создавать новые объекты, программирование представляет собой управление объектами приложения.

Например, распишем, какими именно объектами управляет приложение, созданное в Excel. Упрощенная иерхическая архитектура выглядит так:

```
Application
  Workbook
  .....
  Worksheets
  .....
  Cell
  .....
```

В данном случае главный объект – приложение. В приложении, созданном на основе Excel, может быть несколько книг (Workbook), внутри которых находятся листы (Worksheets), листы разбиты на ячейки (Cell).

Класс в VBA представляется в виде определённого модуля – это обобщённое описание объектов со своими заданными свойствами и функциональностью этих объектов, состоящих в определённой иерархии, во главе которой определён контейнер для других типов. Говорят, что любое приложение VBA с такой иерархической особенностью представляется как объектная модель приложения (интернет ресурс 1).

Рассмотрим объявление пользовательских классов в VBA.

На этапе проектирования программного продукта, когда начинается создание класса, определяют свойства и методы для объектов нового класса.

Основные свойства, события и методы объектов VBA приведены в приложениях.

Например, в некотором приложении существует или создан класс объектов «Студент». Представителем класса или экземпляром класса, или объектом может быть «Студент Иванов». Этот объект должен иметь некоторые свойства, уникальные для каждого экземпляра класса. Свойствами объекта может быть имя, отчество, фамилия, пол, дата и место рождения, паспортные данные и т. д. Методами может быть возможность редактирования информации о студенте, полнота отображения информации.

Приведем пример псевдокода описания объекта:

```
// псевдокод
Студент Студент { Fam: строка           Name: строка
                  Sec: строка  логический Contact: строка }
Студент.Полное_Name: строка, only ready
Студент.Полное_Description: строка, only ready
Студент.Править_Description (New_Fam, New_Name, New_Sec,
                             New_Date_Birth, New_Sex, New_Contact)
```

Подобным образом описывается пользовательский тип данных. Объектный тип отличается тем, что не только данные (Fam, Name и др.), но и код (Студент.Полное_Name и др.) размещены совместно, то есть инкапсулированы.

Вообще, объявление класса в приложениях VBA осуществляется в модуле класса, в эту программную единицу помещается весь код создаваемого класса.

В VBA нет специальных кодов, которые применяют при определении класса (кроме определения свойств), поэтому рассмотрим процесс создания модуля класса по этапам:

- Открыть редактор Visual Basic и активизировать окно модуля класса, щелкнув Вставка\Модуль класса – так мы определили в проекте модуль нашего класса.

- Задать имя (Name) модуля в окне свойств (F4). В нашем примере имя Cstudent получит и модуль, и образованный класс.

- В окне дескриптора модуля объявляем приватные поля класса. Используем стандартные переменные (Private) для описания свойств объекта.
- Присваиваем начальные значения полей с помощью метода C_Init. Это и есть конструктор класса, определяющий начальные свойства объекта.
- Устанавливаем читающее свойство объекта – Property Read, записывающее – Property Write и свойства объекта – Property Set.
- Систематизируем известные процедуры и функции в методы класса.
- Далее определяем метод C_Term – деструктор класса, создаваемый для удаления существующего объекта из памяти по завершении работы с ним.

Таким образом, объект "Студент" получен проектированием класса CStudent.

Полный код модуля с комментариями приведен в листинге 1 (листинг 1 можно изменить на логически другой).

Модуль класса

```

‘ через свойства Private fName As String напрямую
‘ доступен к закрытым членам класса
‘ Private fFam As String фамилия
‘ Private fName As String имя
‘ Private fSec As String отчество
‘ дата рождения Private fBirth As Date
‘ пол Private fSex As String
‘ адрес и телефон Private fContact As String
‘ объектное поле создаёт конструктор
Private Sub C_Init ()
fFam = ""
fName = ""
fSec = ""
fContact = ""
End Sub
‘ only ready свойства
Public Property Get Fam() As String

```

```
Name = fName
End Property
Public Property Get FName() As String
Name = fName
End Property
Public Property Get Sec() As String
Sec = fSec
End Property
Public Property Get Birth() As Date
    Birth = fBirth
End Property
Public Property Get Sex() As String
    Sex = fSex
End Property
Public Property Get Contact() As String
    Contact = fContact
End Property
' write свойства
Public Property Let Fam(ByVal NV As String)
    fFam = NV
End Property
Public Property Let Name(ByVal NV As String)
    fName = NV
End Property
Public Property Let Sec(ByVal NV As String)
    fSec = NV
End Property
Public Property Let Birth (ByVal NV As Date)
    fBirth = NV
End Property
Public Property Let Sex(ByVal NV As String)
    fSex = NewValue
```

```

End Property
Public Property Let Contact(ByVal NV As String)
    fContact = NV
End Property
' only write свойства
Public Property Get Name_F() As String
    Name_F = fFam & " " & fName & " " & fSec
End Property
Public Property Get Info_F() As String
    Info_F = fFam & Chr(9) & fName & Chr(9) & fSec & Chr(9) &
        CStr(fBirth) & Chr(9) & fSex & Chr(9) & fContact
End Property
' свойства объекта изменяет следующий метод
' Optional – необязательные поля,
' значит – не все поля класса можно изменять
Public Sub Info_E(Optional Fam As String, Optional Name As String,
    Optional Sec As String, Optional Birth As Date, Optional Sex As
String,
    Optional Contact As String)
    fFam = Fam
    fName = tName
    fSec = Sec
    fBirth = Birth
    fSex = Sex
    fContact = Contact
End Sub

```

Листинг 1. Код создания модуля класса

Приведем дополнительные пояснения к коду листинга 1 в части синтаксиса:

- Описываются элементы *класса* так же, как и переменные практически любого языка программирования. Модификатор `Private` ограничивает доступ к

элементам класса из других программных единиц и разрешает использование этих элементов только внутри тела класса.

- `C_Init` – конструктор, предназначенный для начальной инициализации и других начальных действий с элементами класса.

При создании объекта как экземпляра класса конструктор запускается автоматически.

Для объявления свойств класса с ключевым словом `Property` используются функции: `Property Get` – для чтения, `Property Let` – для записи в свойства базовых типов, `Property Set` – для определения элементов типа объект (см. листинг 2).

Общий синтаксис объявления свойств:

```
[Public | Private] [Static] Property Get <имяФункции> [(<списокАргументов>)] [As <тип>]
```

```
<операторы>
```

```
имяФункции = возвращаемое значение
```

```
[Exit Property] <операторы>
```

```
имяФункции = возвращаемое значение
```

```
End Property
```

```
' это для чтения
```

```
[Public | Private] [Static] Property Let <имяПроцедуры> [(<списокАргументов>)]
```

```
<операторы>
```

```
[Exit Property]
```

```
<операторы>
```

```
End Property
```

```
' это для записи
```

```
[Public | Private] [Static] Property Set <имяПроцедуры> [(<списокАргументов>)]
```

```
<операторы>
```

```
[Exit Property]
```

```
<операторы>
```

```
End Property
```

```
' работаем элементами класса типа объект
```

Самостоятельно использоваться процедуры и функции, являющиеся методами некоторого класса пакета VBA, не могут.

Как и любой язык объектно-ориентированного программирования, VBA поддерживает одну из основных концепций ООП – наследование. Однако нет прямого, классического способа организации классов-потомков с использованием созданных ранее классов. Пример не прямой, последовательной вставки (встраивания) объектов приведен в листинге 2. Создав в первую очередь классы CStud (Студент) и CFac (Факультет), создаем класс CGr (Группа данного факультета).

' CFac – модуль класса с полученным объектом "Факультет"

Private fTit As String

' Название факультета

Public Property Get Tit () As String

Tit = fTit

End Property

Public Property Let Tit (NewTit As String) f

Tit = NewTit

End Property ...

' Модуль класса CGr – объект "Группа"

' Вставка объектов данного класса

Private fFac As New CFac

Dim fStud() As New CStud

' определение значения объекта

Public Property Set Fac(Title)

Set fFac = Tit

End Property

' определение факультета

Public Property Get Fac () As String

Fac = fFac.Tit

End Property

Листинг 2. Встраивание объектов

Опишем работу над элементами объекта.

Использование класса в программе осуществляется через переменные – экземпляры этого класса. Эти переменные, называются объектными, их можно создать через класс объекта или ссылкой на объект. Продемонстрируем установление связи переменной объекта с самим объектом:

```
Dim <Переменная> As <класс Объекта>
```

Раннее связывание позволяет уже во время трансляции выполнять проверку допустимости операций над объектами.

```
Dim <Переменная> As Object
```

Позднее связывание (динамическое) при исполнении приложения Переменную связывает с объектом того или иного класса.

Ключевое слово New создаваемому экземпляру класса (при раннем связывании) выделяет память и вызывает конструктор класса. Примеры объявления объектной переменной в стандартных модулях и модулях класса:

```
Public Fac As New CFac
```

```
Private Gr (3) As New CGr
```

```
Dim Stud As New CStud
```

Способ «раннее связывание» даёт возможность получить доступ к свойствам и методам класса на этапе разработки в среде VBE: при вводе переменной объекта всплывает список всех имеющихся операций для данного объекта.

Рассмотрим процесс использования объектов.

В VBA принято использовать точки при обращении к свойствам, методам экземпляра класса:

перем = <имяОбъектнойПеременной>.<Свойство> ' чтение свойства
или:

<имяОбъектнойПеременной>.<Свойство>= значение 'запись свойства

Ключевое слово Set используется для обращения к объектным свойствам. Вызов методов и передача параметров аналогичны работе с обычными процедурами и функциями. В листинге 2 приведены примеры работы с экземплярами созданных классов.

Рассмотрим объектную модель компонентов MS Office и библиотеки типов. Пакет MS Office – это программный офисный пакет, в состав которого входят внутренние, взаимодействующие между собой объекты. Являясь единицами ООП, они характеризуются определёнными свойствами и методами. Взаимодействующие между собой объекты с набором специфичных свойств и методов образуют объектную модель, которая является основой приложения MS Office. В виде объектов представлены все доступные разработчику элементы. Офисные программы (приложение – объект Application) состоят из элементов – объектов, доступных программисту, разработчику проектов. Изучив структуру объектной модели Office, можно создавать приложения офисной работы с документами. Основополагающими при этом являются единицы объектной модели приложения Microsoft Office – одноимённые библиотеки, в которых размещаются объекты и семейства объектов. VBA, Office, Word, Excel и Access имеют соответствующие стандартные библиотеки, библиотеки объектов и прочие объектные библиотеки со своим набором различных функциональных возможностей.

Просмотреть указанные библиотеки открытого приложения Office (например, в Access) можно, запустив редактор Visual Basic for Application, далее нужно выбрать команду меню "View(Вид)/Object Browser (Просмотр Объектов)" и раскрыть список Project/Library(Проекты/Библиотеки)" (см. рис. 1) с объектными библиотеками Microsoft Word.

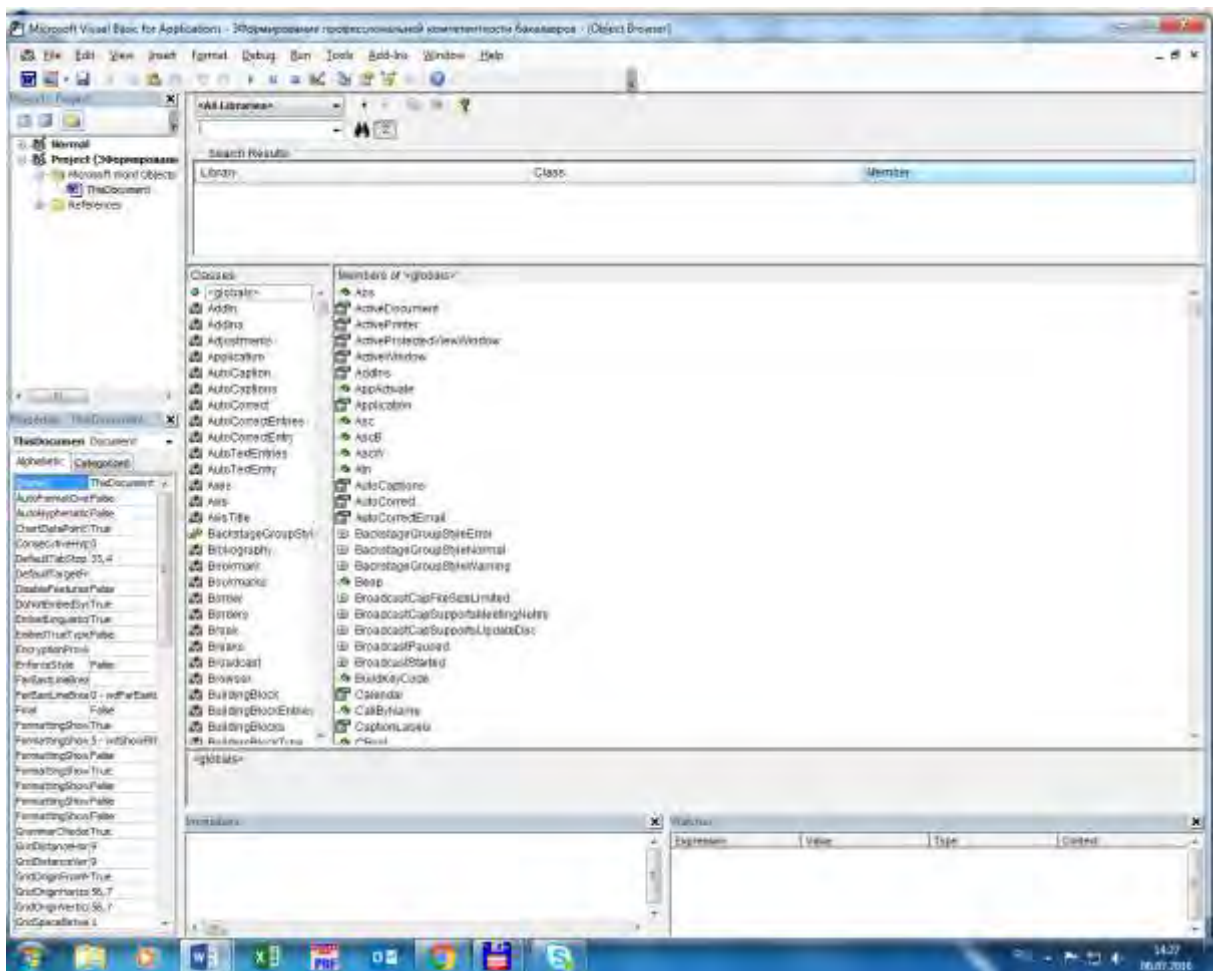


Рисунок 1. Объектные библиотеки Microsoft Word

Word – библиотека с объектами для построения документов Word. Имеются класс с корневым объектом `Word.Application` и классы объектов, из которых состоит корневой объект. Каркас создаваемых документов строится автоматически в момент создания этих документов.

Office – библиотека (например, `CommandBar` с классами инструментальных панелей), состоящая из общих для приложений Office объектов, классы, определяющие "помощник" (объект `Assistant`).

Stdole – библиотека классов для организации ActiveX-объектов (OLE-объектов) с функциями OLE-автоматизации.

VBA – библиотека классов для работы с кодированием на языке VBA (встроенные в язык функции и константы), `Collection`, `Err` и др.

При создании проекта (Project) по умолчанию для некоторого офисного документа разработчик конструирует пользовательские классы (методы, свой-

ства), которые при этом, как и встроенные в стандартные библиотеки классы, доступны для просмотра.

Для хранения вышеперечисленных классов, доступных для всех документов Word, используется проект Normal.

Объекты общего назначения библиотеки VBA

Коллекции (класс Collection) офисного языка программирования есть, по сути своей, ссылки на упорядоченные объекты – экземпляры данного класса, объединённые в единое целое. Коллекция представляется структурой списка значений различных типов и набора определённых операций для них (создания коллекции, добавления новой формы в коллекцию методом Add, удаления элементов коллекции методом Remove, определения элемента по его индексу методом Item). Работать со списком коллекции целесообразно с применением инструкции цикла.

Рассмотрим пример коллекции всех инициализированных форм некоторого приложения UserForms. В нём представлены следующие свойства, которые определяют:

- количество действующих форм (Count);
- элемент (форму) коллекции (Item).

Другой пример – коллекция (класс Controls) всех задействованных элементов управления, помещённых на данной пользовательской форме UserForm (листинг 3). А именно производятся действия над свойством Visible (False) – представление соответствующего элемента прозрачным на форме.

```
' при выполнении приложения в проект добавлена и  
' загружена форма UserForm_1  
For Each m in UserForm_1.Controls  
    c.Visible = False  
Next
```

Листинг 3. Коллекции элементов управления

Объект Debug с 2 методами:

- безусловный вывод выражения в окно отладки (Print);

– вывод по указанному условию (Assert).

Информацию об ошибках объект Err выявляет, используя метод Raise с командой Error для генерации системных ошибок при создании классов. В случае необходимости ручного сброса свойств объекта Err используется метод Clear (см. рис. 2).

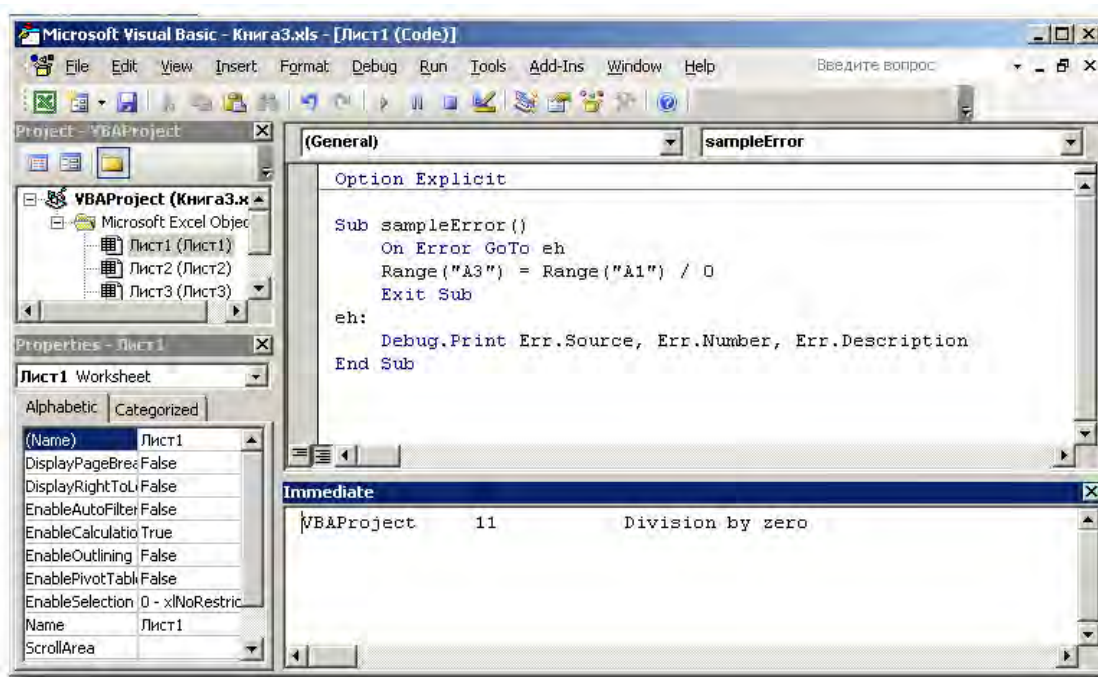


Рисунок 2. Использование объекта Err

Покажем некоторые из около полусотни объектов и коллекций Excel (полный перечень можно посмотреть в справочной системе).

В коллекции Workbooks объект Workbook использует множество свойств и методов для доступа к открытой рабочей книге Excel:

- рабочие листы (Sheets);
- определенный рабочий лист (Worksheet);
- диапазон ячеек (Range);
- текущее выделение (Selection);
- формат ячеек (CellFormat).

```

Sub ChangeCellFormat
  Range("A1").Select
  Selection.Interior.ColorIndex = 36
  MsgBox "Ячейка A1 залита жЕТЫМ ЦВЕТОМ"
  With Application
    .FindFormat.Interior.ColorIndex = 36
    .ReplaceFormat.Interior.ColorIndex = 36
  End With
  ActiveCell.Replace What:="", Replacement:="", LookAt:=xlPart, _
    SearchOrder:=xlByRows, MatchCase:=False, SearchFor-
mat:=True,
    _ReplaceFormat:=True
  MsgBox "Ячейка A1 залита зелЕНЫМ ЦВЕТОМ"
End Sub

```

Листинг 4. Программирование объектов рабочей книги

[Вернуться к содержанию](#)

1.3. Практикум по объектно-ориентированному программированию в среде VBA

Практикум состоит из 10 лабораторных работ. Однако для студентов – будущих учителей информатики – несколько вводных занятий по изучению ООП следует отвести для выполнения заданий из раздела «Объектно-ориентированное программирование на языке Visual Basic» школьного учебника Н. Д. Угринович [24]. Отметим, что большинство студентов не изучали VBA в школе, а будущему учителю информатики, возможно, придется работать в старшей школе именно по этому учебнику. Время, отведенное на выполнение этих заданий, зависит от уровня подготовленности студентов и от количества часов, выделенных на изучение ООП по учебному плану.

В разделе «Альтернативное программирование» данного пособия предлагается реализация предложенных в этом учебнике заданий с использованием ОС GNU/Linux и LibreOffice. Этот раздел включен в пособие для того, чтобы студент – будущий учитель информатики – при работе со старшеклассниками или с учащимися профильных классов не растерялся, если в школе в компьютерном классе будет установлена ОС GNU/Linux.

1.3.1. Создание приложений в VBA с использованием MacroRecorder (2 ч.)

Цель: научиться создавать макросы в VBA.

Задачи: научиться активизировать макрорекодер, изучить его возможности и выполнить процесс создания макросов с помощью макрорекодера.

Содержание лабораторной работы

Теоретическая часть

Макрос – это последовательность действий (программа, процедура), записанная на языке программирования VBA. Макрос можно неоднократно запускать, при этом Excel будет выполнять последовательность действий, кото-

рые пользователь задал макросу. Команды языка VBA, следовательно, и макросы можно реализовать в Excel, Word, Access или Outlook.

MacroRecorder – транслятор, который предназначен для создания программы – макроса, позволяющий переводить производимые пользователем действия на язык VBA. Активизация MacroRecorder происходит через последовательность команд: Сервис\Макрос\Начать запись (Tools, Macro, Record New Macro) \ запустите MacroRecorder на запись. В диалоговом окне Запись макроса (Record Macro) после нажатия кнопки ОК появится панель инструментов на которой имеется кнопка – Остановить запись (Stop Recording). Пока не будет нажата эта кнопка, все действия, производимые пользователем, будут записываться.

Созданный макрос запускают на выполнение следующими способами:

1. Вызвать меню: Сервис\Макрос\Макросы, выбрать из списка имя выполняемого макроса\Выполнить.

2. Нажатием комбинации клавиш, использованных при создании макроса, создать кнопку, под которой назначить макрос следующим образом: Вид\Панели инструментов \Настройка\закладка Команды, выбрать вид кнопки из списка Макросы \ перетащить с помощью мыши на панель инструментов \ выполнить команду “Изменить выделенный объект\Назначить макрос”.

3. Назначить кнопку в поле рабочего листа: Вид\Панель инструментов\Формы\ мышью перетащить кнопку из появившейся панели в поле рабочего листа\ назначить макрос.

4. Вставкой новой строки в существующее меню: Вид\Панели инструментов\Настройка\Команды, выбрать строку из списка Новое меню, мышью перетащить ее в строку основного меню\ выполнить команду Изменить выделенный объект\ Назначить макрос).

Макросы хранятся в программных модулях. В любой книге Excel можно создать любое количество программных модулей и разместить там макросы.

Практическая часть

Задание 1. Выполните предложенную последовательность команд по созданию макроса и изучите технологию его реализации.

Задание 2. Создайте макрос, который для некоторой функции $f(x)$ позволит построить шаблон таблицы соответствия значений аргументов x . Интервал изменения аргументов x , шаг разбиения областей задать числами, размещенными в некоторые ячейки. Запуск макроса должен обеспечить построение графика по таблице зависимости x от $f(x)$.

Задание 3. Создайте макрос, строящий шаблон таблицы рейтинга успеваемости студентов вашей группы. Запуск макроса выдает шаблон таблицы, в которой имеются названия всех строк и столбцов. После заполнения таблицы числами (рейтингами) автоматически строится горизонтальная столбиковая диаграмма с целью наглядного представления соотношения рейтингов.

Задание 4. Создайте самостоятельно полезный для вас макрос. Создать второй макрос, который очищает результаты работы первого.

Задания для самостоятельной работы

1. Проанализируйте содержание школьных учебников, задачников, практикумов на предмет наличия и полноты изучения макросов в школьном курсе информатики.

2. Предложите содержание практической работы по теме «Макросы» для факультативного занятия или для углубленного изучения приложений MS Office учащимися.

3. Изучите содержание олимпиадных заданий для учащихся учебных заведений на наличие задач по работе с макросами.

Контрольные вопросы

1. Что такое макрос?
2. Можно ли привязать макрос к графическому объекту?
3. Как записать макрос с помощью макрорекордера?
4. В чем суть работы макрорекордера?

5. Сколько часов отводится на изучение макросов в школе?

6. Как запустить макрос на выполнение?

Отчетность по работе

Оформите отчет в текстовом редакторе и включите в него следующие результаты выполнения заданий:

1. Результат выполнения задания 2, с описанием этапов выполнения задания.

2. Результат выполнения заданий 3 и 4.

Ответы на контрольные вопросы.

1.3.2. Диалоговые окна приложений VBA. Использование функций (4 ч.)

Цель: научиться создавать диалоговые окна в VBA.

Задачи: научиться создавать диалоговое окно на основе использования функции обработки строк.

Содержание лабораторной работы

Теоретическая часть

Изучим работу со встроенными диалоговыми окнами в процессе использования функций обработки строк. Диалоговые окна, встроенные в VBA, могут быть представлены в следующих видах: окна сообщений и окна ввода. Окно сообщений (MsgBox) позволяет выводить простые сообщения для пользователя приложения, а окно ввода (InputBox) позволяет осуществлять ввод информации.

Чтобы научиться работать со встроенными диалоговыми окнами, выполним задания с использованием функций обработки строк. Перечисление некоторых функций обработки строк предложено в таблице 1. Если для выполнения заданий нужная функция не найдена вами в таблице в явном виде, то следует учесть, что зачастую для достижения нужного результата эти функции приходится вкладывать одну в другую. Или воспользоваться более полным справочником.

Функции обработки строк

Название функции	Действие	Синтаксис (параметр)	Примечание
Asc	Возвращает ASCII – код первого смвола в строке	Asc (строка)	Если строка пустая, то возникает ошибка выполнения
Chr	Возвращает символ по заданному коду (преобразует ASCII код в строку)	Chr (код)	Актуальна, когда нужно вывести служебный символ
Lcase	Преобразует все буквы строки в строчные	Lcase (строка)	
Ucase	Преобразует все буквы в заглавные	Ucase (строка)	
Left (Right)	Возвращает подстроку (строку), состоящую из заданного числа первых символов исходной строки	Left (строка, длина)	Начиная слева (начиная справа)
Mid	Возвращает указанную часть строки	Mid (строка, стартовая позиция [, длина])	Если 3-й параметр не указан, то возвращаются все символы от стартовой позиции до конца строки
Len	Возвращает длину строки	Len (строка)	Определяет количество символов в строке
LTrim RTrim Trim	Возвращает копию строки без пробелов в начале. Удаляет пробелы в конце. Удаляет пробелы как в начале, так и в конце	Ltrim – (строка)	
Space	Создаёт строку, состоящую из указанного числа пробелов	Space (число)	(количество_пробелов)
String	Создаёт строку, состоящую из указанного числа повторений одного и того же символа	String (число, символ)	
StrComp	Сравнивает 2 строки	Str Comp (строка1, строка 2, способ сравнения)	Вернет 0 при равенстве
InStr	Возвращает позицию 1 вхождения одной строки внутри другой	InStr (строка1, строка 2)	Поиск последовательности символов
Replase	Возвращает замену в строке одних символов на другие	Replase (строка, найти, заменить, старт, число_замен, метод_сравнения)	Аргументы: старт, число_замен и метод_сравнения не обязательны. Число_замен используют, чтобы указать, сколько замен следует сделать в тексте

Практическая часть

Задание 1. Выполните простые примеры с использованием следующих функций:

а) строковой функции Len для возврата числа знаков в строке.

Подсказка кода:

' Initializes variable.

```
Dim TestString As String = "Hello World"
```

' Returns 11.

```
Dim TestLen As Integer = Len(TestString)
```

б) функции Replace

```
Dim sStr As String
```

```
    sStr = "Hello, World!"
```

```
    MsgBox Replace(sStr, "Hello", "Bay")
```

Задание 2. Создайте диалоговые окна для своего варианта задания (см. табл. 2).

Таблица 2

№	Задание
1.	Ввести строку символов. В строке удалить все повторяющиеся слова и вывести полученный результат на экран
2.	Ввести строку текста. Заменить все строчные символы прописными.
3.	Ввести строку символов. Определить, содержит ли строка слова – полиндромы (слова, читающиеся одинаково слева направо и справа налево) и если содержит, то вывести их на экран
4.	Введите строку текста, которая содержит арифметическое выражение, состоящее из чисел, соединенных знаками сложения, вычитания и умножения. Вычислить значение этого выражения и результат вывести на экран
5.	Отсортировать слова во введенной символьной строке по возрастанию длины слов. Результат вывести на экран
6.	Отсортировать слова во введенной символьной строке по алфавиту. Результат вывести на экран
7.	Введите строку текста. Сделайте реверс – «отзеркальте» строку, разместив ее символы в обратном порядке
8.	Ввести строку текста, в которой некоторое слово встречается несколько раз. Замените это слова на его синоним или антоним
9.	Введите строку с названием организации в кавычках и замените это название на другое
10.	Ввести строку текста. Поменяйте местами первое и последнее слова в предложении
11.	Ввести строку текста. Определить длину каждого слова и распечатать каждое слово с новой строки
12.	Ввести текст. Определить длину введенного текста. Определить количество слов в тексте
13.	Ввести текст. Удалить из текста пробелы и подсчитать длину получившегося текста
14.	Ввести предложение. Определить длину первого слова. Распечатать каждое слово предложения с новой строки
15.	Ввести предложение. Определить длину второго слова. Заменить все запятые в предложении на пробелы

Задание 3. Создать программу для вычисления значения функции $f(x)$. В программе использовать обращение к ячейкам листа Excel. Вызов программы организовать с листа Excel.

Задание 4. Создать программу для решения квадратного уравнения $ax^2 + bx + c = 0$ с использованием условного оператора Select...Case. Программа должна учитывать все значения дискриминанта: $D = 0$, $D > 0$, $D < 0$. Использовать обращение к ячейкам листа Excel, операторы ввода/вывода или кнопки. Вызов программы организовать с листа Excel. Примерное оформление задачи показано на рисунке 3. Добавьте кнопку Очистка и напишите для нее программу (результат очистки выполняет удаление данных из ячеек для ввода новых значений).

	A	B	C	D
	Решение квадратного уравнения $ax^2+bx+c=0$			
1				
2	Коэффициенты			
3	a	b	c	
4	9	1	1	
5				
6	Решение			
7	D	корней нет		
8	-35			
9				Решить уравнение

Рисунок 3. Пример оформления решения задачи

Задание 5. Разработать приложение с диалоговым окном «Регистрация клиентов отеля «Южный Урал», в котором администратор может выбрать:

- продолжительность проживания клиента (управляется «Счетчиком»);
- тип номера для проживания из списка: одноместный, двухместный, люкс; стоимость проживания в номерах соответственно равна 1000, 1500 и 2500 рублей в сутки;
- клиенту предлагается завтрак в кафе, при этом суточная оплата возрастает на определенную сумму;

- можно добавить дополнительные условия на ваше усмотрение;
- расчет общей стоимости проживания клиента осуществляется нажатием на кнопку «ок».

При этом все данные из диалогового окна должны отражаться в базе данных, организованной на рабочем листе.

Задания для самостоятельной работы

Задание 1. Выполните процесс создания пользовательских функций.

Учитывая тот факт, что VBA предоставляет возможность пользователю создавать собственные функции, с которыми работают на рабочем листе с использованием мастера функции, как и со встроенными функциями, процесс создания функции происходит через команды Excel Сервис/Макрос/Редактор Visual Basic и командами редактора Вставка/Модуль. Программу на VBA нужно оформить в виде подпрограммы Function. Вызов собственной пользовательской функции осуществляется командами Вставка/ Функция, далее выбрать категорию Функции, определенные пользователем.

Таблица 3

№	Задание
1	Функция нахождения наименьшего общего кратного двух натуральных чисел
2	Функция нахождения общего делителя двух натуральных чисел
3	Функция определения количества цифр в натуральном числе
4	Функция нахождения суммы цифр натурального числа
5	Перевести заданное в двоичной системе счисления число в восьмичную систему
6	Определить возможность построения треугольника с заданными длинами сторон
7	Даны произвольные коэффициенты квадратного уравнения. Определить количество корней этого уравнения
8	Определить, является ли пятизначное число симметричным
9	Определить, есть ли среди цифр заданного четырехзначного числа одинаковые
10	Произвольное десятичное целое число перевести в двоичную систему

Задание 2. Проанализируйте содержание школьных учебников, задачников, практикумов на предмет наличия и полноты изучения функций в школьном курсе информатики.

Задание 3. Предложите содержание практической работы по теме «Функции» для факультативного занятия или для углубленного изучения программирования учащимися.

Задание 4. Изучите содержание олимпиадных заданий для учащихся учебных заведений, заданий ЕГЭ на наличие задач по работе с функциями.

Контрольные вопросы

1. Что такое функция в программировании?
2. Каково назначение функций?
3. Приведите пример реализации функции.
4. Что выполнит программа при реализации следующего кода:

```
Dim n, nCharCode As Integer
n = 1
nCharCode = Asc("A")
Do While n <= 20
ActiveWorkbook.ActiveSheet.Range("A" & n). Value =
Chr(nCharCode)
n = n + 1
nCharCode = nCharCode + 1
```

Loop

5. Чем отличается пользовательская функция от встроенной?
6. В чем заключается отличие процедуры от функции и событийной процедуры от процедуры общего типа?

Отчетность по работе

Оформите отчет в текстовом редакторе и включите в него следующее:

1. Результат выполнения заданий, с описанием этапов выполнения задания.
2. Результат выполнения заданий для самостоятельной работы.
3. Ответы на контрольные вопросы.

1.3.3. Работа с модулями формы в VBA (4 ч)

Цель: научиться создавать приложения в VBA.

Задачи: научиться работать с переменными, процедурами, модулями формы, реализовывать функции в приложении VBA, изучить его возможности и выполнить процесс создания полноценного приложения.

Содержание лабораторной работы

Теоретическая часть

Ознакомьтесь с процессом обработки разработанного теста по источникам [5 и 10] из библиографического списка. Коротко разьясним, что одним из этапов апробации теста является формирование результирующей матрицы ответов в Excel, вид которой показан на рисунке 4, в которой строки соответствуют испытуемым, столбцы – заданиям теста, а в ячейки таблицы экспортируются ответы испытуемых, задаваемые в некоторой шкале. В компьютерной программе используется дихотомическая шкала (0 – не выполнено, 1 – выполнено).

Упорядоченная матрица результатов																									
#	ФИО	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	Студент 1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	1	1	1
2	Студент 2	1	1	1	0	1	1	0	1	0	1	0	1	1	1	1	0	1	1	1	1	0	0	0	0
3	Студент 3	1	1	1	0	1	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0
4	Студент 4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	Студент 5	1	0	1	0	0	1	0	1	1	1	1	1	1	0	0	0	1	1	0	1	0	1	0	1
6	Студент 6	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	0	1	0	1	0	1	0	0	1
7	Студент 7	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	Студент 8	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
9	Студент 9	1	0	0	0	1	1	0	1	0	1	0	1	0	1	0	1	1	1	0	0	1	0	1	1
10	Студент 10																								
11	Студент 11																								
12	Студент 12																								
13	число выполненных ответов	9	6	8	1	7	8	1	7	6	8	7	7	8	7	5	6	7	8	5	6	4	5	4	4
Упорядоченная матрица результатов																									
#	ФИО	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
13	Студент 1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
14	Студент 2	0	0	0	0	1	1	0	1	0	1	0	1	0	1	0	1	1	1	0	0	1	0	1	1
15	Студент 3	0	1	1	1	1	1	1	0	1	1	1	1	1	0	1	0	1	0	1	0	0	1	0	0
16	Студент 4	1	0	1	0	0	1	0	1	1	1	1	1	1	0	0	0	1	1	0	1	0	1	1	1
17	Студент 5	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0
18	Студент 6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	Студент 7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	Студент 8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21	Студент 9	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
22	Студент 10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
23	Студент 11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
24	Студент 12	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
25	Студент 13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
26	число выполненных ответов	7	6	8	5	7	8	1	7	6	7	7	7	8	7	5	6	7	8	5	6	4	5	4	4
27	ИИ	0,78	0,67	0,89	0,58	0,78	0,89	0,58	0,78	0,67	0,78	0,78	0,78	0,89	0,78	0,58	0,67	0,78	0,89	0,67	0,67	0,67	0,67	0,67	0,67
28	СКО	0,22	0,33	0,11	0,44	0,22	0,11	0,44	0,22	0,33	0,22	0,22	0,22	0,11	0,22	0,44	0,33	0,22	0,11	0,33	0,33	0,33	0,33	0,33	0,33
29	ВЕРЯНОСТЬ	0,29	0,01	0,29	0,53	0,81	0,80	0,44	0,68	0,05	0,80	0,57	0,82	0,80	0,09	0,58	0,81	0,40	0,10	0,52	0,79	0,38	0,55	0,41	0,33
30	СР зная (исход)	0,78	0,67	0,89	0,58	0,78	0,89	0,58	0,78	0,67	0,78	0,78	0,78	0,89	0,78	0,58	0,67	0,78	0,89	0,67	0,67	0,67	0,67	0,67	0,67

Рисунок 4. Матрица результатов

Эта матрица позволяет получить необходимые величины и параметры для расчета характеристик теста. Характеристики теста, которые подвергаются изучению, показаны на рисунке 5.

32	Размах индивидуальных бал	32	рассчитывается по формуле Коэн-Хэпп
33	Среднее выборочное	27,3	рассчитывается по формуле $\bar{X} = (X_1 + X_2 + \dots + X_N) / N$
34	дисперсия	104	рассчитывается по формуле $s_x^2 = \frac{K \sum_{i=1}^K X_i^2 - \left(\sum_{i=1}^K X_i \right)^2}{K(K-1)}$
35	Надежности теста	0,7	07-0,79 оценка надежности удовлетворительная. $KR_{20} = \frac{N}{N-1} \left(1 - \frac{\sum_{i=1}^N P_i Q_i}{s_x^2} \right)$
	Валидность каждого вопроса (представлены в таблице выше)		рассчитывается по формуле $r = \frac{\sum_{i=1}^K X_i Y_i - \frac{1}{K} \sum_{i=1}^K X_i \sum_{i=1}^K Y_i}{\sqrt{\sum_{i=1}^K X_i^2 - \frac{1}{K} \left(\sum_{i=1}^K X_i \right)^2} \sqrt{\sum_{i=1}^K Y_i^2 - \frac{1}{K} \left(\sum_{i=1}^K Y_i \right)^2}}$

Рисунок 5. Формулы для расчета характеристик теста

Практическая часть

Задание 1. Используя результаты изучения теоретической части данной лабораторной работы, создайте приложение, примерный вид диалогового окна которого показан на рисунке 6.

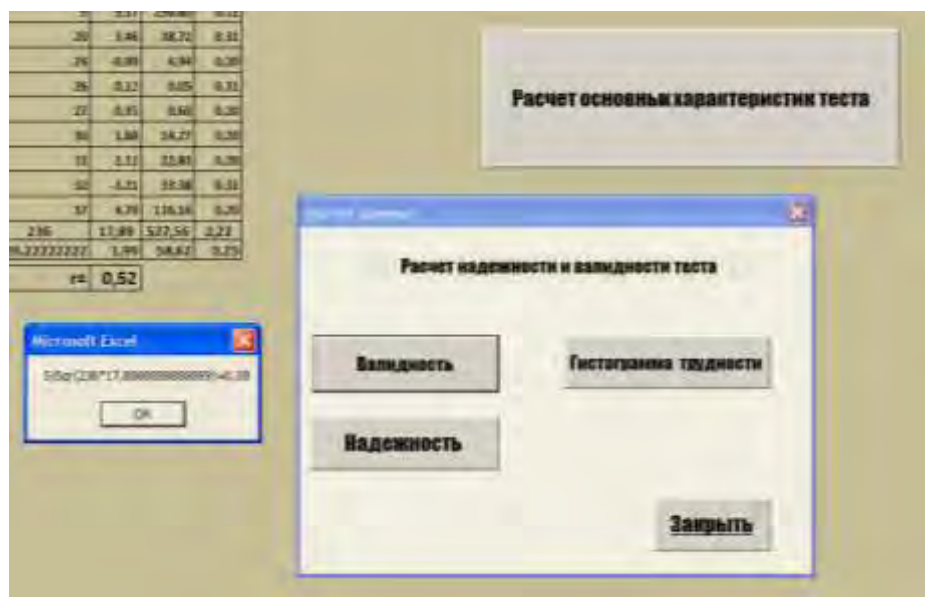


Рисунок 6. Окно приложения «Расчет основных характеристик теста»

Задания для самостоятельной работы

Задание 1. Дополните приложение, разработанное в практической части, недостающими характеристиками, используя источник 18, 25.

Задание 2. Добавьте элементы управления на разработанную форму на ваше усмотрение.

Контрольные вопросы

1. Какие типы переменных используются в VBA?
2. Перечислите операторы объявления переменных.
3. Что такое область видимости и время жизни переменной?
4. Перечислите типы функций, которые используются в VBA.
5. Как вызвать функцию рабочего листа?
6. Какие операторы цикла используются в VBA?
7. В чем отличие цикла с постусловием от цикла с предусловием?
8. В чем особенность оператора цикла объектного типа?
9. Какие операторы используются для принудительного выхода из цикла?
10. Можно ли задать несколько условий: а) в одном условном операторе; б) в одном операторе выбора?
11. Какие окна редактора VBA используются во время отладки программ?
12. Как во время прерывания программы определить значения переменных?

Отчетность по работе

Оформите отчет в текстовом редакторе и включите в него следующие результаты выполнения заданий:

1. Результат выполнения задания 1, с описанием этапов выполнения задания.
 2. Результат выполнения заданий для самостоятельной работы.
- Ответы на контрольные вопросы.

1.3.4. Создание объектно-ориентированного приложения в VBA (4 ч.)

Цель: научиться создавать объектно-ориентированные приложения в VBA.

Задачи: научиться работать с модулями формы, реализовывать функции базы данных в приложении VBA, изучить его возможности и выполнить процесс создания полноценного приложения с базой данных.

Содержание лабораторной работы

Теоретическая часть

В данной лабораторной работе теоретическая часть раскрывается при реализации практической части. Для более полного понимания сути предлагаемой ниже практической части можно воспользоваться любым практическим руководством по изучению среды VBA.

Предложенная ниже в практической части теория предполагает репродуктивный характер работы студентов, то есть коды даны и нужно их реализовать. Однако простое копирование кодов без осмысления студентом их сути, без вдумчивого «прочитывания» листинга программы приведет к большим трудностям при выполнении самостоятельных заданий в следующих лабораторных работах.

Практическая часть

Разработаем приложение, которое позволит работать с базой данных студентов некоторого учебного заведения. В нашем проекте предполагается разработать балльно-рейтинговую систему оценки достижений студентов Орского гуманитарно-технологического института (филиала ОГУ). Первоначально нужно создать базу данных, в которой будут находиться все данные о студентах вуза. В вузе имеется несколько факультетов, на каждом факультете имеются направления подготовки (специальности), по каждой специальности есть несколько групп студентов, обучающихся на разных курсах. Например, студенты бакалавриата учатся с 1-го по 4-й курс, в каждой группе есть студенты, которых можно будет идентифицировать по фамилии, имени, отчеству, а так как эти

данные могут совпадать, то «ключевым полем» выберем номер зачетки студента. Структура таблицы базы данных «Студент» показана на рисунке 7.

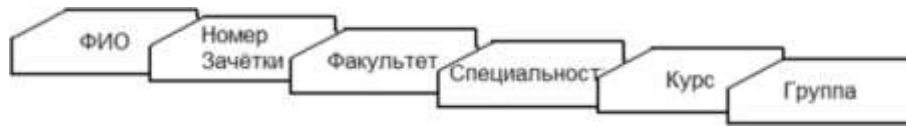


Рисунок 7. Структура таблицы «БазаДанных»

Все эти структурные компоненты будут размещены на форме в диалоговом окне.

Значения свойств и содержимое формы *ВводДанных БД "Студент"* определены в таблице 4.

Таблица 4

Элементы диалогового окна базы данных «Студент»

Объект	Свойство/Метод	Значение
<i>UserForm</i> (Форма)	<i>Name</i> (Имя)	<i>frmVvod</i>
	<i>Caption</i> (Заголовок)	<i>ВводДанных БД "Студент"</i>
<i>ComboBox</i> (ПолеСоСписком)	<i>Name</i> (Имя)	<i>ComboBox1</i> <i>ComboBox2</i> <i>ComboBox3</i>
	<i>AddItem</i> (Элемент)	<i>название факультетов</i> <i>название направлений</i> <i>название курсов</i>
<i>TextBox</i> (ТекстовоеПоле)	<i>Name</i> (Имя)	<i>TextBox3</i> , <i>TextBox1</i> , <i>TextBox2</i> , <i>txtN</i>

Примерное размещение элементов формы в диалоговом окне показано на рисунке 8.

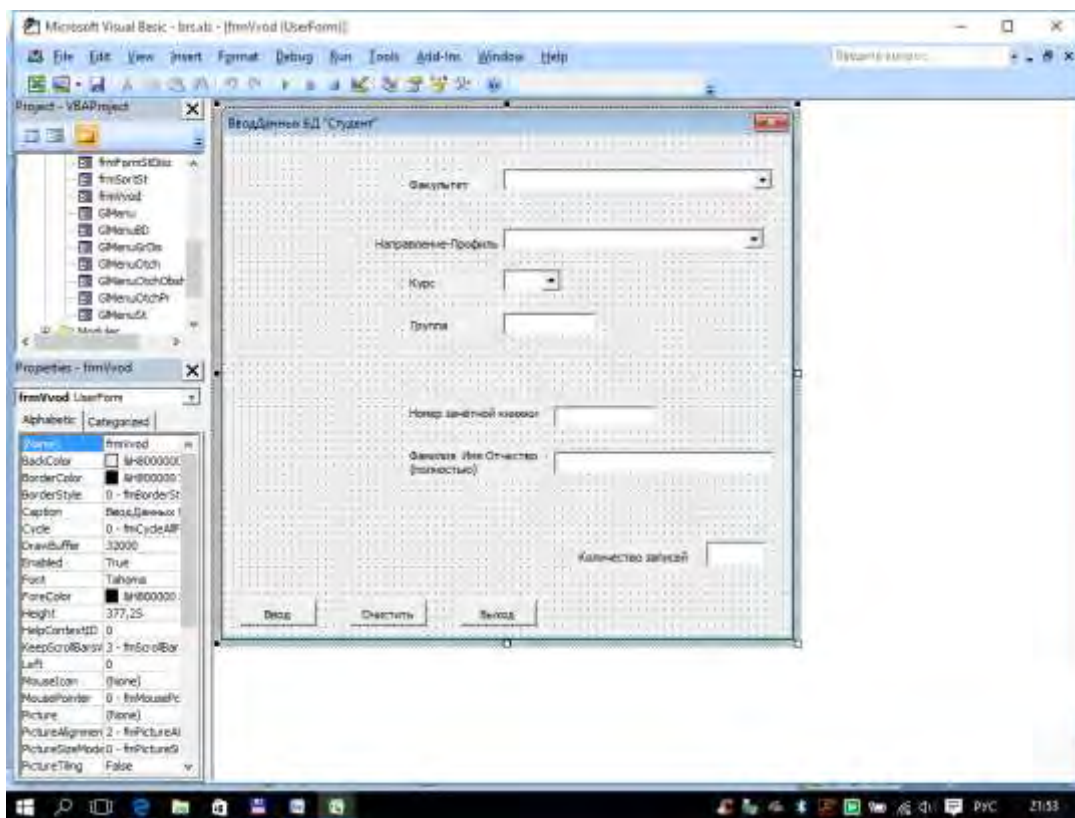


Рисунок 8. Диалоговое окно ВводДанных БД "Студент"

Теперь приступим к написанию кода проекта, который однозначно хранится в виде модулей книги. Модули делятся на общие модули книги (Modules), модули класса (Class Modules), модули форм, которые содержат в себе коды процедур обработки как самой формы (активации, деактивации, инициализации, закрытии, удалении из памяти и т. д.), так и расположенного на форме набора элементов управления (программирование отображения текста, работы переключателей, содержимого окон и списков, графических окон в качестве контейнера, загрузки картинки, работы кнопок и т. д.).

Написание кода процедур будем выполнять по принципу «сверху – вниз»; классического метода структурного программирования: от высокого уровня компонентов к более низкому. Этот принцип намного проще, при этом проще разбираться в программах, редактировать их или применять их в своих задачах. Но алгоритм «сверху – вниз» предполагает необходимость учитывать объекты нижнего уровня, которые и формируют очень часто основу проекта. Определяя функциональность всех без исключения объектов проекта в соответствующих

процедурах модулей, получаем автоматизированный проект, решающий задачи, в нашем случае – бально-рейтинговой системы.

Рассмотрим код модуля формы *frmVvod* (рис. 5) с соответствующими комментариями (листинг 1), которые помогут самостоятельно приобрести навыки практического программирования на VBA.

В модуле работают шесть процедур обработки событий объектов формы: *UserForm_Initialize()*, *UserForm_Activate()*, *ComboBox2_Enter()*, *CmdVvod_Click()*, *CommandButton1_Click()*, *CommandButton2_Click()*. Коды модуля показаны в листинге 5.

```
' переменные объявляются в начальной области _  
модуля – доступны на время работы модуля формы  
' объявление переменной резко упрощает отслеживание _  
переменных и обнаружение ошибок в коде  
Dim i As Double  
' переменная i – для хранения в памяти числа двойной точности  
Dim wsSh As Worksheet  
' объект wsSh – для хранения экземпляра класса РабочийЛист  
Dim li As String  
' переменная li – для хранения текста длиной до 2 млрд символов  
' -----  
Private Sub UserForm_Initialize()  
' инициализация формы UserForm  
' событие инициализации используется для подготовки к работе приложения или  
' формы UserForm до загрузки формы и ее отображения – переменным  
' присваиваются исходные значения;  
' элементам управления КомбинированныйСписок (ПолеСоСписком) присваиваются  
' значения названий факультетов, направлений – профилей, курсов;  
' также могут быть изменены свойства и другие события данной формы  
ComboBox1.AddItem "физико-математического и естественно-_  
научного образования (ФФМЕНО)"  
ComboBox1.AddItem "историко-филологический (ИФФ)"
```

```
ComboBox1.AddItem "механико-технологический (МТФ)"
ComboBox1.AddItem "дошкольного и начального образования (ФДНО)"
ComboBox1.AddItem "педагогике и психологии (ФПП)"
ComboBox1.AddItem "экономический (ЭФ)"
ComboBox1.AddItem "заочный (ЗФ)"
ComboBox1.AddItem "повышения квалификации и переподготовки_
дипломированных специалистов (ФПКПДС)"
ComboBox1.AddItem "дополнительных профессий (ФДП)"
ComboBox2.Clear
ComboBox3.AddItem 1
ComboBox3.AddItem 2
ComboBox3.AddItem 3
ComboBox3.AddItem 4
ComboBox3.AddItem 5
ComboBox3.AddItem 6
End Sub
```

```
' -----
Private Sub UserForm_Activate() ' активация формы UserForm
' событие Activate происходит, когда объект форма становится_
' активным окном (Show – форма становится видимой на_
' экране и в результате получения фокуса);
' события генерируются только при переключении между_
' окнами данного приложения – в другом приложении эти_
' события не будут сгенерированы;
' при активации формы происходит подсчёт и вывод в_
' текстовом окне количества записей базы данных
i = 1
Set wsSh = Sheets("БазаДанных")
' созд. объектную перемен. wsSh как экземпляр класса РабочийЛист
Do While wsSh.Cells(i, 1) > " "
' пока строки непустые подсчит. количество таких строк
i = i + 1
```

```

' элементами объекта РабочийЛист (базы данных) явл. его ячейки
Loop                                ' конец тела цикла
frmVvod.txtN.Enabled = True
' разрешили изменять ТекстовоеПоле txtN
frmVvod.txtN.Text = CStr(i - 2)
' в поле txtN записали количество записей БД "БазаДанных"
frmVvod.txtN.Enabled = False
' запретили изменять ТекстовоеПоле txtN
End Sub
' -----
' следующие коды процедур данного модуля выполняют функции управления
' в некоторый момент времени активного окна ВводДанных БД "Студент" –
' обработка событий нажатий кнопок, выбора элементов списка, _
' ввода текста в текстовые поля и т.д.
' код процедуры обработки события – активация элемента ПолеСоСписком
Private Sub ComboBox2_Enter() ' активация объекта ComboBox2
ComboBox2.Clear
' перед использованием элемент ПолеСоСписком (Направление – Профиль) пустой
li = ComboBox1.Text
' запомнили выбранное нами ранее знач. ПолеСоСписком (название факультета)
If (li = "физико-математического и естественно-научного _
образование (ФФМЕНО)") Then
' добавляем в ПолеСоСписком (Направление – Профиль), если _
факультет физико-математического и естественно-научного образования (ФФМЕНО)
ComboBox2.AddItem "Физика:"
ComboBox2.AddItem "бак_050100_Пед.обр.-Информатика и ИКТ:"
ComboBox2.AddItem "бак_050100_Пед.обр.-Математика:"
ComboBox2.AddItem "Технология:"
End If
If (li = "историко-филологический (ИФФ)") Then
' аналогично добавляем в ПолеСоСписком (Направлени-Профиль)_
по остальным факультетам
ComboBox2.AddItem "g"

```



```

ComboBox2.AddItem "h"
ComboBox2.AddItem "j"
ComboBox2.AddItem "k"
End If
End Sub
' -----
' Код процедуры обработки события – активация стандартного_
' элемента управления
Private Sub CmdVvod_Click() ' УправляющаяКнопка Ввод
Sheets("БазаДанных").Activate
' Активация – метод объекта РабочийЛист "БазаДанных"
Cells(i, 3) = ComboBox1.Text
' Выборка элемента поля КомбинированныйСписок и
Cells(i, 4) = ComboBox2.Text
' заполнение записей базы данных "БазаДанных" полями
Cells(i, 5) = ComboBox3.Text
' соответственно: наименование факультета,_
' направления – профиля, номера курса
Cells(i, 6) = TextBox3.Text
' Выборка элемента поля ТекстовоеПоле и заполнение записей
Cells(i, 1) = TextBox2.Text
' базы данных "БазаДанных" полями соответственно:
Cells(i, 2) = TextBox1.Text
' номера группы, ФИО и номера зачётной книжки
txtN.Enabled = True
txtN.Text = CStr(i - 1)
' И изменили информацию о количестве записей в базе данных
txtN.Enabled = False
i = i + 1
' Увеличили количество записей на 1 при добавлении нового студента
End Sub
' -----

```

```

' Код процедуры обработки события – активация стандартного_
  элемента управления
Private Sub CommandButton1_Click()
' УправляющаяКнопка Очистить
' Присвоение пустых строк полям элементов управления
' КомбинированныйСписок и ТекстовоеПоле формы frmVvod
TextBox1.Text = "": TextBox2.Text = "": TextBox3.Text = ""
ComboBox1.Text = "": ComboBox2.Text = "": ComboBox3.Text = ""
End Sub
' -----
' Код процедуры обработки события – активация стандартного
' элемента управления
Private Sub CommandButton2_Click()
' УправляющаяКнопка Выход
frmVvod.Hide
' скрывание формы – встроенный метод объекта форма frmVvod
End Sub

```

Листинг 5. Текст модуля формы frmVvod

Заполненная форма диалогового окна «Ввод данных о студентах» представлена на рисунке 9.

The image shows a dialog box titled "ВводДанных БД 'Студент'" (Student Database Data Entry). It contains several input fields and buttons. The fields are: "Факультет" (Faculty) with a dropdown menu showing "физико-математического и естественно-нау"; "Направление-Профиль" (Specialization) with a dropdown menu showing "Физика"; "Курс" (Course) with a dropdown menu showing "1"; "Группа" (Group) with a text box containing "1"; "Номер зачётной книжки" (Record Book Number) with a text box containing "12"; "Фамилия Имя Отчество (полностью)" (Full Name) with a text box containing "Иванов Иван Иванович"; and "Количество записей" (Number of Records) with a text box containing "43". At the bottom, there are three buttons: "Ввод" (Input), "Очистить" (Clear), and "Выход" (Exit).

Рисунок 9. Диалоговое окно с заполненными данными о студенте

Приведем пример формы диалогового окна «Ввод данных о студентах с примерами кодов по созданию базы данных факультетов». Пример работы элемента формы показан на рисунке 10.

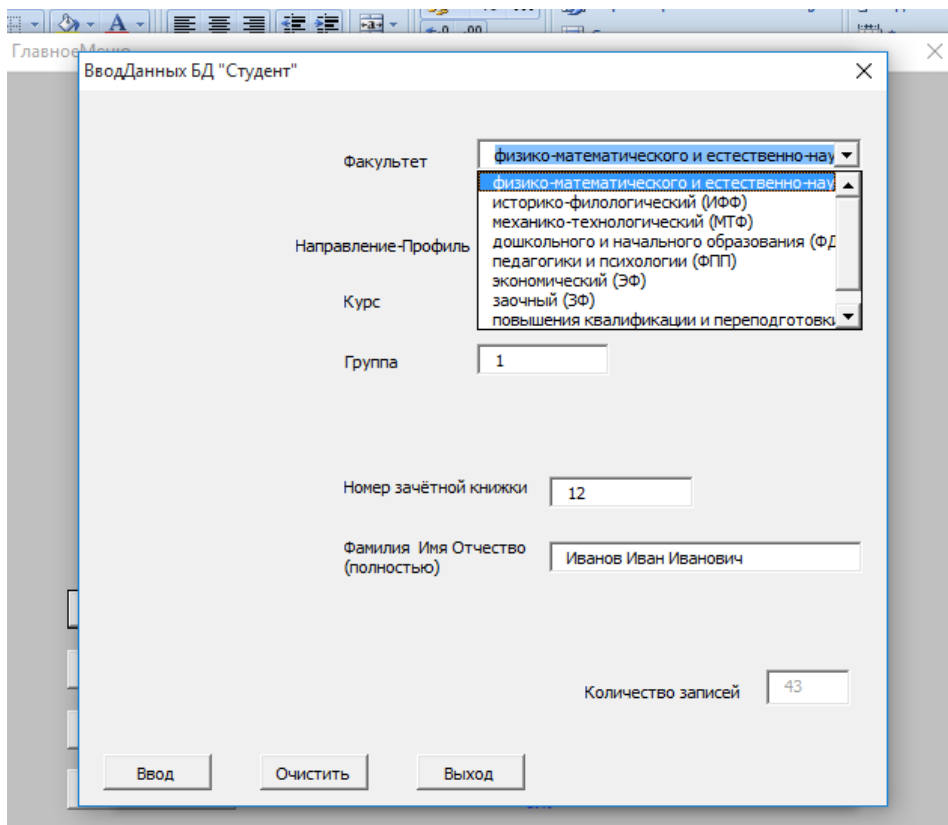


Рисунок 10. Элемент формы «Факультет» в момент активации

Коды по организации работы данного элемента формы показаны в листинге 6.

```
Dim i As Double
```

```
Dim wsSh As Worksheet
```

```
Dim li As String
```

```
Private Sub UserForm_Initialize()
```

```
    ComboBox1.AddItem "физико-математического и _  
                        естественно-научного образования (ФФМЭНО)"
```

```
    ComboBox1.AddItem "историко-филологический (ИФФ)"
```

```
    ComboBox1.AddItem "механико-технологический (МТФ)"
```

```
    ComboBox1.AddItem "дошкольного и начального _  
                        образования (ФДНО)"
```

```
    ComboBox1.AddItem "педагогика и психологии (ФПП)"
```

```
    ComboBox1.AddItem "экономический (ЭФ)"
```

```
    ComboBox1.AddItem "заочный (ЗФ)"
```

```
    ComboBox1.AddItem "повышения квалификации и _
```

```
        переподготовки
```

```
        дипломированных
```

```
        специалистов
```

```
(ФПКПДС)"
```

```
ComboBox1.AddItem "дополнительных профессий (ФДП)"
ComboBox2.Clear
ComboBox3.AddItem 1
ComboBox3.AddItem 2
ComboBox3.AddItem 3
ComboBox3.AddItem 4
ComboBox3.AddItem 5
ComboBox3.AddItem 6
```

End Sub

```
Private Sub UserForm_Activate()
```

```
    i = 1
```

```
    Set wsSh = Sheets("БазаДанных")
```

```
    Do While wsSh.Cells(i, 1) > " "
```

```
        i = i + 1
```

```
    Loop
```

```
    frmVvod.txtN.Enabled = True
```

```
    frmVvod.txtN.Text = CStr(i - 2)
```

```
    frmVvod.txtN.Enabled = False
```

End Sub

```
Private Sub ComboBox2_Enter()
```

```
    ComboBox2.Clear
```

```
    li = ComboBox1.Text
```

```
    If (li = "физико-математического и естественно-научного_
        образования (ФФМЕНО)") Then
```

```
        ComboBox2.AddItem "Физика:"
```

```
        ComboBox2.AddItem "бак_050100_Пед.обр.-Информатика и ИКТ:"
```

```
        ComboBox2.AddItem "бак_050100_Пед.обр.-Математика:"
```

```
        ComboBox2.AddItem "Технология:"
```

```
    End If
```

```
    If (li = "историко-филологический (ИФФ)") Then
```

```
        ComboBox2.AddItem "g"
```

```
        ComboBox2.AddItem "h"
```

```

        ComboBox2.AddItem "j"
        ComboBox2.AddItem "k"
    End If
End Sub
Private Sub CmdVvod_Click()
    Sheets("БазаДАННЫХ").Activate
    Cells(i, 3) = ComboBox1.Text
    Cells(i, 4) = ComboBox2.Text
    Cells(i, 5) = ComboBox3.Text
    Cells(i, 6) = TextBox3.Text
    Cells(i, 1) = TextBox2.Text
    Cells(i, 2) = TextBox1.Text
    txtN.Enabled = True
    txtN.Text = CStr(i - 1)
    txtN.Enabled = False
    i = i + 1
End Sub
Private Sub CommandButton1_Click()
    TextBox1.Text = "": TextBox2.Text = "": TextBox3.Text = ""
    ComboBox1.Text = "": ComboBox2.Text = "": ComboBox3.Text = ""
End Sub
Private Sub CommandButton2_Click()
    frmVvod.Hide
End Sub

```

Листинг 6. Код элемента формы «Факультет»

Образец заполнения кодами проекта в VBA показан на рисунках 11 и 12.

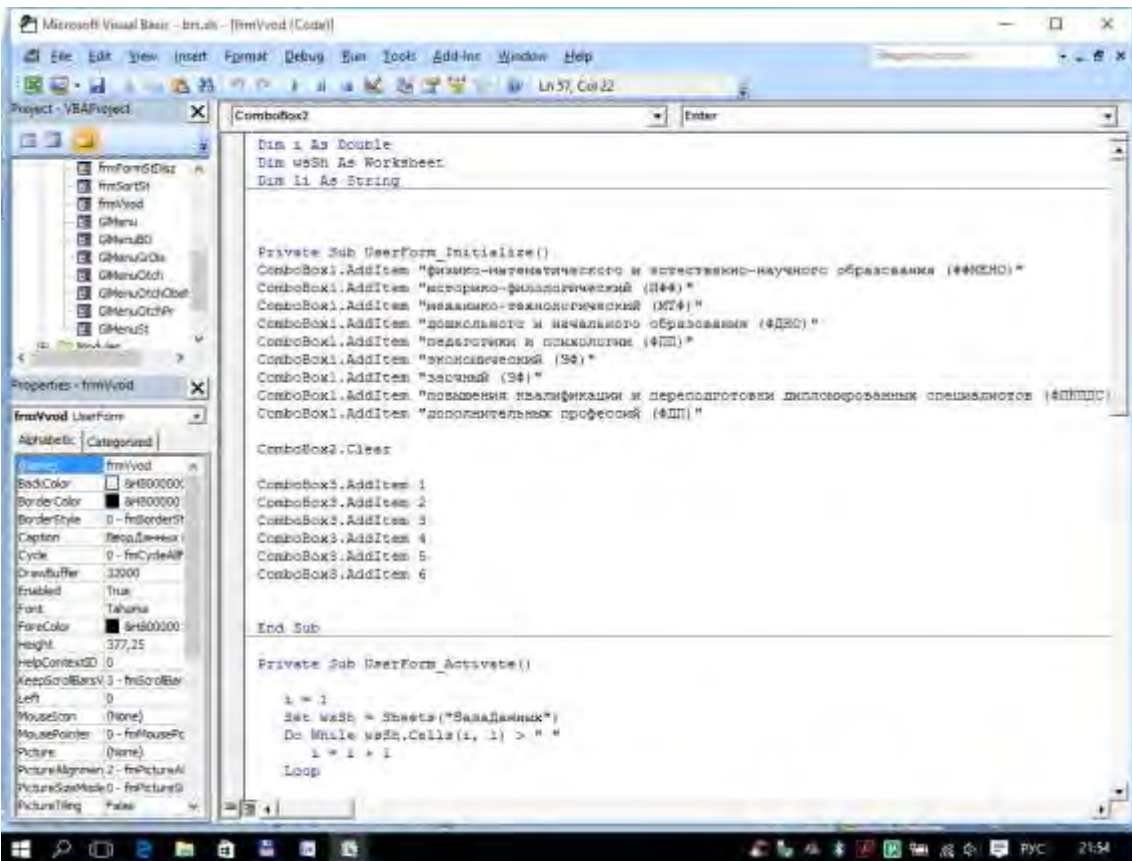


Рисунок 11. Образец написания кода проекта

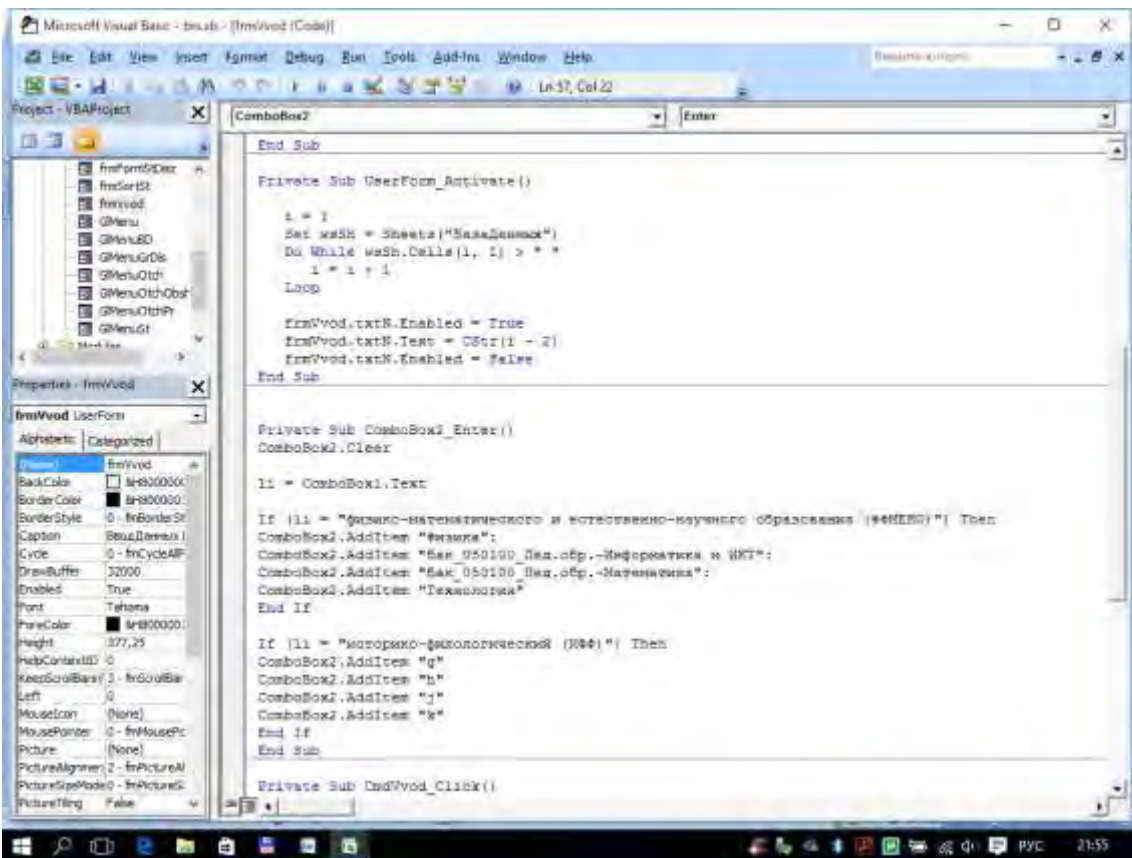


Рисунок 12. Образец написания кода проекта

Задания для самостоятельной работы

1. Перечислите элементы диалогового окна «Базы данных «Студенты».
2. Какие объекты формы были созданы, перечислите их и назовите присущие им свойства\методы.
3. Приведите пример реализации цикла в программе. Какой это цикл, в чем суть его работы.

Контрольные вопросы

1. Для решения каких задач используют объектно-ориентированный подход?
2. Назовите фундаментальные характеристики объектно-ориентированного мышления.
3. Назовите принципы объектной модели. Какие преимущества дает объектная модель?
4. В чем суть инкапсуляции и в чем заключаются ее преимущества?
5. В чем суть наследования и в чем заключается его важность?
6. В чем суть полиморфизма и для чего он полезен?
7. Объясните понятие «агрегирование объекта».
8. Перечислите этапы процесса построения объектной модели.
9. Объясните способы взаимодействия объектов между собой в программе.
10. Назовите процедуры перестройки схемы наследования классов.
11. В чем важность анализа функционирования системы?

Отчетность по работе

Оформите отчет в текстовом редакторе и включите в него следующее:

- Результат выполнения задания 1 в виде готового проекта.
- Ответы на контрольные вопросы.
- Результат выполнения самостоятельных заданий.

1.3.5. Создание элементов базы данных объектно-ориентированного приложения в VBA (4 ч.)

Цель: научиться создавать объекты обработки элементов базы данных объектно-ориентированного приложения в VBA.

Задачи: научиться работать с модулями формы, реализовывать сортировку элементов базы данных в приложении VBA, выполнить процесс создания полноценного приложения с базой данных.

Содержание лабораторной работы

Теоретическая часть

Эта лабораторная работа является логическим продолжением предыдущей лабораторной работы. Здесь студентам предлагается проявить самостоятельность в выполнении практических заданий. То есть теоретическая часть для выполнения заданий может быть использована из предыдущей лабораторной работы.

Практическая часть

Задание 1. На рисунке 13 показаны элементы диалогового окна базы данных «Студенты». Реализуйте элемент формы «Сортировка» данного диалогового окна.

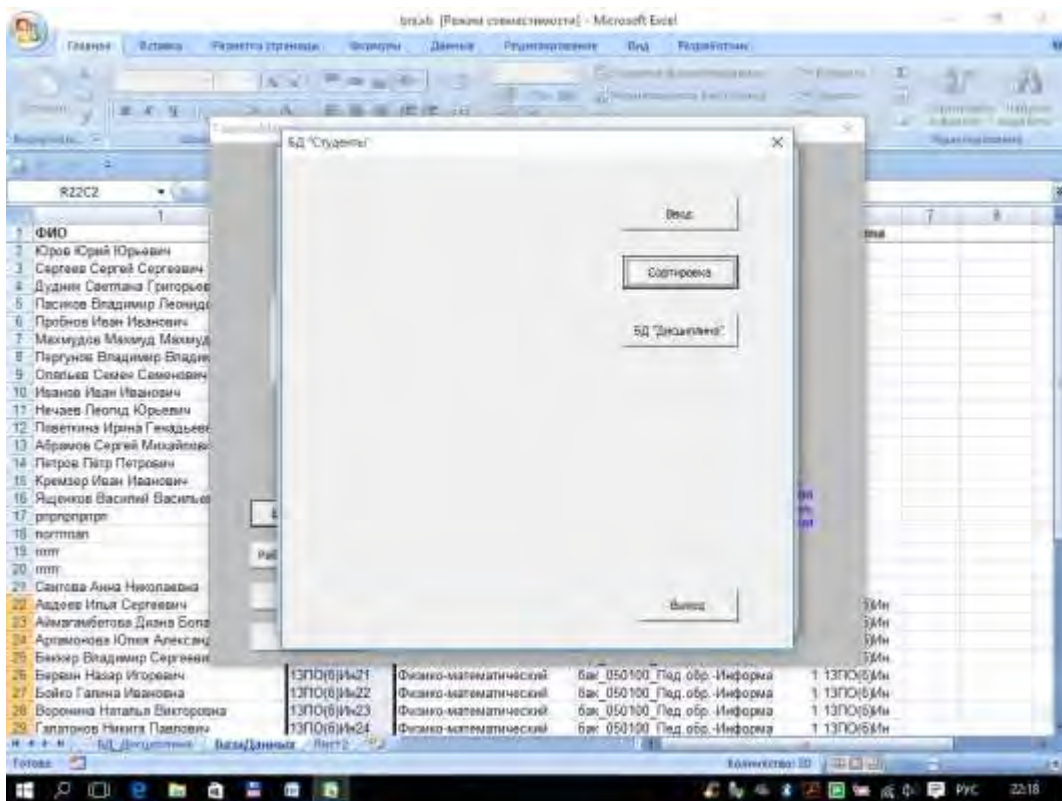


Рисунок 13. Диалоговое окно с элементом «Сортировка»

Примерный результат активации кнопки «Сортировка» показан на рисунке 14.

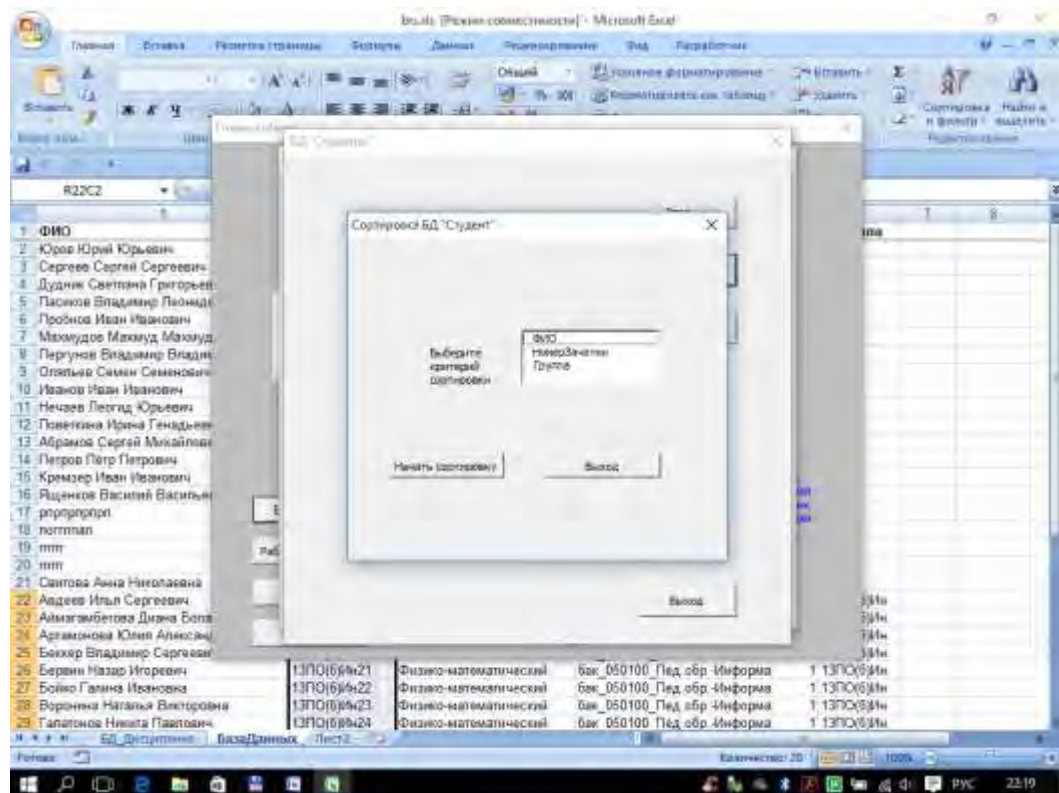


Рисунок 14. Диалоговое окно «Сортировка»

15. Фрагмент реализации в окне редактора VBA показан на рисунке

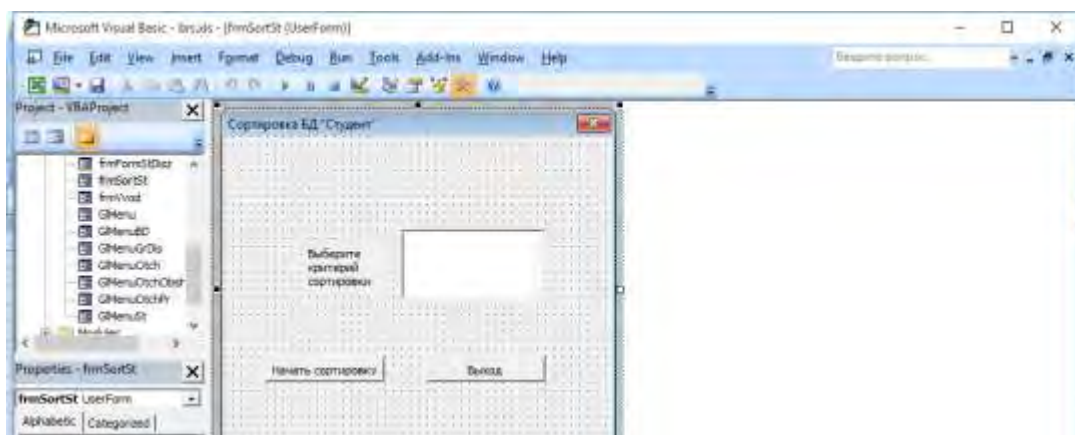


Рисунок 15. Реализация элемента «Сортировка» в окне редактора

Пример кодов по реализации элемента «Сортировка» показан на листинге 7 программы.

```
Private Sub CommandButton2_Click()  
    frmSortSt.Hide  
End Sub  
Private Sub CommandButton3_Click()  
End Sub  
Private Sub UserForm_Initialize()  
    ListBox1.AddItem "ФИО"  
    ListBox1.AddItem "НомерЗачетки"  
    ListBox1.AddItem "Группа"  
End Sub
```

Листинг 7. Код реализации элемента «Сортировка»

Задания для самостоятельной работы

Проанализируйте содержание школьных учебников, задачников, практикумов по информатике на предмет наличия лабораторных работ по работе с формами, объектами, классами, модулями.

Предложите содержание практической работы по изученным темам для факультативного занятия или для углубленного изучения приложений MS Office учащимися.

Сформулируйте основные понятия, которые были изучены в данной лабораторной работе.

Контрольные вопросы

1. Какие модули, классы, объекты были вами реализованы в приложении.
2. Перечислите их методы, свойства. Приведите примеры.
3. Опишите преимущества ООП в сравнении с классическим программированием.
4. Сформулируйте понятие полиморфизма. В чем проявляется полиморфизм?
5. Объясните, что такое событие и приведите пример.

Отчетность по работе

Оформите отчет в текстовом редакторе и включите в него следующее:

1. Результат выполнения заданий лабораторной работы, с описанием этапов выполнения задания.
2. Ответы на контрольные вопросы.
3. Результат выполнения заданий для самостоятельной работы.

1.3.6. Разработка диалоговых окон базы данных

Цель: изучить работу с модулями форм.

Задачи: научиться создавать записи в базе данных и использовать свойства форм.

Содержание лабораторной работы

Теоретическая часть

Код проекта хранится в виде модулей книги. Модули делятся на общие модули книги (Modules), модули класса (Class Modules), модули форм, которые содержат в себе коды процедур обработки как самой формы (активации, деактивации, инициализации, закрытии, удалении из памяти и т. д.), так и расположенного на форме набора элементов управления (программирование отображе-

ния текста, работы переключателей, содержимого окон и списков, графических окон в качестве контейнера, загрузки картинки, работы кнопок и т.д.).

Практическая часть

Задание 1. Выполните шаги, указанные на рисунке 16, с помощью диалогового окна Создание БД "Группа – Дисциплина" (рис. 17), введите наименование дисциплины и количество занятий для данной группы. При этом данные записи должны располагаться в базе данных "БД_Дисциплина" вместе с существующими данными о факультете, группе, студенте и так далее, то есть в ранее заполненной базе данных.

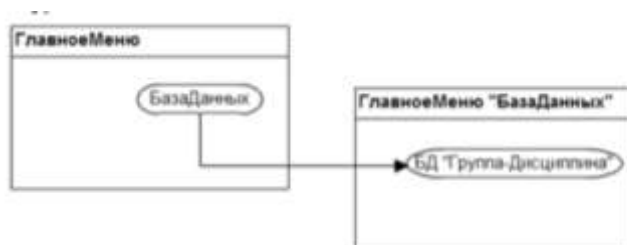


Рисунок 16. Диаграмма реализации диалогового окна

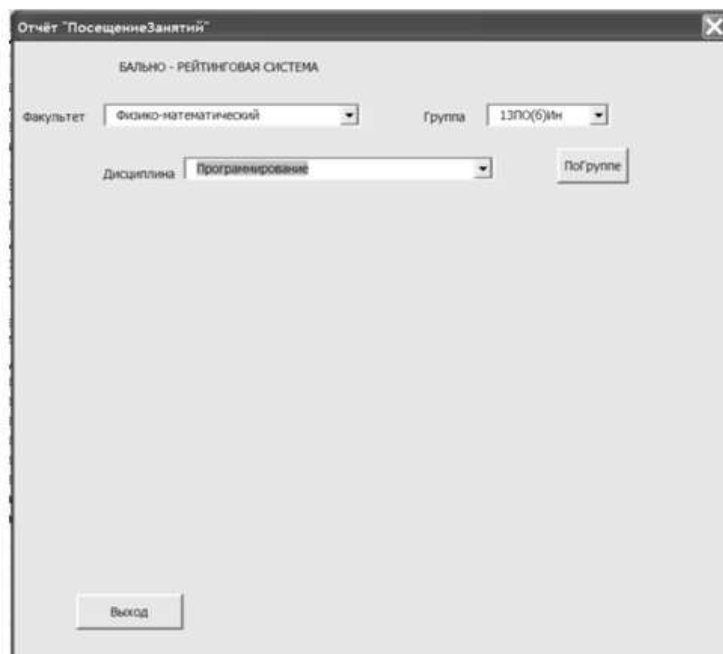


Рисунок 17. Диалоговое окно базы данных «Группа - Дисциплина»

Значения свойств формы базы данных приведены в таблице 5.

Таблица 5

Значения свойств формы ВводДанных БД «Студент»

Объект	Свойство/Метод	Значение
<i>UserForm</i> (Форма)	<i>Name</i> (Имя)	<i>frmFormGrDisz</i>
	<i>Caption</i> (Заголовок)	Создание БД "Группа-Дисциплина"
<i>ComboBox</i> (ПолеСоСписком)	<i>Name</i> (Имя)	<i>ComboBox2</i> <i>ComboBox1</i>
	<i>AddItem</i> (Элемент)	название факультетов название групп
<i>TextBox</i> (ТекстовоеПоле)	<i>Name</i> (Имя)	<i>TextBox4</i>
		<i>TextBox5</i>
		<i>TextBox6</i>
<i>Label</i> (Надпись)	<i>Name</i> (Имя)	<i>Label7</i> , <i>Label5</i> , <i>Label6</i> , <i>Label9</i> , <i>Label8</i>
	<i>Caption</i> (Заголовок)	факультет, группа,
		дисциплина, количество лекций, количество практических
<i>CommandButton</i> (УправляющаяКнопка)	<i>Name</i> (Имя)	<i>CmdVvod</i> , <i>CommandButton1</i> , <i>CommandButton2</i>
	<i>Caption</i> (Заголовок)	Ввод, Очистить, Выход

Рассмотрим код (листинг 8) модуля формы *frmFormGrDisz* (рис. 20), в котором определены шесть процедур: одна процедура с параметрами *InitBoxSort* и процедуры *ComboBox2_Enter()*, *ComboBox1_Enter()*, *CmdVvod_Click()*, *CommandButton1_Click()*, *CommandButton2_Click()*. Модуль выполняет добавление в базу данных новой дисциплины с соответствующим количеством часов лекционных и практических занятий для каждой группы некоторого факультета.

' Код процедуры с параметрами используется для сортировки_ элементов любого объекта *ПолеСоСписком* данной_ формы – параметром процедуры является экземпляр *b* класса_ элемента управления формы

Private Sub InitBoxSort(b As MSForms.ComboBox)

' *InitBoxSort* – процедура с параметрами

```

Dim i As Long, c As Long, s$, l As Long, r As Long
Dim li As String
Sheets("БазаДанных").Activate
li = ComboBox2.Text
    For i = 2 To Cells.Rows.Count
        'заполняем со второй строки и первого столбца
        If Cells(i, 3) = li Then s = Cells(i, 6)
            l = 0
            r = b.ListCount - 1
            Do While r >= 1
                c = (r + 1) \ 2
                Select Case StrComp(s, b.List(c))
                    Case -1: r = c - 1
                    Case 1: l = c + 1
                    Case Else: GoTo NEXTI
                End Select
            Loop
            b.AddItem s, l
        NEXTI:
    Next i
End Sub

```

Листинг 8. Код модуля формы frmFormGrDisz

Задание 2. Создайте приложение по индивидуальному варианту задания.

Таблица 6

Варианты итоговых проектных работ

Вариант	Задание
1	2
1	Необходимо создать программу для расчета расхода топлива в транспортной компании
2	Создать электронный документ «Аттестация студентов группы 14 ПобИН». Разработайте модуль «Работа со списком выбранной группы», в котором нужно предусмотреть ввод фамилий студентов, удаление фамилий из списка и добавление, аналогично тому, как это сделано в примере с названиями студенческих групп

1	2
3	Расположить элементы управления на форме и создать код, обеспечивающий ввод данных в соответствующие листы рабочей книги. Перечислить дисциплины, по которым проводится аттестация. Для каждого предмета применяется однотипный набор элементов, добавить календарь. Вставить кнопки управления. Для отображения всех оценок и среднего балла, ввода оценок, очистки текстовых полей и закрытия формы понадобятся 4 кнопки с именами: Просмотр, Ввод, Очистить и Выход
4	Создать электронный журнал успеваемости ученического класса с выводом на печать информации за четверть (для родителей)
5	Создать программу под названием «Телефонный справочник». Справочник содержит отдельно объекты: «Преподаватели», «Студенты группы 14 ПобИН», «Родители студентов». Предоставить возможность выбора нужной фамилии из списка фамилий, представленного в пользовательской форме. При выборе из списка нужной фамилии на форме должен отразиться соответствующий номер телефона. Пользователь должен иметь возможность добавить фамилии в список справочника и возможность удалить ненужные фамилии из справочника
6	Создать диагностическую программу, в которой пользователю на форме высвечивается вопрос и предлагается выбрать один из трёх предложенных ответов. За каждый ответ присуждается различное количество баллов. Тематику теста продумать самостоятельно
7	Создать программу-опросник, в которой пользователю на форме показывается вопрос и предлагается выбрать один из трёх предложенных ответов из списка. На форме должно выводиться сообщение о том, сколько человек приняли участие в опросе и сколько человек выбрали ответ 1, 2, 3. Тематику опроса выбрать самостоятельно
8	Создать программу диагностики «Уровень воспитанности студентов группы 14 ПОБИН» с вариантами ответов отличных от традиционных тестовых программ
9	Разработать программу-калькулятор, которая будет выполнять арифметические операции (сложение, умножение, вычитание) над комплексными числами, которые вводятся в поля диалогового окна «Калькулятор». Организуйте список операций, которые выполняются над комплексными числами (операция должна выбираться из списка)
10	Составить программу для обработки экзаменационной ведомости, которая позволит подсчитать средний балл оценок экзамена, проведенного в студенческой группе, количественную и качественную успеваемость. На форме предоставьте возможность по фамилии студентов посмотреть его оценку. Создайте в программе обработчик ошибок, который будет проверять, являются ли данные из списка числами
11	Создать приложение «Склад», позволяющее учитывать количество товара на складе. Диалоговое окно приема товара на склад должно позволить

1	2
	ввести наименование товара, цену, количество, дату приема и единицу измерения товара. Поступающие товары должны записываться в базу данных рабочего листа. Обеспечить различные виды сортировки товара. Отчет программа выдает в MS Word
12	Разработать приложение с диалоговым окном “Учащиеся школы № X”. В текстовые поля вводится следующая информация: фамилия, класс, адрес места жительства, номер телефона, номер телефона родителей, пол. В раскрывающемся списке выводятся все возможные виды информации (результаты выполнения пробных ГИА, КИМ, ЕГЭ или другие данные (анкетирования, тестирования), информация о которых хранится в отдельной таблице, там же указана успеваемость). При нажатии кнопки ОК данные из диалогового окна должны отражаться на рабочем листе
13	Разработать приложение, которое позволит построить поверхность. Пользователь вводит уравнение функции двух переменных, начальное и конечное значения аргументов, шаг их изменения. Обеспечить использование полосы прокрутки для возможности изменения ориентации пространственного местоположения поверхности
14	Разработать приложение для вывода листа коммунальных платежей (за основу взять реальную квитанцию на оплату коммунальных платежей)
15	Создать приложение для решения линейных алгебраических уравнений методом Крамера и при помощи обратной матрицы. 1. Разработать диалоговое окно для ввода информации о системе линейных уравнений, содержащее следующие элементы управления: а) ввод количества уравнений и неизвестных (количество уравнений равно количеству неизвестных); б) ввод коэффициентов уравнения; с) выбор метода решения; д) необходимые управляющие кнопки 2. Создать два листа для двух методов решения. На рабочий лист должна выводиться следующая информация: а) начальные данные; б) промежуточная информация; с) решение
16	Создать приложение для регистрации клиентов компьютерного салона, содержащее следующие поля: № компьютера (выбирается из списка номеров свободных компьютеров), фамилия, имя клиента, тип игры (логические, карточные, аркады, квест, 3д-шутер, гонки и т. д.). Список типов должен существовать на отдельном листе, название игры, время начала игры, стоимость 1 часа. Разработать диалоговое окно для заполнения таблицы регистрации клиентов компьютерного салона. Разработать диалоговое окно для заполнения таблицы расчета платы за использованное компьютерное время. Создать бланк счета в Word. Внести данные для конкретного человека

1	2
17	<p>Разработать программу для решения кубических уравнений. Диалоговое окно № 1 создать для ввода коэффициентов уравнения. Диалоговое окно № 2 должно состоять из двух страниц, одна – для создания таблицы данных для построения графика решения и вторая – для вывода графика. Все необходимые расчеты проводить на именованных рабочих листах</p>
18	<p>Разработать программу, которая позволит автоматизировать работы туристической фирмы. Диалоговое окно должно содержать следующие поля: фамилия, имя, отчество, паспортные данные, страну, курорт, тип обслуживания (полный пансион и т. д.), тип гостиницы (*****, ***** и т.д.), дополнительные услуги (питание, экскурсии и т. д.), количество путевок, льготы (детская путевка, постоянный клиент и т. д.), стоимость путевки (в зависимости от типа гостиницы и льгот), итоговая сумма.</p> <p>Должно быть два диалоговых окна. В одном из диалоговых окон должна отображаться информация о наличии свободных путевок для выбранных типов стран и курортов, наличии гостиниц. Программа должна позволить распечатать заполненную таблицу продажи путевок</p>
19	<p>Разработать программу, которая позволит автоматизировать процесс регистрации клиентов в гостинице. Таблица регистрации должна содержать следующие поля: № регистрации, фамилию, имя, отчество, паспортные данные, тип номера (люкс, 1 класс, 2 класс и т. д.), № комнаты – выбирается из списка номеров свободных комнат выбранного типа номера, цена номера за сутки, дополнительные услуги: телевизор, интернет и так далее, дата заселения. Диалог для регистрации клиентов должен состоять из двух диалоговых окон. Второе диалоговое окно – для номера (типы номеров, списки свободных номеров для данного типа и т. д.).</p> <p>Разработать диалоговое окно для заполнения таблицы отъезда, которое информирует о ФИО, дате выезда, необходимые графы для расчета оплаты (в зависимости от цены номера, дополнительных услуг и т. д.). Бланк счета должен быть распечатан в MS Word</p>
20	<p>Разработать приложение, работающее с файлом произвольного доступа. Данный файл предназначен для хранения информации о студентах по фамилии, имени, номеру группы. Диалоговое окно «Информация» позволяет перемещаться от записи к записи, редактировать существующие и создавать новые записи. Переход от записи к записи производится при помощи счетчика. При активизации диалогового окна доступным для пользователя являются только поле со списком и кнопка Открытие файла. В поле со списком отображаются все файлы текущей папки с расширением dat. Пользователь может выбрать файл из этого списка или ввести в поле имя нового файла. После ввода имени файла поле со списком и кнопка Открытие файла становятся недоступными для пользователя, а кнопки Новая запись, Записать изменения, Закрывать, счетчик Номер записи и поля ввода-вывода информации становятся, наоборот, доступными</p>

Задания для самостоятельной работы

Задание 1. Разработайте программу, полезную для будущего учителя информатики, классного руководителя. Это может быть электронный классный журнал успеваемости, журнал учета учащихся, система для расчета рейтинга учащихся и т. д.

Контрольные вопросы

1. Что такое модуль класса?
2. Объясните суть разработк модуля класса.
3. Какие элементы управления изучены вами при выполнении лабораторной работы?

Отчетность по работе

Оформите отчет в текстовом редакторе и включите в него следующее:

1. Результат выполнения задания 2, с описанием этапов выполнения задания.
2. Ответы на контрольные вопросы.
3. Результат выполнения задания 3.

[Вернуться к содержанию](#)

2. Альтернативное программирование

2.1. Программное обеспечение на базе ОС Linux

В нашей стране, как и во многих странах мира, принята государственная программа перехода на «альтернативное» программное обеспечение. Поэтому на сегодняшний день становится эффективным использование операционных систем и приложений, которые являются свободно распространяемыми.

В школьных компьютерных классах в основном установлена операционная система Windows. Установка ОС Windows требует наличия лицензии, которая является платной. Платными являются и остальные программы, необходимые для реализации учебного плана по информатике, включая антивирусные программы. Реальной альтернативой становится использование свободного программного обеспечения на базе ОС Linux (Линукс). Под Линукс обычно понимают общее название Unix-подобных операционных систем, основанных на одноимённом ядре. Ядро Linux создаётся и распространяется по модели свободного и открытого программного обеспечения. Единой комплектации Linux нет. Распространяются в основном бесплатно различные готовые дистрибутивы, имеющие свой набор прикладных программ и уже настроенных под конкретные нужды пользователя. GNU – операционная система типа Unix. Дистрибутивы GNU/Linux состоят из свободных программ.

В настоящее время системы Linux лидируют на рынках смартфонов (Android занимает 85% рынка [9]), интернет-серверов (60%), самых мощных суперкомпьютеров (97%). Дистрибутивы Linux широко используются в качестве операционной системы суперкомпьютеров: по данным на ноябрь 2015 г., 98,8% компьютеров из списка 500 самых мощных работают под управлением различных вариантов Linux [68].

Операционной системой самого мощного современного суперкомпьютера – Tianhe 2 – является Kylin Linux [Operating system Family / Linux (англ.). TOP500. <http://www.top500.org/statistics/details/osfam/1>

Проверено 6 марта 2014. Dongarra, Jack. Visit to the National University for Defense Technology Changsha, China. Netlib (03.06.2013.)

Для поддержки внедрения свободного программного обеспечения (ПО) в учебных заведениях России создан сайт LinuxSchool.ru, а также начал работу учебный портал edu.linuxschool.ru. Эти ресурсы поддерживаются компанией «АЛЬТ-ЛИНУКС» и учебным центром «Навигатор» при активном содействии региональных участников пилотного внедрения свободного ПО в учебных заведениях Российской Федерации и созданы с целью сформировать сообщество специалистов и пользователей, объединенных задачами создания и внедрения в работу школ свободного программного обеспечения.

Перечислим явные преимущества использования GNU/Linux для преподавания школьного курса информатики: свободное распространение GNU/Linux позволяет обойтись без покупки лицензии на операционную систему (ОС), офисный пакет и антивирусные программы; использование GNU/Linux соответствует политике импортозамещения и дистрибутив GNU/Linux (Alt Linux) поставляется в учебные заведения как пакет «Первая помощь 2.0»; высокий уровень безопасности при эксплуатации различного программного обеспечения означает практическую защищенность системы GNU/Linux от вирусных программ, также система контроля учетных записей не позволит случайно повредить ОС неопытному пользователю.

Поэтому актуальным является изучение и использование ОС GNU/Linux и LibreOffice в школьном курсе информатики. Однако анализ содержания школьных учебников по информатике показал, что основной операционной системой является ОС Windows, в связи с этим для многих учителей информатики существенной проблемой является недостаточное методическое обеспечение для изучения ОС GNU/Linux и LibreOffice. Например, преподавание раздела «Объектно-ориентированное программирование на языке Visual Basic» по учебнику Н. Д. Угриновича в старшей школе основано на изучении VBA [2]. VBA является встроенным приложением Excel (который входит в состав в MS Office).

LibreOffice является практически полным аналогом пакета MS Office и может быть установлен бесплатно и на любую ОС (поддерживаются Windows/OS X/Linux). Синтаксис макросов в LibreOffice незначительно отличается от своего аналога в MS Office и подходит для изучения основ объектно-ориентированного программирования.

Наиболее сложным разделом школьного курса информатики является программирование. В связи с этим выполнение студентами заданий по адаптации содержания школьного учебника информатики по изучению VBA под LibreOffice позволит повысить их методическую компетентность.

[Вернуться к содержанию](#)

2.2. Программирование с использованием LibreOffice

Программирование с использованием ОС GNU/Linux и LibreOffice реализуем в виде комплекса заданий, некоторые из которых аналогичны заданиям из школьного учебника информатики.

Задание 1. Создать проект, в котором после запуска программы форма располагается в центре экрана монитора, а четыре кнопки – в центре формы. После щелчка по кнопкам они должны перемещаться в углы формы.

Например, результат выполнения задания по расстановке кнопок по углам показан на рисунке 18 (слева показана реализация задачи в VBA, справа – аналог в LibreOffice).

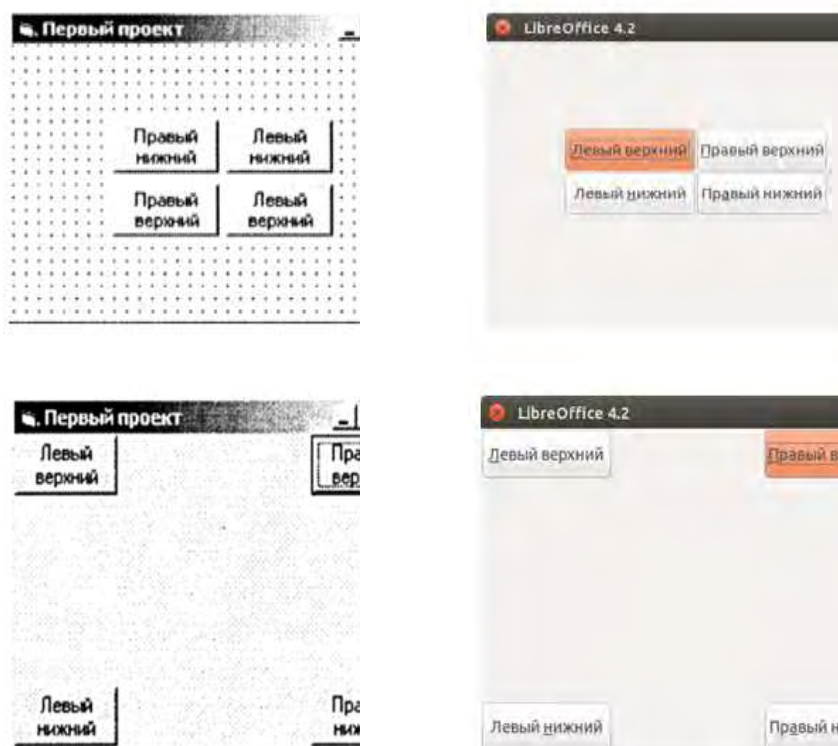


Рисунок 18. Результат решения задачи в VBA и в LibreOffice

Отметим, что операционных отличий в системах не существует, однако отличия есть в интерфейсе, файловой системе и логике взаимодействия с пользователем.

Программный код решения данной задачи по «расстановке кнопок по углам формы» предложен в листинге 10.

```
REM ***** BASIC *****
```

```
Dim Dlg as object
```

```
Sub Window
```

```
    DialogLibraries.LoadLibrary("Standard")
```

```
    Dlg = CreateUnoDialog(DialogLibraries.Standard.Dialog1)
```

```
    Dlg.Execute()
```

```
    Dlg.Dispose()
```

```
End Sub
```

```
Sub TopLeft
```

```
    DialogLibraries.LoadLibrary("Standard")
```

```
    Dlg.getControl("cmd1").model.PositionX = 0
```

```
    Dlg.getControl("cmd1").model.PositionY = 0
```

```
    Dlg.execute()
```

```
End Sub
```

```
Sub TopRight
```

```
    DialogLibraries.LoadLibrary("Standard")
```

```
    Dlg.getControl("cmd2").model.PositionX = 99
```

```
    Dlg.getControl("cmd2").model.PositionY = 0
```

```
    Dlg.execute()
```

```
End Sub
```

```
Sub BottomRight
```

```
    DialogLibraries.LoadLibrary("Standard")
```

```
    Dlg.getControl("cmd3").model.PositionX = 99
```

```
    Dlg.getControl("cmd3").model.PositionY = 109
```

```
    Dlg.execute()
```

```
End Sub
```


Sub BottomLeft

DialogLibraries.LoadLibrary("Standard")

Dlg.getControl("cmd4").model.PositionX = 0

Dlg.getControl("cmd4").model.PositionY = 109

Dlg.execute()

End Sub

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE dlg:window PUBLIC "-//OpenOffice.org//DTD_
OfficeDocument 1.0//EN" "dialog.dtd">
```

```
<dlg:window          xmlns:dlg=http://openoffice.org/2000/dialog
```

```
xmlns:script="http://openoffice.org/2000/script"          dlg:id="Dialog1" _
```

```
dlg:left="106"      dlg:top="68"      dlg:width="145"      dlg:height="129" _
```

```
dlg:closeable="true" dlg:moveable="true">
```

```
  <dlg:bulletinboard>
```

```
    <dlg:button  dlg:id="cmd1"  dlg:tab-index="0"  dlg:left="27" _
```

```
dlg:top="39" dlg:width="46" dlg:height="20" dlg:value="ЛЕВЫЙ верхний">
```

```
  <script:event script:event-name="on-performaction" script:_
macroname="vnd.sun.star.script:Standard.Module1.TopLeft?_
language=Basic&location=document" script:language="Script"/>
```

```
  </dlg:button>
```

```
  <dlg:button dlg:id="cmd2" dlg:tab-index="1" dlg:left="73" _
```

```
dlg:top="39" dlg:width="46" dlg:height="20" dlg:value="Правый верхний">
```

```
  <script:event script:event-name="on-performaction" script:_
macroname="vnd.sun.star.script:Standard.Module1.TopRight?_
language=Basic&location=document" script:language="Script"/>
```

```
  </dlg:button>
```

```
  <dlg:button dlg:id="cmd3" dlg:tab-index="3" dlg:left="73" _
```

```
dlg:top="59" dlg:width="46" dlg:height="20" dlg:value="Правый нижний">
```

```

<script:event script:event-name="on-performaction" _
script:macroname="vnd.sun.star.script:Standard.Module1 _
.BottomRight?language=Basic&location=document" _
script:language="Script"/>
</dlg:button>
<dlg:button dlg:id="cmd4" dlg:tab-index="2" dlg:left="27" _
dlg:top="59" dlg:width="46" dlg:height="20" dlg: value="Левый нижний">
  <script:event script:event-name="on-performaction" script:_
macroname="vnd.sun.star.script:Standard.Module1.BottomLeft?_
language=Basic&location=document" script:language="Script"/>
</dlg:button>
</dlg:bulletinboard>
</dlg>window>

```

Листинг 10. Программный код расстановки кнопок по углам формы

Результат работы программного кода в окне LibreOffice представлен на рисунке 19.



Рисунок 19. Окно LibreOffice с результатом работы программного кода

Задание 2. Создать массив случайных чисел и найти минимальный элемент массива.

Программный код для организации работы с массивом приведен в листинге 11.

```
Dim bytI ,bytMin, bytJ, bytK, bytR, bytN , bytA(1 to 10) As
Byte
Dim Dlg as object
Sub Window
    DialogLibraries.LoadLibrary("Standard")
    Dlg = CreateUnoDialog(DialogLibraries.Standard.Dialog1)
    Dlg.Execute()
End Sub
Sub Grnt
    Dim sic as String
    Randomize
    For bytI = 1 to 10
        bytA(bytI) = Int(Rnd * 10)
        sic = sic + Str(bytA(bytI))
    Next bytI
    Dlg.getControl("txt1").model.Text = sic
End Sub
Sub Minimal
    bytMin = bytA(1)
    bytN = 1
    For bytI = 2 To 10
        If bytA(bytI) < bytMin Then bytMin = bytA(bytI): bytN = bytI
    Next bytI
    MsgBox ("Минимальный элемент равен " + bytMin + ". Его индекс " + bytN + ".")
End Sub
Sub MinElemnt
    bytMin = bytA(bytI)
    bytN = bytI
    For bytJ = bytI + 0 To 10
        If bytA(bytJ) < bytMin Then bytMin = bytA(bytJ) : bytN = bytJ
    Next bytJ
```

End Sub

Sub Sort

```
vbCrLf = Chr (13) + Chr (10)
```

```
Dlg.getControl("txt2").model.Text = ""
```

```
For bytI = 1 To 9
```

```
Call MinElemnt(bytI, bytN)
```

```
bytR = bytA(bytI)
```

```
bytA(bytI) = bytA(bytN)
```

```
bytA(bytN) = bytR
```

```
For bytK = 1 To 10
```

```
Dlg.getControl("txt2").model.Text =
```

```
Dlg.getControl("txt2").model.Text + Str(bytA(bytK))
```

```
Next bytK
```

```
Next bytI
```

End Sub

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE dlg:window PUBLIC "-//OpenOffice.org_
```

```
//DTD OfficeDocument 1.0//EN" "dialog.dtd">
```

```
<dlg:window xmlns:dlg=http://openoffice.org/2000/dialog\_
```

```
xmlns:script="http://openoffice.org/2000/script" dlg:id="Dialog1" _
```

```
dlg:left="108" dlg:top="77" dlg:width="81" dlg:height="184" _
```

```
dlg:closeable="true" dlg:moveable="true">
```

```
<dlg:bulletinboard>
```

```
<dlg:button dlg:id="cmd1" dlg:tab-index="0" dlg:left="1" _
```

```
dlg:top="22" dlg:width="57" dlg:height="18" dlg:_
```

```
value="Сгенерировать" dlg:multiline="true">
```

```
<script:event script:event-name="on-performaction" _
```

```
script:macroname="vnd.sun.star.script:Standard.Module1.Gnrt? _
```

```
language=Basic&location=document" script:language="Script"/>
```

```
</dlg:button>
```

```
<dlg:textfield dlg:id="txt1" dlg:tab-index="1" dlg:left="1" _
```

```

dlg:top="4"    dlg:width="77"    dlg:height="14"    dlg:align="center" _
dlg:readonly="true" dlg:value="Тут будет появляться массив"/>
    <dlg:button    dlg:id="cmd2"    dlg:tab-index="2"    dlg:left="63"
dlg:top="22" dlg:width="16" dlg:height="18" dlg:value="Min"> _
    <script:event script:event-name="on-performaction" script:_
        macroname="vnd.sun.star.script:Standard.Module1.Minimal?_
        language=Basic&location=document" script:language="Script"/>
    </dlg:button>
    <dlg:fixedline dlg:id="FixedLine1" dlg:tab-index="3" dlg:left="1" _
        dlg:top="43" dlg:width="79" dlg:height="4"/>
    <dlg:textfield    dlg:id="txt2"    dlg:tab-index="4"    dlg:left="6" _
dlg:top="52"    dlg:width="44"    dlg:height="107"    dlg:page="10" _
dlg:maxlength="10" dlg:multiline="true" dlg:readonly="true"/>
    <dlg:button    dlg:id="cmd3"    dlg:tab-index="5"    dlg:left="8" _
dlg:top="164" dlg:width="66" dlg:height="18" dlg:value="Сортировать" dlg:multiline="true">
    <script:event script:event-name="on-performaction" _
        script:macroname="vnd.sun.star.script:Standard.Module1.Sort?_
        language=Basic&location=document" script:language="Script"/>
    </dlg:button>
</dlg:bulletinboard>
</dlg>window>

```

Листинг 11. Программный код «Работа с массивом»

Результат работы программного кода показан на рисунке 20.

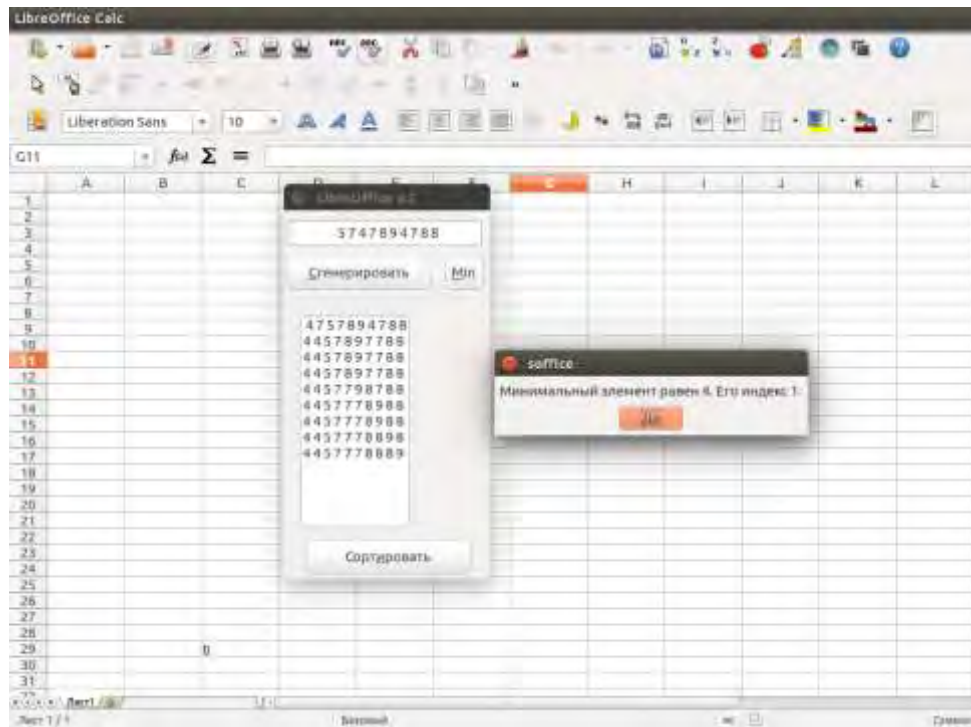


Рисунок 20. Результат выполнения программного кода «Работа с массивом»

Задание 3. Создайте программу для учета финансов. Программа позволяет ввести доход пользователя, произведенные расходы и рассчитать баланс денежных средств.

Программный код программы «Учет финансов» предложен в листинге 12.

```

REM ***** BASIC *****
Dim Dlg1, Dlg2, Dlg3, Dlg4, Sheet, Doc as object
Dim id, z as Integer
Sub Window1
    DialogLibraries.LoadLibrary("Standard")
    Doc = StarDesktop.CurrentComponent
    Sheet = Doc.Sheets(0)
    Dlg1 = CreateUnoDialog(DialogLibraries.Standard.Dialog1)
    Dlg1.Execute()
End Sub
Sub Dlg1Off
    Dlg1.endExecute
End Sub

```

Sub Window2

DialogLibraries.LoadLibrary("Standard")

Dlg2 = CreateUnoDialog(DialogLibraries.Standard.Dialog2)

id = 2

While Sheet.getCellByPosition(0,id).String <> ""

id = id+1

Wend

Dlg2.getControl("LabelDate").model.Label = Date

Dlg2.getControl("LabelId").model.Label = id - 3

Dlg2.Execute()

End Sub

Sub Window3

DialogLibraries.LoadLibrary("Standard")

Dlg3 = CreateUnoDialog(DialogLibraries.Standard.Dialog3)

Dlg3.Execute()

End Sub

Sub Window4

DialogLibraries.LoadLibrary("Standard")

Dlg4 = CreateUnoDialog(DialogLibraries.Standard.Dialog4)

Dohod = 0

Rashod = 0

id = 2

While Sheet.getCellByPosition(0,id).String <> ""

id = id+1

Wend

For z = 0 to id

If Sheet.getCellByPosition(2,z).String = "Доход" Then

Dohod = Dohod + Sheet.getCellByPosition(3,z).Value

ElseIf Sheet.getCellByPosition(2,z).String = "Расход" Then

Rashod = Rashod + Sheet.getCellByPosition(3,z).Value

End If

Next z

```
Dlg4.getControl("Label3").model.Label = Dohod
Dlg4.getControl("Label5").model.Label = Rashod
Raznica = Dohod - Rashod
```

```
    If Raznica < 0 then
```

```
        Raznica = - Raznica
```

```
    End If
```

```
Dlg4.getControl("Label7").model.Label = Raznica
```

```
Dlg4.Execute()
```

```
End Sub
```

```
Rem INPUT-----
```

```
Sub WriteToTable
```

```
    Sheet.getCellByPosition(0,id).Value = id - 3
```

```
    Sheet.getCellByPosition(1,id).String =
```

```
    Dlg2.getControl("LabelDate").model.Label
```

```
    Sheet.getCellByPosition(2,id).String =
```

```
    Dlg2.getControl("ComboBoxType").model.Text
```

```
    Sheet.getCellByPosition(3,id).Value =
```

```
    Dlg2.getControl("NumericFieldSumma").model.Value
```

```
    Sheet.getCellByPosition(4,id).String =
```

```
    Dlg2.getControl("TextFieldComment").model.Text
```

```
    Dlg2.EndExecute
```

```
rem Save -----
```

```
    dim document as object
```

```
    dim dispatcher as object
```

```
    document = ThisComponent.CurrentController.Frame
```

```
    dispatcher = createUnoService("com.sun.star.frame.DispatchHelper")
```

```
    dispatcher.executeDispatch(document, ".uno:Save", "", 0, Array())
```

```
End Sub
```

```
Sub Dlg2Off
```

```
    Dlg2.EndExecute
```

```
End Sub
```

```
Rem OUTPUT-----
```



```

Sub ChooseDate
Proverka = False
Dlg3.getControl("CurrencyField1").model.Enabled = False
Dlg3.getControl("TextField1").model.Enabled = False
id = 3
  While Sheet.getCellByPosition(0,id).String <> ""
    id = id+1
  Wend
  For z = 0 to id
    If Dlg3.getControl("DateField1").model.Text = _
      Sheet.getCellByPosition(1,z).String Then
      Proverka = true
      Exit For
    End If
  Next z
If Dlg3.getControl("DateField1").model.Text = "" Then
  MsgBox("Выберите дату")
ElseIf Proverka = False Then
  MsgBox("В этот день вы ничего не записывали")
ElseIf Proverka = True Then
  Dlg3.getControl("Label7").model.Label = _
    Sheet.getCellByPosition(1,z).String
  Dlg3.getControl("Label8").model.Label = _
    Sheet.getCellByPosition(0,z).Value
  Dlg3.getControl("Label9").model.Label = _
    Sheet.getCellByPosition(2,z).String
  Dlg3.getControl("CurrencyField1").model.Enabled = True
  Dlg3.getControl("TextField1").model.Enabled = True
  Dlg3.getControl("CurrencyField1").model.Value = _
    Sheet.getCellByPosition(3,z).Value
  Dlg3.getControl("TextField1").model.Text = _
    Sheet.getCellByPosition(4,z).String

```

```

        MsgBox(z)
    End If
End Sub
Sub Dlg3Off
    Dlg3.EndExecute
End Sub
Sub NextZ
    if z = 0 then z = 3
    z=z+1
    Dlg3.getControl("Label7").model.Label = _
        Sheet.getCellByPosition(1,z).String
    Dlg3.getControl("Label8").model.Label = _
        Sheet.getCellByPosition(0,z).Value
    Dlg3.getControl("Label9").model.Label = _
        Sheet.getCellByPosition(2,z).String
    Dlg3.getControl("CurrencyField1").model.Enabled = True
    Dlg3.getControl("TextField1").model.Enabled = True
    Dlg3.getControl("CurrencyField1").model.Value = _
        Sheet.getCellByPosition(3,z).Value
    Dlg3.getControl("TextField1").model.Text = _
        Sheet.getCellByPosition(4,z).String
    Dlg3.getControl("DateField1").model.Text = _
        Sheet.getCellByPosition(1,z).String
End Sub
Sub PreviousZ
    z = z-1
    Dlg3.getControl("Label7").model.Label = _
        Sheet.getCellByPosition(1,z).String
    Dlg3.getControl("Label8").model.Label = _
        Sheet.getCellByPosition(0,z).Value
    Dlg3.getControl("Label9").model.Label = _
        Sheet.getCellByPosition(2,z).String

```

```
Dlg3.getControl("CurrencyField1").model.Enabled = True
Dlg3.getControl("TextField1").model.Enabled = True
Dlg3.getControl("CurrencyField1").model.Value = _
    Sheet.getCellByPosition(3,z).Value
Dlg3.getControl("TextField1").model.Text = _
    Sheet.getCellByPosition(4,z).String
Dlg3.getControl("DateField1").model.Text = _
    Sheet.getCellByPosition(1,z).String
```

End Sub

Sub SaveChanges

```
Sheet.getCellByPosition(4,z).String =
Dlg3.getControl("TextField1").model.Text
Sheet.getCellByPosition(3,z).Value =
Dlg3.getControl("CurrencyField1").model.Value
Dlg3.EndExecute
```

End Sub

Rem Balance -----

Sub Dlg4off

```
Dlg4.EndExecute
```

End Sub

Первый диалог (форма)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE dlg:window PUBLIC "-//OpenOffice.org//_
    DTD OfficeDocument 1.0//EN" "dialog.dtd">
```

```
<dlg:window          xmlns:dlg=http://openoffice.org/2000/dialog_
xmlns:script="http://openoffice.org/2000/script"          dlg:id="Dialog1" _
dlg:left="106"      dlg:top="77"      dlg:width="145"      dlg:height="129" _
dlg:closeable="true" dlg:moveable="true" dlg:title=_
"Учет домашних финансов 1.0.">
```

```
<dlg:bulletinboard>
```

```
<dlg:button dlg:id="Cmd1" dlg:tab-index="0" dlg:left="23" _
```

```

dlg:top="31" dlg:width="99" dlg:height="18" dlg:_
value="Ввод данных о доходах и расходах">
  <script:event script:event-name="on-performaction" _
script:macroname="vnd.sun.star.script:Standard.Module1._
Window2?language=Basic&location=document" script:language="Script"/>
</dlg:button>
<dlg:button dlg:id="Cmd2" dlg:tab-index="1" dlg:left="23" _
dlg:top="56" dlg:width="99" dlg:height="16" _
dlg:value="Вывод данных о доходах и расходах">
  <script:event script:event-name="on-performaction" _
script:macroname="vnd.sun.star.script:Standard.Module1.Window3_
?language=Basic&location=document" script:language="Script"/>
</dlg:button>
<dlg:text dlg:id="Lb1" dlg:tab-index="2" dlg:left="15" _
dlg:top="9" dlg:width="122" dlg:height="18" _
dlg:value="Добро пожаловать в систему учета финансов."/>
<dlg:button dlg:id="Cmd3" dlg:tab-index="3" dlg:left="23" _
dlg:top="78" dlg:width="99" dlg:height="16" dlg:_
value="Баланс расходов и доходов">
  <script:event script:event-name="on-performaction" _
script:macro-name="vnd.sun.star.script:Standard.Module1._
Window4?language=Basic&location=document" script:language="Script"/>
</dlg:button>
<dlg:button dlg:id="Cmd4" dlg:tab-index="4" dlg:left="23" _
dlg:top="100" dlg:width="99" dlg:height="16" dlg:value="Выход">
  <script:event script:event-name="on-performaction" _
script:macroname="vnd.sun.star.script:Standard.Module1._
Dlg1Off?language=Basic&location=document" script:language="Script"/>
</dlg:button>
</dlg:bulletinboard>
</dlg>window>

```

Второй диалог

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE dlg:window PUBLIC "-//OpenOffice.org/_
DTD OfficeDocument 1.0//EN" "dialog.dtd">
<dlg:window
    xmlns:dlg=http://openoffice.org/2000/dialog
    xmlns:script="http://openoffice.org/2000/script"
    dlg:id="Dialog2"
    dlg:left="183"
    dlg:top="151"
    dlg:width="130"
    dlg:height="114"
    dlg:closeable="true"
    dlg:moveable="true"
    dlg:title="Ввод данных о расходах и доходах">
    <dlg:bulletinboard>
        <dlg:text
            dlg:id="Label1"
            dlg:tab-index="0"
            dlg:left="6"
            dlg:top="6"
            dlg:width="40"
            dlg:height="8"
            dlg:value="Текущая дата:"/>
        <dlg:text
            dlg:id="Label2"
            dlg:tab-index="1"
            dlg:left="6"
            dlg:top="24"
            dlg:width="40"
            dlg:height="8"
            dlg:value="Номер записи:"/>
        <dlg:text
            dlg:id="Label5"
            dlg:tab-index="4"
            dlg:left="6"
            dlg:top="42"
            dlg:width="40"
            dlg:height="8"
            dlg:value="Тип записи" />
        <dlg:text
            dlg:id="Label6"
            dlg:tab-index="5"
            dlg:left="6"
            dlg:top="60"
            dlg:width="40"
            dlg:height="8"
            dlg:value="Сумма"/>
        <dlg:text
            dlg:id="Label7"
            dlg:tab-index="6"
            dlg:left="6"
            dlg:top="78"
            dlg:width="40"
            dlg:height="8"
            dlg:value="Примечание"/>
        <dlg:fixedline
            dlg:id="FixedLine1"
            dlg:tab-index="12"
            dlg:left="64"
            dlg:top="6"
            dlg:width="2"
            dlg:height="80"
            dlg:align="vertical"/>
        <dlg:button
            dlg:id="Cmd2"
            dlg:tab-index="11"
            dlg:left="71"
            dlg:top="96"
            dlg:width="53"
            dlg:height="12"
            dlg:value="Выход">
            <script:event
                script:event-name="on-performaction"
                script:macroname="vnd.sun.star.script:Standard.Module1.Dlg2Off
                ?language=Basic&location=document"
                script:language="Script"/>
        </dlg:button>
        <dlg:button
            dlg:id="Cmd1"
            dlg:tab-index="10"
            dlg:left="6"
            dlg:top="96"
            dlg:width="53"
            dlg:height="12"
            dlg:value="Записать">
            <script:event
                script:eventname="on-performaction"
                script:macro-name="vnd.sun.star.script:Standard.Module1.
                WriteToTable?language=Basic&location=document"
                script:language="Script"/>
        </dlg:button>
        <dlg:text
            dlg:id="LabelDate"
            dlg:tab-index="2"
            dlg:left="71"
            dlg:top="6"
            dlg:width="43"
            dlg:height="8"
            dlg:value="Date"/>
```

```

<dlg:textfield      dlg:id="TextFieldComment"      dlg:tab-index="9" _
dlg:left="71" dlg:top="74" dlg:width="53" dlg:height="12"/>
<dlg:text      dlg:id="LabelId"      dlg:tab-index="3"      dlg:left="71" _
dlg:top="24" dlg:width="38" dlg:height="8" dlg:value="id"/>
<dlg:combobox      dlg:id="ComboBoxType"      dlg:tab-index="7" _
dlg:left="71"      dlg:top="38"      dlg:width="53"      dlg:height="12" _
dlg:spin="true" dlg:linecount="16">
  <dlg:menupopup>
    <dlg:menuitem dlg:value="Расход"/>
    <dlg:menuitem dlg:value="Доход"/>
  </dlg:menupopup>
</dlg:combobox>
<dlg:numericfield dlg:id="NumericFieldSumma" _
  dlg:tab-index="8" dlg:left="71" dlg:top="56" dlg:width="53" _
  dlg:height="12" dlg:value-min="0" dlg:spin="true"/>
</dlg:bulletinboard>
</dlg>window>

```

Третий диалог

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dlg>window PUBLIC "-//OpenOffice.org//_
DTD OfficeDocument 1.0//EN" "dialog.dtd">
<dlg>window      xmlns:dlg=http://openoffice.org/2000/dialog _
xmlns:script="http://openoffice.org/2000/script"      dlg:id="Dialog3" _
dlg:left="106"      dlg:top="123"      dlg:width="106"      dlg:height="174" _
dlg:closeable="true" dlg:moveable="true" dlg:title="Вывод данных о доходах и расходах">
  <dlg:bulletinboard>
    <dlg:datefield dlg:id="DateField1" dlg:tab-index="0" _
  dlg:left="29" dlg:top="16" dlg:width="51" dlg:height="14" _
  dlg:date-format="system_short_YYYY" dlg:value="20160401" _
  dlg:value-min="20000101" dlg:dropdown="true">
    <script:event script:event-name="on-textchange" _
  script:macro-name="vnd.sun.star.script:Standard.Module1._
ChooseDate?language=Basic&location=document"script:language="Script"/>

```

```
</dlg:datefield>
<dlg:text dlg:id="Label1" dlg:tab-index="1" dlg:left="7" _
dlg:top="4" dlg:width="98" dlg:height="8" dlg:value="Выберите дату для вывода записи"/>
<dlg:fixedline dlg:id="FixedLine1" dlg:tab-index="2" dlg:left="3" _
  dlg:top="36" dlg:width="101" dlg:height="2"/>
<dlg:text dlg:id="Label2" dlg:tab-index="3" dlg:left="6" _ dlg:top="48"
dlg:width="42" dlg:height="8" dlg:value="Дата записи:"/>
<dlg:text dlg:id="Label3" dlg:tab-index="4" dlg:left="6" _
dlg:top="62" dlg:width="43" dlg:height="8" dlg:value="Номер записи:"/>
<dlg:text dlg:id="Label4" dlg:tab-index="5" dlg:left="6" _ dlg:top="76"
dlg:width="42" dlg:height="8" dlg:value="Тип записи:"/>
<dlg:text dlg:id="Label5" dlg:tab-index="6" dlg:left="6" _ dlg:top="90"
dlg:width="43" dlg:height="8" dlg:value="Сумма:"/>
<dlg:text dlg:id="Label6" dlg:tab-index="7" dlg:left="6" _
  dlg:top="104" dlg:width="41" dlg:height="8" dlg:value="Комментарий:"/>
<dlg:button dlg:id="CommandButton1" dlg:tab-index="8" _ dlg:left="6"
dlg:top="148" dlg:width="64" dlg:height="18" dlg:value="Сохранить изменения">
  <script:event script:event-name="on-performaction" _
script:macro-name="vnd.sun.star.script:Standard.Module1._
SaveChanges?language=Basic&location=document" script:language="Script"/>
</dlg:button>
<dlg:button dlg:id="CommandButton2" dlg:tab-index="9" _ dlg:left="76"
dlg:top="148" dlg:width="27" dlg:height="16" _ dlg:value="Выход">
  <script:event script:event-name="on-performaction" _
script:macro-name="vnd.sun.star.script:Standard.Module1._
Dlg3Off?language=Basic&location=document" script:language="Script"/>
</dlg:button>
<dlg:button dlg:id="CommandButton3" dlg:tab-index="10" _ dlg:left="78"
dlg:top="59" dlg:width="10" dlg:height="12" dlg:value="&lt;">
  <script:event script:event-name="on-performaction" _
script:macro-name="vnd.sun.star.script:Standard.Module1._
PreviousZ?language=Basic&location=document" script:language="Script"/>
</dlg:button>
<dlg:button dlg:id="CommandButton4" dlg:tab-index="11" _ dlg:left="89"
```

```

dlg:top="59" dlg:width="10" dlg:height="12" dlg:value="&gt;">
  <script:event script:event-name="on-performaction" _
script:macro-name="vnd.sun.star.script:Standard.Module1._
NextZ?language=Basic&location=document" script:language="Script"/>
</dlg:button>
<dlg:text   dlg:id="Label7"   dlg:tab-index="12"   dlg:left="59" _
dlg:top="48" dlg:width="36" dlg:height="8"  dlg:value="kek"/>
<dlg:text   dlg:id="Label8"   dlg:tab-index="13"   dlg:left="59" _
dlg:top="62" dlg:width="15" dlg:height="8"  dlg:value="id"/>
<dlg:text   dlg:id="Label9"   dlg:tab-index="14"   dlg:left="59" _
dlg:top="76" dlg:width="35" dlg:height="8"  dlg:value="type"/>
<dlg:currencyfield  dlg:id="CurrencyField1"  dlg:tab-index="15" _
dlg:disabled="true"  dlg:left="57"  dlg:top="89"  dlg:width="36" _
dlg:height="12" dlg:spin="true"/>
<dlg:textfield      dlg:id="TextField1"      dlg:tab-index="16" _
dlg:disabled="true"  dlg:left="5"  dlg:top="116"  dlg:width="97" _
dlg:height="25" dlg:multiline="true"/>
<dlg:textfield  dlg:id="TextField2"  dlg:tab-index="17"  dlg:left="86" _
dlg:top="116"  dlg:width="0"  dlg:height="2"/>
<dlg:fixedline  dlg:id="FixedLine2"  dlg:tab-index="18"  dlg:left="2" _
dlg:top="102"  dlg:width="105"  dlg:height="2"/>
</dlg:bulletinboard>
</dlg>window>

```

Четвертый диалог

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dlg>window PUBLIC "-//OpenOffice.org//_
DTD OfficeDocument 1.0//EN" "dialog.dtd">
<dlg>window      xmlns:dlg=http://openoffice.org/2000/dialog
xmlns:script="http://openoffice.org/2000/script"      dlg:id="Dialog4" _
dlg:left="183"   dlg:top="151"   dlg:width="144"   dlg:height="90" _
dlg:closeable="true"  dlg:moveable="true"  dlg:title="Баланс доходов и расходов">
<dlg:bulletinboard>
  <dlg:text  dlg:id="Label1"  dlg:tab-index="0"  dlg:left="10" _

```



```

dlg:top="19" dlg:width="126" dlg:height="8" _
dlg:value="Баланс доходов и расходов на текущую дату"/>
<dlg:text dlg:id="Label2" dlg:tab-index="1" dlg:left="6" _ dlg:top="36"
dlg:width="22" dlg:height="8" dlg:value="Доход:"/>
<dlg:text dlg:id="Label3" dlg:tab-index="2" dlg:left="40" _ dlg:top="36"
dlg:width="26" dlg:height="8" dlg:value="Label3"/>
<dlg:text dlg:id="Label4" dlg:tab-index="3" dlg:left="78" _ dlg:top="36"
dlg:width="22" dlg:height="8" dlg:value="Расход:"/>
<dlg:text    dlg:id="Label5"    dlg:tab-index="4"    dlg:left="112" _
dlg:top="36" dlg:width="26" dlg:height="8" dlg:value="Label5"/>
<dlg:text dlg:id="Label6" dlg:tab-index="5" dlg:left="6" _ dlg:top="54"
dlg:width="99" dlg:height="8" dlg:value="Разница между доходом и расходом:"/>
<dlg:fixedline dlg:id="FixedLine1" dlg:tab-index="6" dlg:left="70" _
dlg:top="33" dlg:width="2" dlg:height="16" dlg:align="vertical"/>
<dlg:text    dlg:id="Label7"    dlg:tab-index="7"    dlg:left="112" _
dlg:top="54" dlg:width="16" dlg:height="8" dlg:value="Label7"/>
<dlg:button dlg:id="CommandButton1" dlg:tab-index="8" _ dlg:left="96"
dlg:top="77" dlg:width="46" dlg:height="11" dlg:value="Выход">
<script:event script:event-name="on-performaction" _
    script:macro-name="vnd.sun.star.script:Standard.Module1._
    Dlg4off?language=Basic&location=document" script:language="Script"/>
</dlg:button>
</dlg:bulletinboard>
</dlg>window>

```

Листинг 12. Программный код для выполнения задания 3

Результат работы программного кода представлен на рисунке 21.

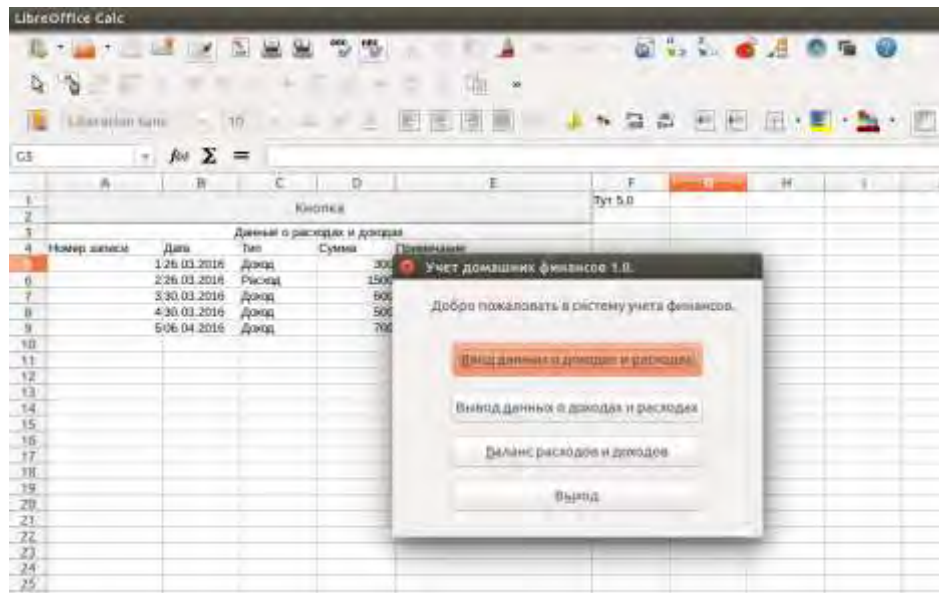


Рисунок 21. Окно программы для учета финансов

Задание 4. Создайте форму, которая позволит распечатать на принтере приказ о поощрении студентов вашей группы за какие-то достижения.

Код программы приведен в листинге 13. Результат выполнения программы показан на рисунке 22.

```

REM ***** BASIC *****
dim Dlg, Doc, Sheet as Object
dim Zasluga as String
Sub Window
  DialogLibraries.LoadLibrary("Standard")
  Dlg = CreateUnoDialog(DialogLibraries.Standard.Dialog1)
  Doc = StarDesktop.CurrentComponent
  Sheet = Doc.Sheets(0)
  dim box1(15)
  for i=0 to 15
    box1(i) = Sheet.getCellByPosition(0,i).String
  Next i
  for i = 0 to 15
    if Sheet.getCellByPosition(0,i).String = "" then k = k + 0 else k = k + 1
  Next i
  dim box2(k-1)
  for x= 0 to k - 1

```

```

        box2(x)=box1(x)
    Next x
    Dlg.getControl("ComboBox1").AddItems(box2,0)
    Dlg.Execute()
End Sub
Sub Other
    Dlg.getControl("txt1").model.Enabled = True
End Sub
Sub First
    Dlg.getControl("txt1").model.Enabled = False
    Zasluga = "За успешное освоение объектно-ориентированного программирования"
End Sub
Sub Second
    Dlg.getControl("txt1").model.Enabled = False
    Zasluga = "За внедрение новых программных продуктов"
End Sub
Sub Premia
    Dlg.getControl("NumericField1").model.Enabled = Not
    Dlg.getControl("NumericField1").model.Enabled
End Sub
Sub Generate
    Sheet = Doc.Sheets.GetByName("prikaz")
    if Dlg.getControl("txt1").model.Enabled = True then
        Zasluga = "За "
        + Dlg.getControl("txt1").model.Text
    Sheet.getCellByPosition(0,3).String = Zasluga + " награждается"
    Sheet.getCellByPosition(0,4).String =
    Dlg.getControl("ComboBox1").model.Text
    if Dlg.getControl("NumericField1").model.Enabled = True then
        Razmer = Dlg.getControl("NumericField1").model.value
        PSR = "– Денежной премией в размере " + razmer + " руб-
лей."
    if Dlg.getControl("CheckBox1").model.State = 1 then

```

```

NL = " – Почетной грамотой"
Sheet.getCellByPosition(0,5).String = PSR + Chr$(10) + NL
Vremya = "Дата: " + Day(Now) + "/" + Month(Now) + "/" + Year(Now)
Sheet.getCellByPosition(0,2).String = Vremya

```

'Печать -----

```

dim document as object
dim dispatcher as object
document = ThisComponent.CurrentController.Frame
dispatcher = createUnoService("com.sun.star.frame.DispatchHelper")
dim args1(0) as new com.sun.star.beans.PropertyValue
args1(0).Name = "Nr"
args1(0).Value = 2
Dlg.EndExecute
dispatcher.executeDispatch(document, ".uno:JumpToTable", "", 0,
args1())
dispatcher.executeDispatch(document, ".uno:PrintDefault", "", 0, Array())
MsgBox("После завершения печати нажмите Ок")
args1(0).Value = 1
dispatcher.executeDispatch(document, ".uno:JumpToTable", "", 0,
args1())
End Sub

```

```

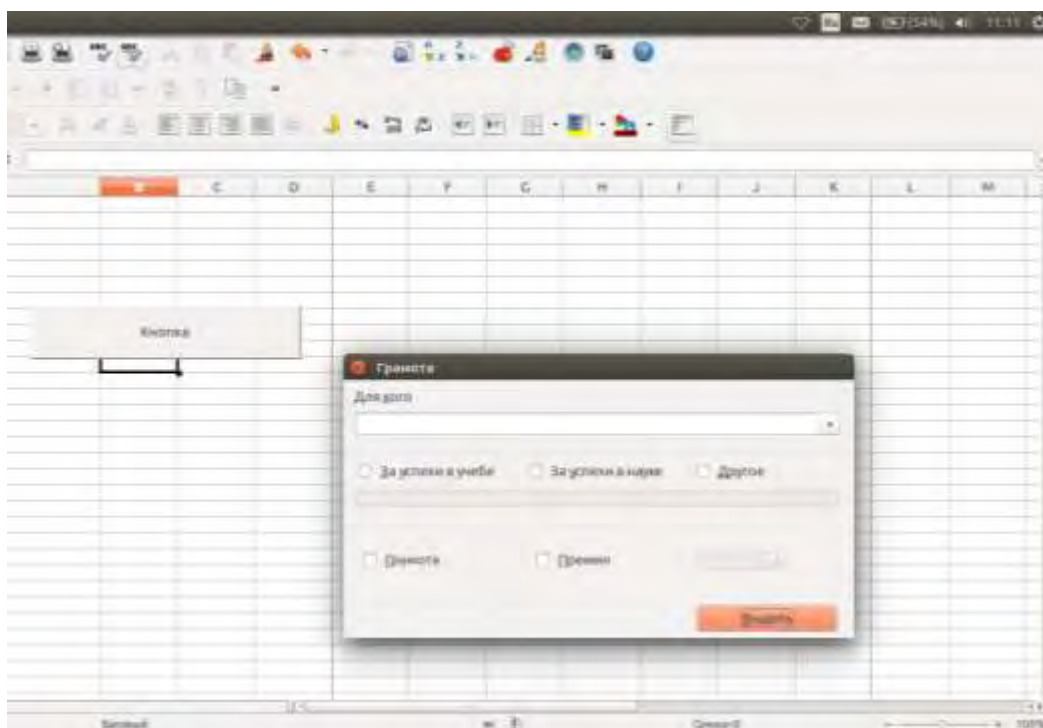
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dlg:window PUBLIC "-//OpenOffice.org//_
DTD OfficeDocument 1.0//EN" "dialog.dtd">
<dlg:window
xmlns:dlg=http://openoffice.org/2000/dialog
xmlns:script="http://openoffice.org/2000/script"
dlg:id="Dialog1"
dlg:left="0"
dlg:top="141"
dlg:width="201"
dlg:height="129"
dlg:closeable="true"
dlg:moveable="true"
dlg:title="Грамота">
<dlg:bulletinboard>
<dlg:radiogroup>
<dlg:radio dlg:id="Ob1"
dlg:tab-index="0"
dlg:left="4"
dlg:top="40"
dlg:width="60"
dlg:height="8"
dlg:value="За успехи в учебе"
dlg:group-name="g1">

```

```
<script:event script:event-name="on-performaction" _
script:macro-name="vnd.sun.star.script:Standard.Module1.First? _
language=Basic&location=document" script:language="Script"/>
</dlg:radio>
<dlg:radio dlg:id="Ob2" dlg:tab-index="1" dlg:left="71" _
dlg:top="40" dlg:width="60" dlg:height="8" _
dlg:value="За успехи в науке" dlg:group-name="g1">
<script:event script:event-name="on-performaction" _
script:macro-name="vnd.sun.star.script:Standard.Module1.Second? _
language=Basic&location=document" script:language="Script"/>
</dlg:radio>
<dlg:radio dlg:id="Ob3" dlg:tab-index="2" dlg:left="137" dlg:top="40" _
dlg:width="60" dlg:height="8" dlg:value="Другое" dlg:group-name="g1">
<script:event script:event-name="on-performaction" _
script:macro-name="vnd.sun.star.script:Standard.Module1.Other? _
language=Basic&location=document" script:language="Script"/>
</dlg:radio>
</dlg:radiogroup>
<dlg:textfield dlg:id="txt1" dlg:tab-index="3" dlg:disabled="true" _
dlg:left="4" dlg:top="55" dlg:width="192" dlg:height="9"/>
<dlg:button dlg:id="cmd1" dlg:tab-index="4" dlg:left="139" _
dlg:top="112" dlg:width="56" dlg:height="14" dlg:value="Выдать">
<script:event script:event-name="on-performaction" script:_
macro-name="vnd.sun.star.script:Standard.Module1.Generate? _
language=Basic&location=document" script:language="Script"/>
</dlg:button>
<dlg:numericfield dlg:id="NumericField1" dlg:tab-index="5" _
dlg:disabled="true" dlg:left="139" dlg:top="86" dlg:width="35" _
dlg:height="9" dlg:value-min="0" dlg:value-step="100" dlg:spin="true"/>
<dlg:text dlg:id="Label1" dlg:tab-index="6" dlg:left="4" dlg:top="4" _
dlg:width="42" dlg:height="8" dlg:value="Для кого"/>
<dlg:combobox dlg:id="ComboBox1" dlg:tab-index="7" dlg:left="4" _
dlg:top="16" dlg:width="192" dlg:height="12" dlg:spin="true" dlg:linecount="7"/>
```

```
<dlg:fixedline dlg:id="FixedLine1" dlg:tab-index="8" dlg:left="4" _
dlg:top="31" dlg:width="193" dlg:height="8"/>
<dlg:checkbox dlg:id="CheckBox1" dlg:tab-index="10" dlg:left="6" _
dlg:top="84" dlg:width="60" dlg:height="9" dlg:value="Грамота" _
dlg:checked="false">
  <script:event script:event-name="on-performaction" script:_
macro-name="vnd.sun.star.script:Standard.Module1.Gramota?_
language=Basic&location=document" script:language="Script"/>
</dlg:checkbox>
<dlg:checkbox dlg:id="CheckBox2" dlg:tab-index="9" dlg:left="74" _
dlg:top="84" dlg:width="57" dlg:height="7" dlg:value="Премия" _
dlg:checked="false">
  <script:event script:event-name="on-performaction" script:_
macro-name="vnd.sun.star.script:Standard.Module1.Premia?_
language=Basic&location=document" script:language="Script"/>
</dlg:checkbox>
<dlg:fixedline dlg:id="FixedLine2" dlg:tab-index="11" dlg:left="3" _
dlg:top="70" dlg:width="194" dlg:height="4"/>
<dlg:fixedline dlg:id="FixedLine3" dlg:tab-index="12" dlg:left="69" _
dlg:top="77" dlg:width="1" dlg:height="50" dlg:align="vertical"/>
</dlg:bulletinboard>
</dlg>window>
```

Листинг 13. Программный код для выполнения задания 4



1	Приказ				
2	ОГТИ филиал ОГУ Физико-математический факультет Группы 14700014				
3	Дата: 12/9/2016				
4	За предоставление учебных программных продуктов на территории				
5	Норм тип				
6	Денежной помощи в размере 100 рублей				
7	Почтовый перевод				
8	Сторона группы:	Уладков Елена			
9	От: Исполнитель:	Афанасов Сергей Михайлович			

Рисунок 22. Окно формы для создания приказа

В случае возникновения проблем с оплатой и продлением лицензии на использование Microsoft, использование альтернативного свободного программного обеспечения является успешным решением. Следует отметить, что в последних версиях OpenOffice и LibreOffice открываются и корректно отображаются документы, таблицы, базы данных и презентации, созданные в последних версиях Microsoft Office, так что методический материал не нужно будет перерабатывать.

[Вернуться к содержанию](#)

Библиографический список

1. Абдулвелеева, Р. Р. Технология моделирования компьютерной диагностики методической готовности студентов педагогического вуза / Р. Р. Абдулвелеева // Информатизация образования и науки. – 2014. – № 3 (23). – С. 42-53.
2. Абдулвелеева, Р. Р. Использование программирования на РНР в инженерных расчетах / Р. Р. Абдулвелеева, Р. И. Абдулвелеев, И. Р. Абдулвелеев // Информатизация образования и науки. – 2015. – № 3 (27). – С. 74-83.
3. Абдулвелеева, Р. Р. Компьютерная диагностика методической готовности студентов педагогического вуза : диссертация на соискание ученой степени кандидата педагогических наук / Р. Р. Абдулвелеева. – Оренбург, 2001.
4. Абдулвелеева, Р. Р. Компьютерная программа мониторинга методической компетентности будущего учителя информатики / Р. Р. Абдулвелеева, И. Э. Биккулов, Р. И. Абдулвелеев // Научные труды Sword. – Выпуск № 2 (39) – Иваново : Научный мир, 2015. – С. 45-49.
5. Абдулвелеева, Р. Р. Автоматизация расчета основных характеристик теста / Р. Р. Абдулвелеева, Р. И. Абдулвелеев, Н. В. Воронина // Социосфера. – 2015. – № 3. – С. 107-112.
6. Абдулвелеева, Р. Р. Адаптация альтернативного программного обеспечения в курсе информатики / Р. Р. Абдулвелеева, Р. И. Абдулвелеев, Е. С. Соболев // Современные тенденции развития и перспективы внедрения инновационных технологий в машиностроении, образовании и экономике. – 2016. – № 1. – С. 162-163.
7. Абдулвелеева, Р. Р. Компьютерное адаптивное тестирование в процессе изучения программирования / Р. Р. Абдулвелеева, Р. И. Абдулвелеев, Д. В. Скубакова // Современные тенденции развития и перспективы внедрения инновационных технологий в машиностроении, образовании и экономике. – 2016. – № 1. – С. 155-157.
8. Абдулвелеева, Р. Р. Объектно-множественная модель компьютерной диагностики учебно-профессиональной деятельности бакалавров по направлению подготовки 050100 – Педагогическое образование (профиль Информатика

и ИКТ) / Р. Р. Абдулвелеева, Р. И. Абдулвелеев // X Міжнародна науково-практична студентська конференція магістрантів : Матеріали конференції. – 2016. – С. 77-78.

9. Абдулвелеева, Р. Р. Дифференцированный подход при изучении программирования в среде VBA / Р. Р. Абдулвелеева, Д. В. Скубакова, Р. И. Абдулвелеев // Мир науки и инноваций. – 2016. – Т. 6. – № 1 (3). – С. 73-76.

10. Абдулвелеева, Р. Р. Визуализация показателей математической статистики в Excel для изучения качества диагностических заданий / Р. Р. Абдулвелеева, Н. В. Воронина, Р. И. Абдулвелеев // Мир науки и инноваций. – 2016. – Т. 6. – № 1 (3). – С. 76-79.

11. Абдулвелеева, Р. Р. Дифференцированный подход при изучении программирования в среде VBA / Р. Р. Абдулвелеева, Д. В. Скубакова, Р. И. Абдулвелеев // Научный взгляд в будущее. – 2016. – Т. 6. – № 2 (2). – С. 73-76.

12. Абдулвелеева, Р. Р. Визуализация показателей математической статистики в Excel для изучения качества диагностических заданий / Р. Р. Абдулвелеева, Н. В. Воронина, Р. И. Абдулвелеев // Научный взгляд в будущее. – 2016. – Т. 6. – № 2 (2). – С. 76-79.

13. Абдулвелеева, Р. Р. Автоматизация процесса диагностики учебно-профессиональной деятельности бакалавров / Р. Р. Абдулвелеева, Р. И. Абдулвелеев // Информатика, управління та штучний інтелект: матеріали другої науково-технічної конференції студентів, магістрів та аспірантів. Національний технічний університет "Харківський політехнічний інститут". – 2015. – С. 4.

14. Функции мониторинга профессионально-методической деятельности студентов педагогического вуза / Р. Р. Абдулвелеева // Сборники конференций НИЦ Социосфера. – 2015. – № 16. – С. 110-111.

15. Абдулвелеева, Р. Р. Компьютерная программа оценки качества теста / Р. Р. Абдулвелеева, Н. В. Воронина // Мир науки и инноваций. – 2015. – Т. 6. – С. 68-71.

16. Абдулвелеева, Р. Р. Электронный дидактический комплекс управления учебно-профессиональной деятельностью студента / Р. Р. Абдулвелеева //

Хроники объединенного фонда электронных ресурсов Наука и образование. – 2013. – № 4 (47). – С. 27.

17. Абдулвелеева, Р. Р. Диагностика уровня развития рефлексии учебно-профессиональной деятельности студента / Р. Р. Абдулвелеева // Научные труды SWorld. – 2013. – Т. 19. – № 1. – С. 68-72.

18. Аванесов, В. С. Теория и методика педагогических измерений. Режим доступа: <http://viperson.ru/data/200812/jbjejbjxjklmjuje.pdf>

19. Бычков, М. И. Основы программирования на VBA для Microsoft Excel : учебное пособие / М. И. Бычков. – Новосибирск : НГТУ, 2010. [Электронный ресурс]. Режим доступа: http://biblioclub.ru/index.php?page=book_view_red&book_id=228951

20. Гавришина, О. Н. Технология программирования на Visual Basic for Application в MsOffice : учебное пособие / О. Н. Гавришина. – Кемерово : Кемеровский государственный университет, 2010. Режим доступа: http://biblioclub.ru/index.php?page=book_view_red&book_id=232354

21. Лавлинский, В. В. Технология программирования на современных языках программирования [Электронный ресурс] / В. В. Лавлинский, О. В. Коровина. – Воронежская государственная лесотехническая академия, 2012. Режим доступа : http://biblioclub.ru/index.php?page=book_view_red&book_id=142453

22. Практические задания по созданию пользовательских приложений с помощью VBA : учебно-методическое пособие [Электронный ресурс] / М. В. Маркина, В. В. Денисов, О. Г. Савихин.

23. Тихомиров, В. О. Введение в LINUX : учебное пособие / В. О. Тихомиров. – М. : МИФИ, 2007. Режим доступа: http://biblioclub.ru/index.php?page=book_view_red&book_id=231109

24. Угринович, Н. Д. Практикум по информатике и информационным технологиям / Н. Д. Угринович, Л. Л. Босова, Н. И. Михайлова. – М. : БИНОМ. Лаборатория знаний, 2004. – 394 с.

25. Чельшкова, М. Б. Теория и практика конструирования педагогических тестов : учебное пособие / М. Б. Чельшкова. – М. : Логос, 2002. - 432 с.: ил. ISBN 5-94010-143-7.

26. Abdulveleeva, R. R. The differentiated approach when studying programming in the environment of vba / R. R. Abdulveleeva, R. I. Abdulvelev // SWorldJournal. – 2016. – Т. 08. – № j116 (10). – С. 35-38.

27. Abdulveleeva, R. R. Visualization of indicators of mathematical statistics in excel for studying of quality of diagnostic tasks / R. R. Abdulveleeva, R. I. Abdulvelev // SWorldJournal. – 2016. – Т. 08. – № j116 (10). – С. 38-40.

28. Abdulveleeva, R. R. Automation of approbation of test in excel / R. R. Abdulveleeva // SWorldJournal. – 2015. – Т. 8. – № 1 (8). – С. 195-198.

29. Abdulveleeva, R. R. Computer program monitoring of the methodical competence future teacher of informatics / R. R. Abdulveleeva, R. I. Abdulvelev // SWorldJournal. – 2015. – Т. J21510. – № 2. – С. 27-32.

Интернет ресурсы:

1. http://studopedia.su/4_19290_sozdanie-ob-ektnih-peremennih.html
2. <http://www.4stud.info/ppp/lecture5.html>
3. <http://www.4stud.info/ppp/practice.html>
4. <http://www.4stud.info/ppp/lecture5.html>
5. <http://www.intuit.ru/studies/courses/2251/113/lecture/3305?page=8>

Приложение 1

Основные свойства объекта *UserForm*

Свойство	Описание
ActiveControl	Возвращает объектную ссылку на элемент управления, находящийся в фокусе в данный момент. Свойство только для чтения
BackColor	Возвращает целое значение типа Long, которое определяет цвет фона формы
BorderStyle	Устанавливает тип границы
Caption	Возвращает текст, отображаемый в строке заголовка формы
Controls	Возвращает коллекцию всех элементов управления формы. Только для чтения
Cycle	Определяет, должно ли нажатие клавиши табуляции вызывать последовательный выбор всех элементов управления во всех группах и на каждой странице многостраничных элементов управления или только в пределах текущей группы или страницы. Может принимать значение одной из встроенных констант: <code>fmCycleAllForms</code> или <code>fmCycleCurrentForms</code>
Enabled	Содержит значение типа Boolean, указывающее, доступна ли форма. Если его значение равно <code>False</code> , ни один из элементов управления формы не доступен
Font	Позволяет выбрать параметры шрифта формы или элемента управления
ForeColor	То же самое, что и свойство <code>BackColor</code> , но устанавливает цвет, используемый для переднего плана (обычно это цвет текста) объекта формы
Height и Width	Возвращают высоту и ширину формы в пунктах
Left и Top	Возвращают местоположение левого верхнего угла формы в пунктах
Name	Возвращает имя пользовательской формы
Picture	Указывает рисунок, отображаемый как фон формы
StartPosition	Возвращает значение, определяющее положение формы при ее первом отображении на экране. Допустимые значения: <code>Manual</code> – начальное значение не устанавливается; <code>CenterOwner</code> – выравнивание по центру объекта, которому принадлежит форма; <code>CenterScreen</code> – выравнивание по центру экрана; <code>Windows Default</code> – положение верхнего левого угла экрана

[Вернуться к содержанию](#)

Приложение 2

Основные методы объекта *UserForm*

Метод	Назначение
Copy	Копирует выделенный в элементе управления текст в буфер обмена Windows
Cut	Вырезает выделенный в элементе управления текст и помещает его в буфер обмена Windows
Hide	Скрывает форму, не выгружая ее из памяти, сохраняя значения элементов управления формы и всех переменных, объявленных в модуле класса формы
Move	Изменяет положение и размер формы
Paste	Вставляет содержимое буфера обмена Windows в текущий элемент управления
PrintForm	Выводит на используемый в Windows по умолчанию принтер изображение формы, включая все данные, введенные в элементы управления
Repaint	Перерисовывает форму, выведенную на экран.
Show	Выводит форму на экран. Если форма еще не загружена в память, то данный метод сначала ее загружает

[Вернуться к содержанию](#)

Основные события объекта UserForm

Событие	Описание
Activate	Происходит, когда окно формы становится активным. Используется для обновления содержимого диалоговых элементов управления, чтобы отразить любые изменения, которые произошли, пока окно формы было неактивным
Click	Происходит при щелчке мышью по форме
DblClick	Происходит при двойном щелчке мышью по форме
Deactivate	Происходит, когда форма перестает быть активной
Initialize	Происходит, когда форма впервые загружается в память посредством выполнения оператора Load или с помощью метода Show. Используйте это событие для инициализации элементов управления формы при ее появлении на экране
Resize	Происходит при изменении размеров формы
Terminate	Происходит при закрытии формы, то есть когда форма выгружается из памяти. Используется для осуществления специальных задач, которые необходимо выполнить прежде, чем переменные формы будут выгружены

[Вернуться к содержанию](#)

Приложение 4

Стандартные элементы управления, включенные в VBA

Элемент управления	Назначение
Label (надпись, метка)	Позволяет создавать заголовки элементов управления, которые не имеют собственных встроенных заголовков
TextBox (текстовое поле)	Окно редактируемого текста свободной формы для ввода данных. Может быть одно- и многострочным
ComboBox (поле со списком)	Объединяет окно редактирования и окно списка
ListBox (список)	Отображает список значений, из которых пользователь может сделать выбор
CheckBox (флажок)	Стандартный флажок, который используется для выбора вариантов, не являющихся взаимоисключающими
OptionButton (переключатель)	Стандартная кнопка-переключатель. Используется, когда пользователю необходимо сделать выбор между "включено/выключено"
ToggleButton (выключатель)	Выключатели служат для той же цели, что и флажки, но выводят установки в виде кнопки, находящейся в "нажатом" или "отжатом" состоянии
Frame (рамка)	Визуально и логически объединяет некоторые элементы управления (особенно флажки, переключатели и выключатели)
CommandButton (кнопка)	Используется для выполнения действий Cancel (Отмена), Save (Сохранить), Ok и т.д. Когда пользователь щелкает по кнопке, выполняется VBA-процедура, закрепленная за данным элементом управления
TabStrip (набор вкладок)	Состоит из области, в которую следует помещать другие элементы управления (такие, как текстовые поля, флажки и т. д.)
MultiPage (набор страниц)	Состоит из нескольких страниц. Можно выбрать любую из них, щелкнув по соответствующей вкладке
ScrollBar (полоса прокрутки)	Позволяет выбирать линейное значение, аналогичное тому, как это можно сделать при помощи счетчика
SpinButton (счетчик)	Разновидность текстового поля. Используется для ввода последовательных величин, которые заведомо находятся в определенном интервале значений (число, дата и т.п.)
Image (рисунок)	Выводит на форме графическое изображение любым из следующих форматов: *.bmp, *.cur, *.gif, *.ico, *.jpg, *.wmf

[Вернуться к содержанию](#)

Приложение 5

Основные свойства стандартных элементов управления

Свойство	Описание
1	2
Accelerator	Содержит символ, используемый в качестве клавиши быстрого вызова элемента управления. При нажатии Alt+<клавиша быстрого вызова> происходит выбор элемента управления
AutoSize	Значение типа Boolean. Если равно True – устанавливает режим автоматического изменения размеров элемента управления так, чтобы на нем полностью помещался текст, присвоенный свойству Caption
BackColor	Устанавливает цвет фона элемента управления
BorderColor	Устанавливает цвет границы элемента управления
BorderStyle	Устанавливает тип границы элемента управления Допустимые значения: fmBorderStyleSingle – граница в виде контура; fmBorderStyleNone – граница невидима
Caption	Надпись, отображаемая при элементе управления
Cancel	Задаёт кнопку отмены диалогового окна. Используется для элемента управления CommandButton. При нажатии на эту кнопку или клавишу Esc диалоговое окно исчезает
ControlTopText	Определяет текст, который отображается в виде всплывающей подсказки, когда указатель мыши помещается на элемент управления. В следующем примере элементу управления CommandButton1 назначен текст всплывающей подсказки "Это кнопка": CommandButton1.ControlTopText = "Это кнопка"
Default	Используется для элемента управления CommandButton. Определяет заданную по умолчанию кнопку. При нажатии на клавишу Enter эта кнопка ведёт себя так, как если бы по ней щёлкнули мышью
Enabled	Определяет, доступен элемент управления (значение True) или нет (значение False)
ForeColor	Устанавливает цвет для переднего плана элемента управления – как правило, символов текста
Height и Width	Устанавливают геометрические размеры объекта (высоту и ширину)
Left и Top	Устанавливают координаты верхнего левого угла элемента управления, определяющие его местоположение в форме
List	Представляет список, содержащийся в элементе управления (массив типа Variant). Используется для элементов управления ComboBox и ListBox

Свойство	Описание
1	2
Max	Переменная типа Long – определяет максимальное значение счетчика или значение, при котором полоса прокрутки находится в самом верху (для вертикальной полосы) или справа (для горизонтальной). Используется для элементов управления ScrollBar и SpinButton
Min	Переменная типа Long – определяет минимальное значение счетчика или значение, при котором полоса прокрутки находится в самом низу (для вертикальной полосы) или слева (для горизонтальной). Используется для элементов управления ScrollBar и SpinButton
Name	Содержит имя элемента управления
Picture (создание картинки)	Внедряет картинку на элемент управления. Например, на поверхности кнопки картинка отображается с помощью следующей инструкции: CommandButton1.Picture = LoadPicture("c:\my_doc\Круг.bmp") Функция LoadPicture (Полное имя файла) считывает графическое изображение
Picture (удаление картинки)	После того, как картинка создана на элементе управления, иногда возникает необходимость ее удалить. Это легко достигается присвоением свойству Picture значения LoadPicture("")
RowSource	Задает источник, из которого ComboBox или ListBox "берет" список объекта
SpecialEffect	Устанавливает тип границы. Отличается от свойства BorderStyle тем, что позволяет установить несколько типов, но одного цвета. BorderStyle позволяет установить только один тип, но различных цветов
TabIndex	Определяет число, указывающее положение элемента управления в порядке табуляции. Может иметь значение от 0 до значения, равного количеству элементов управления на форме
TabStop	Значение типа Boolean, которое указывает, может ли элемент управления быть выбран клавишей Tab
Tag	Используется для хранения дополнительной информации о форме или элементе управления, которая может быть в последующем востребована в программе

Свойство	Описание
1	2
Value	Значение текущих установок элемента управления: текст в текстовом поле, какие выбраны флажки и переключатели, индекс выбранного раздела списка или число, указывающее текущее положение полосы прокрутки или счетчика
Visible	Значение типа Boolean, указывающее, является ли элемент управления видимым

[Вернуться к содержанию](#)

Приложение 6

Основные общие методы элементов управления и события объектов управления

Метод	Описание
Add	Позволяет добавить элемент управления во время выполнения программы
Move	Перемещает элемент управления
SetFocus	Устанавливает фокус на вызвавшем этот метод элементе управления. Часто применяется в программах обработки ошибок
Zorder	Помещает объект до или после всех пересекающихся с ним объектов
Click	Происходит, когда пользователь выбирает элемент управления с помощью одинарного щелчка кнопкой мыши
DbClick	Происходит, когда пользователь выбирает элемент управления с помощью двойного щелчка кнопкой мыши
KeyDown	Происходит при нажатии пользователем какой-либо клавиши в тот момент, когда форма выполняется и имеет фокус
KeyPress	Происходит, когда пользователь нажимает любую клавишу на клавиатуре, кроме функциональных и клавиш управления курсором
KeyUp	Происходит, когда пользователь отпускает клавишу
Change	Происходит при изменении значения элемента управления
GotFocus	Происходит, когда элемент управления получает фокус
LostFocus	Происходит, когда элемент управления теряет фокус
Error	Используется при уведомлении об ошибке
MouseDown	Происходит при нажатии кнопки мыши
MouseUp	Происходит при отпускании кнопки мыши
MouseMove	Происходит при перемещении указателя мыши
BeforeDragOver	Происходит, пока совершается операция перемещения (drag-and-drop) элемента управления
BeforeDropOrPaste	Происходит перед завершением операции перемещения (drag-and-drop) элемента управления
AfterUpdate	Происходит после обновления значения элемента управления
BeforeUpdate	Происходит после того, как было изменено значение элемента управления, но перед тем, как был обновлен сам элемент управления
Enter	Происходит, когда выделяется элемент управления
Exit	Происходит, когда с элемента управления снимается выделение
SpinDown	Происходит, когда пользователь щелкает стрелку "вниз" ("влево") кнопки счетчика
SpinUp	Происходит, когда пользователь щелкает стрелку "вверх" ("вправо") кнопки счетчика

[Вернуться к содержанию](#)