

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего образования
Санкт-Петербургский горный университет

Кафедра информатики и компьютерных технологий

ИНФОРМАТИКА

СОЗДАНИЕ И РЕДАКТИРОВАНИЕ

ПОЛЬЗОВАТЕЛЬСКИХ ФОРМ

В СРЕДЕ VBA

*Методические указания к лабораторным работам
для студентов всех специальностей и направлений подготовки*

САНКТ-ПЕТЕРБУРГ
2020

ИНФОРМАТИКА. Создание и редактирование пользовательских форм в среде VBA: Методические указания к лабораторным работам / Санкт-Петербургский горный университет. Сост.: *Е.Н. Овчинникова, С.Ю. Кротова*. – СПб, 2020. 34 с.

В методических указаниях к лабораторным работам приведены примеры создания и редактирования пользовательских форм в среде программирования Visual BASIC for Applications. В конце каждой работы даны варианты для индивидуальных заданий.

Методические указания предназначены для студентов всех специальностей и направлений подготовки.

Научный редактор доц. *А.Б. Маховиков*

Рецензент проф. *С.М. Одоевский* (Санкт-Петербургская военная академия связи)

© Санкт-Петербургский
горный университет, 2020

ВВЕДЕНИЕ

Visual Basic for Applications (VBA) – это упрощённая версия среды программирования Visual Basic, встроенная в линейку продуктов Microsoft Office, а также во многие другие программные пакеты, в том числе AutoCAD, CorelDraw, Adobe Creative Suite и др. В среде VBA объединились возможности языков программирования высокого уровня и средства визуального и объектно-ориентированного программирования.

VBA является полноценной средой программирования, позволяющей создавать интерактивные Windows-приложения с графическим интерфейсом.

В методических указаниях рассмотрены основные возможности среды VBA для создания и редактирования пользовательских (экранных) форм. Представлены три лабораторные работы, включающие разработку приложений для решения вычислительных задач, технологию создания списков и приложений-тестов с кнопками выбора. В начале каждой лабораторной работы приводятся необходимые теоретические сведения по теме работы. Затем разбираются учебные примеры, которые служат образцом для выполнения индивидуальных заданий.

Цель методических указаний: содействовать формированию общекультурных и общепрофессиональных компетенций (умение пользоваться компьютером как средством управления и обработки информационных массивов; способность решать задачи профессиональной деятельности на основе информационной культуры с применением информационно-коммуникационных технологий); способствовать развитию внимательности, целеустремленности, навыков самоорганизации и самоконтроля.

Учебно-методический материал может быть использован как на аудиторных лабораторных занятиях, так и в рамках самостоятельной работы студентов всех специальностей и направлений подготовки, изучающих дисциплины «Информатика», «Информационные технологии», «Информатика и информационно-коммуникационные технологии».

ЛАБОРАТОРНАЯ РАБОТА №1

РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОЙ ФОРМЫ ДЛЯ РЕШЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ

1.1. ОБЩИЕ СВЕДЕНИЯ

Среда программирования VBA позволяет создавать интерактивные Windows-приложения с графическим интерфейсом. Для этого используются пользовательские формы (объект **UserForm**).

Для того чтобы в разрабатываемое приложение можно было добавить форму, необходимо выполнить следующие действия:

- запустить редактор VBA;
- выполнить команду **Insert – UserForm (Вставка – Экранная форма)**. На рабочий лист будет добавлена экранная форма, показанная на рисунке 1.1.

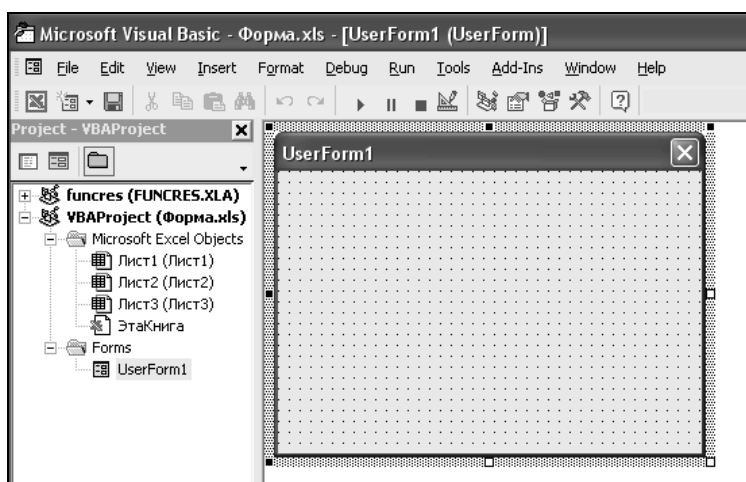


Рис. 1.1. Окно редактора VBA в режиме пользовательской формы

Экранная (пользовательская) форма представляет собой пустое серое окно со строкой заголовка. Это та основа, на которой пользователь может создать собственные окна в понимании идеологии операционной системы Windows.

Обычно на экранную форму добавляются необходимые элементы управления: переключатели, кнопки выбора, линейки прокрутки и др. Элементы управления расположены на специальной панели инструментов – **Toolbox** (рис. 1.2). Если панель элементов управления не отображена на экране, следует выполнить команду **Вид – Панель элементов (View-Toolbox)**.



Рис. 1.2. Панель элементов управления

Каждый из объектов и элементов управления в среде VBA обладает определенными свойствами (имя, цвет, размер, надпись и др.), которые можно изменять в окне свойств **Properties** (рис. 1.3). Если окно свойств не отображено на экране, необходимо выполнить команду **Вид – Окно свойств** (или нажать клавишу **F4**).

Рассмотрим некоторые свойства объектов класса *UserForms* среды программирования VBA.

1) **Форма** – главный объект визуального интерфейса, на котором размещаются другие объекты (кнопки-переключатели, текстовые поля, командные кнопки и др.).

Основные свойства формы:

- **Name (Имя)** – задает имя объекта (*UserForm1* по умолчанию);
- **Caption (Надпись)** – текст, выводимый в качестве заголовка будущего приложения;
- **BackColor** – цвет фона формы (по умолчанию – серый);
- **Enabled** – указывает, доступна ли форма (содержит значение типа *Boolean*).

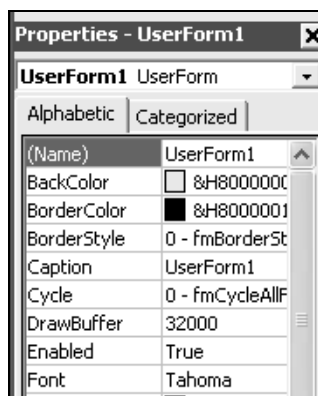



Рис. 1.3. Окно свойств


2) Объект Label (Метка)

С помощью объекта **Label**  на форме можно разместить некоторый поясняющий текст (комментарий), который не может быть изменен пользователем в процессе работы приложения.

Основные свойства объекта Label:

- **Name (Имя)** – имя объекта, по умолчанию *Label1*.
- **Caption** – содержит текст надписи;
- **AutoSize** – автоподбор ширины по размеру надписи;
- **Font** – используется для установки параметров шрифта.

3) Объект управления и контроля TextBox (Текстовое поле)

Объект **TextBox**  размещает в форме поле (окно), предназначенное для ввода текстовой или числовой информации.

Основные свойства объекта TextBox:


- **Name (Имя)** – имя объекта, по умолчанию *TextBox1*.
- **Text (Текст)** – главное свойство, позволяющее редактировать содержимое текстового окна;
- **Font** – используется для установки параметров шрифта.
- **ForeColor** – определяет цвет шрифта.

4) Объект Command Button (Командная кнопка)

Элемент управления **Command Button**  активизирует выполнение некоторой команды (программного кода).


Основные свойства объекта Command Button:


- **Name (Имя)** – имя объекта, по умолчанию `CommandButton1`.
- **Caption** – задает текст надписи на кнопке.

5) Объект **Image (Изображение)**  применяется для отображения в определенном участке формы графических объектов, например, «скриншотов» с условием задачи.

Файл с сохраненным изображением открывается в свойстве **Picture** объекта `Image`. Во время выполнения программы для загрузки изображения можно также использовать функцию `LoadPicture`:

```
Image1.Picture = LoadPicture("ИмяФайла")
```

6) Объект **OptionButton (Переключатель)**  создает в форме переключателя (кнопки выбора) для выбора режима работы или настроек выполнения программы.

7) Объект **ListBox (Окно списка)**  применяется для отображения прокручиваемого списка с некоторыми элементами.

При разработке пользовательской формы необходимо последовательно выполнить следующие этапы:

1. **Этап проектирования (визуальное программирование)** – разработка графического интерфейса пользователя: создание окна Windows-приложения, размещение необходимых элементов управления на форме, установка свойств объектов управления.
2. **Этап программирования** – ввод программного кода для обработки определенных событий для заданных объектов.
3. **Тестирование и отладка приложения** (проверка его работоспособности).
4. **Сохранение приложения.**

1.2. ПРИМЕРЫ РАЗРАБОТКИ ПОЛЬЗОВАТЕЛЬСКОЙ ФОРМЫ ДЛЯ РЕШЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ

Пример 1. *Разработать интерактивное приложение для вычисления площади кольца.*

Порядок работы:

1. Создать пользовательскую форму (рис. 1.4).
2. Написать программный код, выполняющий следующие функции:
 - пользователь вводит в текстовые поля значения радиусов кольца;
 - при нажатии командной кнопки «**Площадь кольца**» в четвертом текстовом окне выводится результат вычислений;
 - при нажатии командной кнопки «**Очистка**» текстовые окна очищаются;
 - при нажатии командной кнопки «**Выход**» приложение завершает работу.

Расчетная формула: $S = \pi \cdot (R1^2 - R2^2)$,

где $R1$ и $R2$ – значения внешнего и внутреннего радиусов кольца.

Решение

Первая часть - визуальное программирование

1. Запустите редактор VBA. Добавьте пользовательскую форму командой **Insert – UserForm**. Переименуйте форму в соответствии с рисунком 1.4, присвоив свойству **Caption** значение «**Вычисление площади кольца**».

2. Нанесите на форму два объекта **Image**, разместив в одном из них – рисунок кольца, а в другом – условие задачи.

Примечание. Для вставки рисунков в объект *Image* необходимо выполнить следующие действия:

1) вставьте скриншот с условием задачи в графический редактор **Paint** (или нарисуйте необходимый рисунок); измените размер окна Paint по размеру рисунка;

- 2) сохраните рисунок в формате *.jpeg*;
- 3) в свойстве *Picture* объекта **Image** укажите (откройте) файл с сохраненным рисунком.
3. Нанесите на форму три **Метки**, присвоив свойству **Caption** – значения «**Вычислить площадь кольца**», «**Внешний радиус**», «**Внутренний радиус**», соответственно; свойству **Font** – шрифт **Arial**, размер **12**, полужирный.

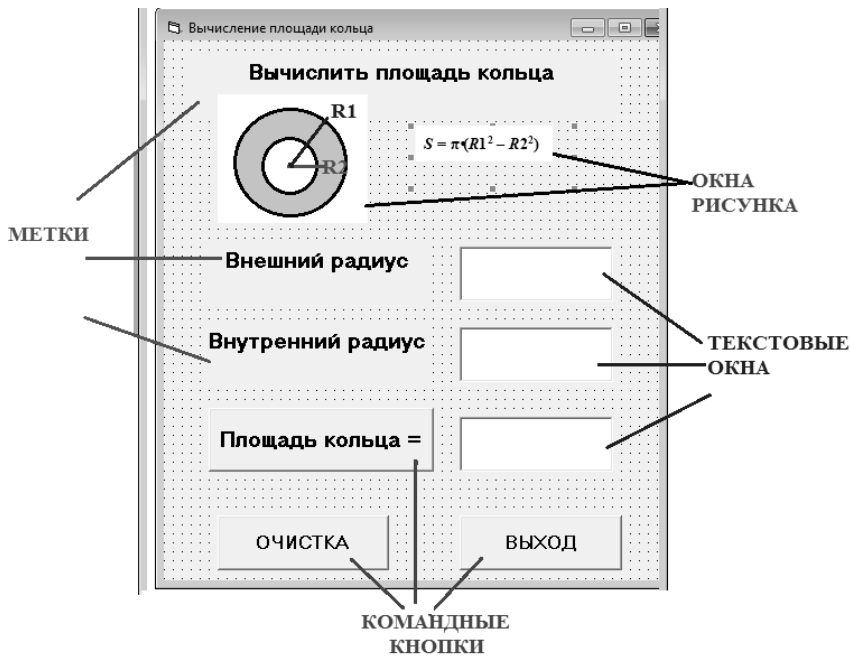


Рис. 1.4. Пользовательская форма для приложения «**Вычисление площади кольца**»

4. Нанесите на форму три **Текстовых окна**; измените свойство **Font** (шрифт), выбрав шрифт **Arial**, размер **10**, полужирный.
5. Нанесите на форму три **Командные кнопки**, изменив их размеры и свойство **Caption** на «**Площадь кольца =>**», «**Очистка**» и «**Выход**», соответственно (рис. 1.4).

Вторая часть – написание кода программы

1. Выполните двойное нажатие по командной кнопке «Площадь кольца =>» и введите программный код:

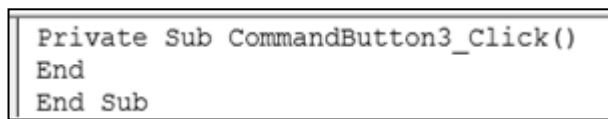
```
Private Sub CommandButton1_Click()  
R1 = Val(TextBox1.Text) 'внешний радиус  
R2 = Val(TextBox2.Text) 'внутренний радиус  
S = 4 * Atn(1) * (R1 * R1 - R2 * R2) '4*atn(1)= число пи  
TextBox3.Text = S 'вывод значения площади кольца  
End Sub
```

Примечание. В процедуре обработки данного события необходимо применить встроенную функцию *val*, чтобы присвоить переменным R1 и R2 значения введенных пользователем в текстовые поля значения радиусов кольца.

2. Выполните двойное нажатие по командной кнопке «Очистка» и введите программный код:

```
Textbox1.Text = "" 'две кавычки подряд  
Textbox2.Text = ""  
Textbox3.Text = ""
```

3. Выполните двойное нажатие по командной кнопке «Выход» и введите команду **End** (рис. 1.5):



```
Private Sub CommandButton3_Click()  
End  
End Sub
```

Рис. 1.5. Программный код для кнопки «End»

4. Запустите приложение на выполнение командой **Run - Start** и протестируйте его (рис. 1.6).

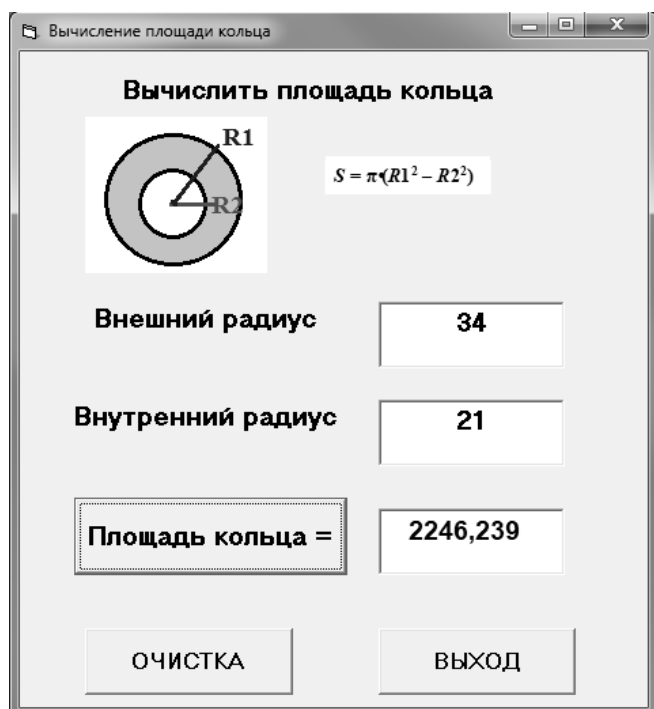


Рис. 1.6. Готовое приложение «Вычисление площади кольца»

ВАРИАНТЫ ЗАДАНИЯ 1

Разработать интерактивное приложение для вычисления указанных величин (таблица 1.1).

В отчете отразить основные этапы разработки интерактивного приложения с графическим интерфейсом:

- условие задания с расчетными формулами,
- пользовательскую форму с элементами управления,
- программный код,
- результат работы готового приложения.

Таблица 1.1

№	Вычисляемая величина	Исходные данные
1	Образующая конуса $l = \sqrt{R^2 + H^2}$	Радиус основания R , высота H
2	Площадь боковой поверхности конуса $S = \pi \cdot R \cdot L$	Радиус основания R , образующая конуса L
3	Площадь боковой поверхности цилиндра $S = 2 \cdot \pi \cdot R \cdot H$	Радиус основания R , высота H
4	Площадь поверхности конуса $S = \pi \cdot R \cdot (R + L)$	Радиус основания R , образующая конуса L
5	Площадь поверхности цилиндра $S = 2 \cdot \pi \cdot R \cdot (R + H)$	Радиус основания R , высота H
6	Объем цилиндра $V = \pi \cdot R^2 \cdot H$	Радиус основания R , высота H
7	Объем конуса $V = 1/3 \cdot \pi \cdot R^2 \cdot H$	Радиус основания R , высота H
8	Объем сферы $V = 4/3 \cdot \pi \cdot R^3$	Радиус сферы R
9	Площадь поверхности сферы $S = 4 \cdot \pi \cdot R^2$	Радиус сферы R
10	Площадь треугольника $S = 1/2 \cdot a \cdot b \cdot \sin c$	Стороны a, b ; угол между ними c (в радиан.)
11	Площадь сектора круга $S = \frac{R \cdot L}{2}$	Радиус круга R , длина дуги L
12	Площадь трапеции $S = \frac{(a + b) \cdot h}{2}$	Нижнее основание a , верхнее основание b , высота трапеции h
13	Объем пирамиды $V = 1/3 \cdot S \cdot h$	Площадь основания S , высота h
14	Объем правильного параллелепипеда $V = a \cdot b \cdot c$	Длины ребер a, b, c
15	Площадь поверхности правильного параллелепипеда $S = 2(a \cdot b + a \cdot c + b \cdot c)$	Длины ребер a, b, c

Пример 2. Разработать пользовательскую форму для вычисления значений переменной y :

$$y = \begin{cases} x^2, & \text{если } x \leq 0 \\ x+1, & \text{если } x > 0 \end{cases}$$

Порядок работы:

1. Создать пользовательскую форму (рис. 1.7).
2. Написать программный код, выполняющий следующие функции:

- пользователь вводит в текстовое поле значение переменной x ;
- при нажатии командной кнопки «**Результат**» во втором текстовом окне выводится результат вычислений;
- при нажатии командной кнопки «**Очистка**» текстовые окна очищаются.

Решение

Первая часть - визуальное программирование

1. Запустите редактор VBA. Добавьте пользовательскую форму командой **Insert – UserForm**. Переименуйте форму в соответствии с рис. 1.7, присвоив свойству **Caption** значение «**Вычисление значений переменной**».

2. Нанесите на форму объект **Image**, разместив в нем условие задачи.

3. Нанесите на форму две **Метки**, присвоив свойству **Caption** значения « x » и « y » соответственно; свойству **Font** – шрифт **Arial**, размер **12**, полужирный.

4. Нанесите на форму два **Текстовых окна**; измените свойство **Font** (шрифт), выбрав шрифт **Arial**, размер **10**, полужирный.

5. Нанесите на форму две **Командные кнопки**, изменив свойство **Caption** на «**Результат**» и «**Очистка**» (рис. 1.7).



Рис. 1.7. Пользовательская форма для приложения «Вычисление значений переменной»

Вторая часть – написание программного кода

1. Выполните двойное нажатие по командной кнопке «Результат» и введите программный код:

```
Private Sub CommandButton1_Click()  
x = Val(TextBox1.Text)  
If x <= 0 Then  
y = x * x  
Else  
y = x + 1  
End If  
TextBox2.Text = y 'вывод значений переменной y  
End Sub
```

2. Выполните двойное нажатие по командной кнопке «Очистка» и введите программный код:

```
Textbox1.Text = ""  
Textbox2.Text = ""
```

3. Запустите приложение на выполнение командой **Run - Start** и протестируйте его при различных значениях переменной x (рис. 1.8).

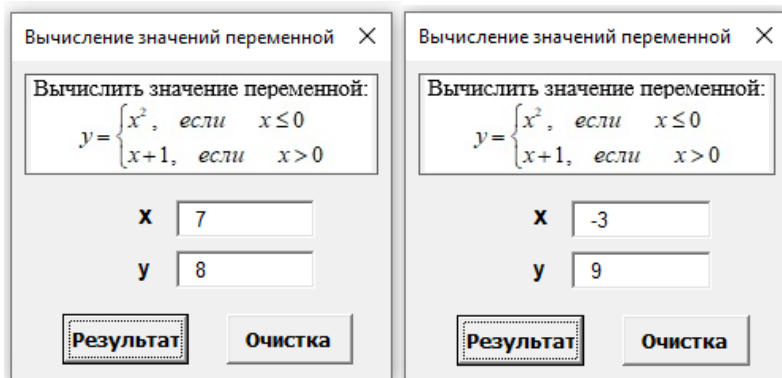


Рис. 1.8. Готовое приложение для примера 2

ВАРИАНТЫ ЗАДАНИЯ 2

Разработать интерактивное приложение для вычисления значений переменной y (таблица 1.2).

В отчете отразить основные этапы разработки интерактивного приложения с графическим интерфейсом:

- условие задания с расчетными формулами,
- пользовательскую форму с элементами управления,
- программный код,
- результат работы готового приложения.

Таблица 1.2

Вариант	Формулы для вычисления y
1	$y = \begin{cases} \frac{1+x^2}{\sqrt{1+x^4}}, & \text{если } x \leq 0 \\ \frac{2x + \sin^2 x}{2+x}, & \text{если } x > 0 \end{cases}$

Продолжение таблицы 1.2

2	$y = \begin{cases} \sqrt{1+x^2 - \cos^2 x}, & \text{если } x \leq 0 \\ \frac{x}{\sqrt[3]{e^{x+1}}}, & \text{если } x > 0 \end{cases}$
3	$y = \begin{cases} \sqrt{1+x^2}, & \text{если } x \leq 0 \\ \frac{1+x}{1+\sqrt[3]{e^{0.2x}}}, & \text{если } x > 0 \end{cases}$
4	$y = \begin{cases} \frac{3x^2}{1+x^2}, & \text{если } x \leq 0 \\ \sqrt{1+\frac{2x}{1+x^2}}, & \text{если } x > 0 \end{cases}$
5	$y = \begin{cases} \sqrt{1+ x }, & \text{если } x \leq 0 \\ \frac{1+3x}{2+\sqrt[3]{1+x}}, & \text{если } x > 0 \end{cases}$
6	$y = \begin{cases} \frac{3+\sin x}{1+x^2}, & \text{если } x \leq 0 \\ 2x^2 \cos^2 x, & \text{если } x > 0 \end{cases}$
7	$y = \begin{cases} \sqrt[3]{1+x^2}, & \text{если } x \leq 0 \\ \frac{1+x}{1+\cos^2 x}, & \text{если } x > 0 \end{cases}$
8	$y = \begin{cases} \frac{1+\cos x}{1+x^2}, & \text{если } x \leq 0 \\ x \cos x, & \text{если } x > 0 \end{cases}$
9	$y = \begin{cases} 3\sin^2 x - \cos x, & \text{если } x \leq 0 \\ \sqrt{2+x^2}, & \text{если } x > 0 \end{cases}$

Окончание таблицы 1.2

10	$y = \begin{cases} 3\sin x - \cos^2 x, & \text{если } x \leq 0 \\ 3\sqrt{1+x^2}, & \text{если } x > 0 \end{cases}$
11	$y = \begin{cases} \sin x - 2\cos x, & \text{если } x \leq 0 \\ \sqrt{1+x^2}, & \text{если } x > 0 \end{cases}$
12	$y = \begin{cases} \frac{ x }{1+x^2} e^{-2x}, & \text{если } x \leq 0 \\ \sqrt{1+x}, & \text{если } x > 0 \end{cases}$
13	$y = \begin{cases} \frac{ x }{1+x^2}, & \text{если } x \leq 0 \\ 2x^2 \cos^2 x, & \text{если } x > 0 \end{cases}$
14	$y = \begin{cases} x e^{-2x}, & \text{если } x \leq 0 \\ \frac{1}{\sqrt{1+x^2}}, & \text{если } x > 0 \end{cases}$
15	$y = \begin{cases} \frac{1 + \sin x}{1 + 2\cos x}, & \text{если } x \leq 0 \\ \sqrt{1+x}, & \text{если } x > 0 \end{cases}$

ЛАБОРАТОРНАЯ РАБОТА №2

ПРИМЕНЕНИЕ ОБЪЕКТА ПЕРЕКЛЮЧАТЕЛЬ

2.1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Элемент управления **Переключатель** (*OptionButton*) предназначен для выбора одного из нескольких взаимоисключающих значений. При выборе пользователем нужного элемента остальные элементы в группе автоматически устанавливаются в состояние «не выбран».

Основные свойства элемента Переключатель:

- **Caption** – надпись, которая выводится рядом с элементом;

- **Value** – принимает два значения: *True* – позиция выбрана (помечена точкой); *False* – позиция не выбрана.

Обычно переключатели объединяются в группу. Для этого используются элементы-контейнеры, например, объект Рамка. При удалении объекта-контейнера удаляются все элементы, содержащиеся в нем.

Элемент **Рамка** (*Frame*) используется в приложениях для создания визуальных или функциональных групп элементов, чаще всего переключателей и флажков. Основным свойством элемента Рамка является **Caption**. Оно содержит текст, который выводится в левом верхнем углу рамки и обычно является названием группы элементов. Если в качестве фона внутри рамки используется рисунок, то его необходимо указать в свойстве **Picture**.

2.2. ПРИМЕРЫ ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ПЕРЕКЛЮЧАТЕЛЕЙ

Пример 1. *Разработать интерактивное приложение с переключателями, позволяющее проверить знания студента по одному вопросу из курса информатики.*

Порядок работы:

1. Создать форму в соответствии с рисунком 2.1.

2. Написать программу, выполняющую следующие функции:

- пользователь вводит в текстовое поле свои данные (фамилию, имя, отчество);
- на вопрос «Укажите название первой в мире ЭВМ» пользователь выбирает один из трех вариантов ответа;
- при нажатии командной кнопки «Тестирование» появляется диалоговое окно с соответствующим сообщением (верный или неверный ответ);
- при нажатии командной кнопки «Выход» приложение завершает работу.

Решение

1 часть – визуальное программирование

1. Запустите редактор VBA. Добавьте пользовательскую форму командой **Insert – UserForm**. Переименуйте форму, присвоив свойству **Caption** значение «Тестирование» (рис. 2.1).

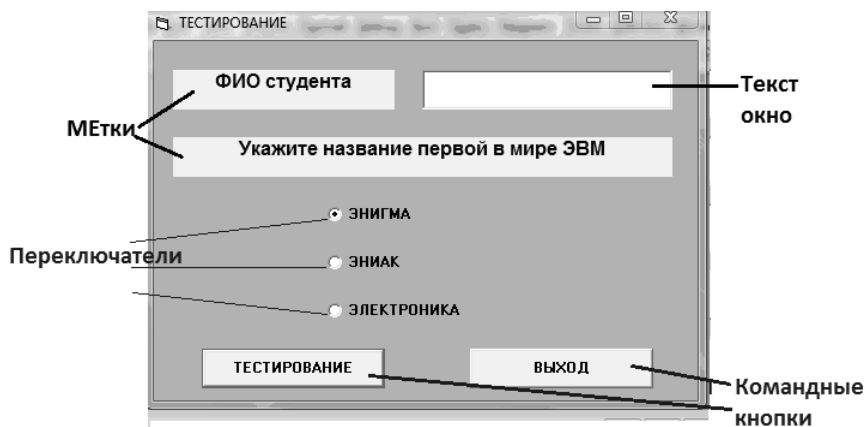


Рис. 2.1. Пользовательская форма для приложения «Тестирование»

2. Нанесите на форму две **Метки**, присвоив их свойству **Caption** значения «**ФИО студента**» и «**Укажите название первой в мире ЭВМ**» соответственно; измените свойство **Font** – шрифт **Arial**, размер **12**, полужирный.

3. Нанесите на форму **Текстовое окно**, измените свойство **Font** – шрифт **Arial**, размер **12**, полужирный.

4. Нанесите на форму **три Переключателя**, изменив их свойство **Caption** на «**ЭНИГМА**», «**ЭНИАК**», «**ЭЛЕКТРОНИКА**» соответственно.

5. Нанесите на форму две **Командные кнопки**, изменив их размеры и свойство **Caption** на «**Тестирование**» и «**Выход**» соответственно (рис. 2.1).

Вторая часть - написание кода программы

1. Дважды нажмите кнопку «**Тестирование**» и введите следующий программный код:

```
If OptionButton2.Value = True Then  
MsgBox (Textbox1.Text + ", ответ верный!")  
Else  
MsgBox (Textbox1.Text + ", ответ неверный!")  
End If
```

Примечание. При написании программного кода важно правильно указать имя объекта-переключателя с верным вариантом ответа (в данном случае – это объект **OptionButton2**).

2. Нажатие кнопки «**Выход**». В процедуре обработки этого события необходимо завершить работу программы, прописав команду **END**.

3. Запустите приложение на выполнение и протестируйте его.

Готовое приложение «**Тестирование**» показано на рис. 2.2.

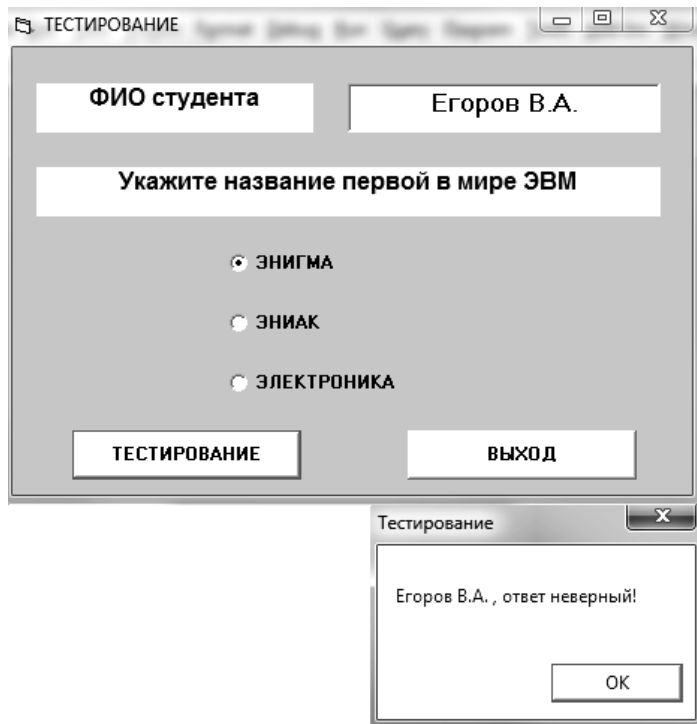


Рис. 2.2. Готовое приложение «Тестирование»

ВАРИАНТЫ ЗАДАНИЯ 1

Разработать интерактивное приложение с переключателями (кнопками выбора), позволяющее проверить знания по одному вопросу из любой предметной области.

В отчете отразить пользовательскую форму, программный код и результат работы готового приложения.

Пример 2. Разработать интерактивное приложение с контейнерами-рамками и переключателями, позволяющее оценить знания студента по трем вопросам из курса информатики.

Порядок работы:

1. Создать форму в соответствии с рисунком 2.3.

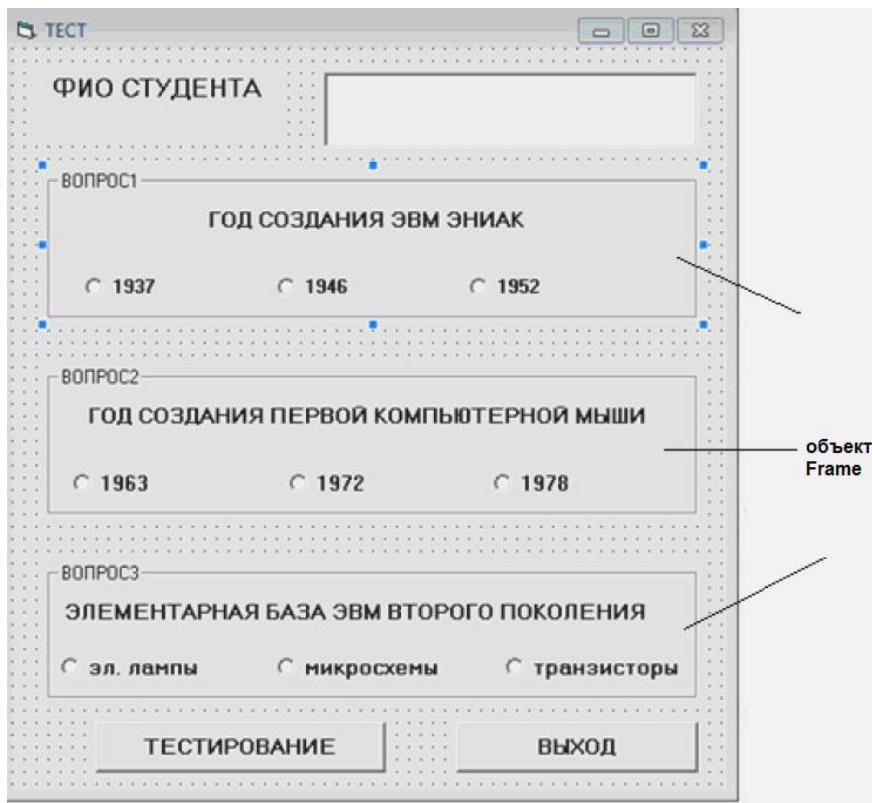


Рис. 2.3. Пользовательская форма для приложения-теста

2. Написать программу, выполняющую следующие функции:

- пользователь вводит в текстовое поле свои данные (фамилию, имя, отчество);
- пользователь выбирает один из трех вариантов ответа на три вопроса;
- при нажатии командной кнопки «Тестирование» появляется диалоговое окно с оценкой;
- при нажатии командной кнопки «Выход» приложение завершает работу.

Примечание. При разработке графического интерфейса необходимо применить три объекта **Рамка** (*Frame*), разместив в каждом из них по три переключателя.

При написании программного кода важно правильно указать имена объектов-переключателей с верными вариантами ответов (рис. 2.4).

```
Private Sub CommandButton1_Click()  
If OptionButton2.Value = True Then n1 = 1  
If optionButton4.Value = True Then n2 = 1  
If optionButton9.Value = True Then n3 = 1  
n = n1 + n2 + n3  
If n = 3 Then MsgBox (TextBox1.Text + ", Ваша оценка - 5")  
If n = 2 Then MsgBox (TextBox1.Text + ", Ваша оценка - 4")  
If n = 1 Then MsgBox (TextBox1.Text + ", Ваша оценка - 3")  
If n = 0 Then MsgBox (TextBox1.Text + ", Ваша оценка - 2")  
End Sub
```

Рис. 2.4. Программный код для приложения-теста

ВАРИАНТЫ ЗАДАНИЯ 2

Разработать интерактивное приложение с переключателями (кнопками выбора) и контейнерами-рамками, позволяющее оценить знания по трем вопросам из любой предметной области.

В отчете отразить пользовательскую форму, программный код и результат работы готового приложения.

ЛАБОРАТОРНАЯ РАБОТА №3

ПРИМЕНЕНИЕ ОБЪЕКТА СПИСОК

3.1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Объект **Список** (*ListBox*) – это визуальный элемент, позволяющий выводить на экран список значений. Во время работы приложения пользователь может выбрать из списка одно или несколько значений.

Основным свойством объекта Список является **RowSource** – определяет источник элементов списка. В качестве значения используется ссылка на диапазон рабочего листа Microsoft Excel. Например, **MyList.RowSource = "A1:A10"**.

Основной метод элемента Список – **AddItem** – добавляет элемент Строка в список.

3.2. ПРИМЕРЫ ПОЛЬЗОВАТЕЛЬСКИХ ФОРМ С ИСПОЛЬЗОВАНИЕМ СПИСКОВ

Пример 1. *Разработать пользовательскую форму, позволяющую выводить список городов, расположенных в некоторой области, выбранной с помощью переключателя.*

Порядок работы:

1. Откройте программу MS Excel.

На Листе1 в ячейке **A1** введите название **Область**, в ячейке **B1 – Город**.

В ячейку **A2** поместите название **Воронежская**, в ячейку **A17 – Курская**, в ячейку **A27 – Тамбовская**.

Ячейки **B2:B16** заполните значениями: Воронеж, Лиски, Нововоронеж, Бобров, Богучар, Борисоглебск, Бутурлиновка, Калач, Новохоперск, Острогожск, Павловск, Поворино, Россошь, Семилуки, Эртиль.

Ячейки **B17:B26** заполните значениями: Курск, Дмитриев-Льговский, Железногорск, Курчатов, Льгов, Обоянь, Рылск, Суджа, Фатеж, Щигры.

Ячейки **B27:B34** заполните значениями: Тамбов, Котовск, Жердевка, Кирсанов, Мичуринск, Моршанск, Рассказово, Уваров.

2. В редакторе VBA создайте пользовательскую форму и разместите на ней следующие элементы управления: **3 переключателя** (*OptionButton*), **рамку** (*Frame*), **надпись** (*Label*) и **список** (*ListBox*). Примерное расположение элементов управления показано на рисунке 3.1.

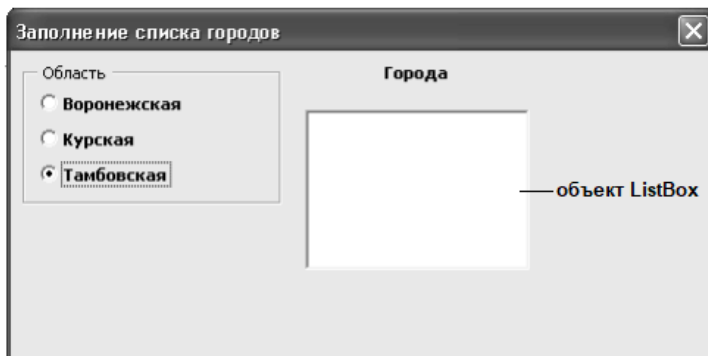


Рис. 3.1. Графический интерфейс для примера 1

3. Напишите для каждого переключателя программный код (рис. 3.2):

```
Private Sub OptionButton1_Click()  
SpCity.RowSource = "Лист1!B2:B16"  
End Sub  
  
Private Sub OptionButton2_Click()  
SpCity.RowSource = "Лист1!B17:B26"  
End Sub  
  
Private Sub OptionButton3_Click()  
SpCity.RowSource = "Лист1!B27:B34"  
End Sub
```

Рис. 3.2. Программный код для примера 1

Примечание. Для объекта Список (*Listbox*) свойству **Name** необходимо задать значение **SpCity**.

Результат работы программного кода показан на рисунке 3.3.

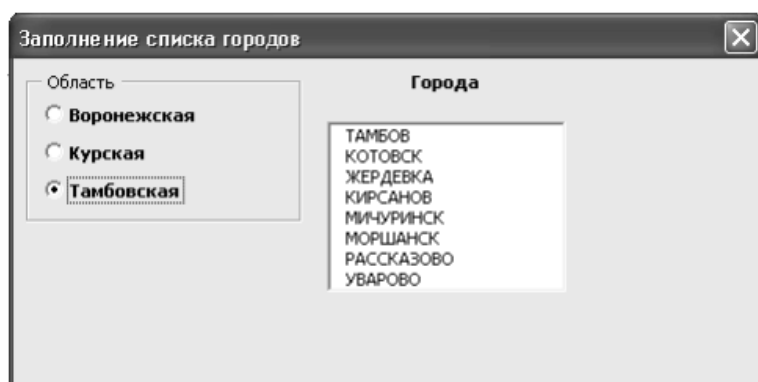


Рис. 3.3. Результат работы приложения для примера 1

ВАРИАНТЫ ЗАДАНИЯ 1

Доработайте приложение для примера 1, добавив еще одну область, состоящую из нескольких городов (Ленинградскую область, Московскую, Саратовскую или др.)

Пример 2. Разработать пользовательскую форму для вычисления значений функции $y=x^2e^{-x}\cdot\sin x$ в интервале $[-3, 2]$ с шагом 0,5.

Решение

Первая часть - визуальное программирование

1. Запустите редактор VBA. Добавьте пользовательскую форму командой **Insert – UserForm**. Переименуйте форму, присвоив свойству **Caption** значение «**Табулирование функции**» (рис. 3.4).

2. Нанесите на форму объект **Image**, разместив в нем условие задачи.

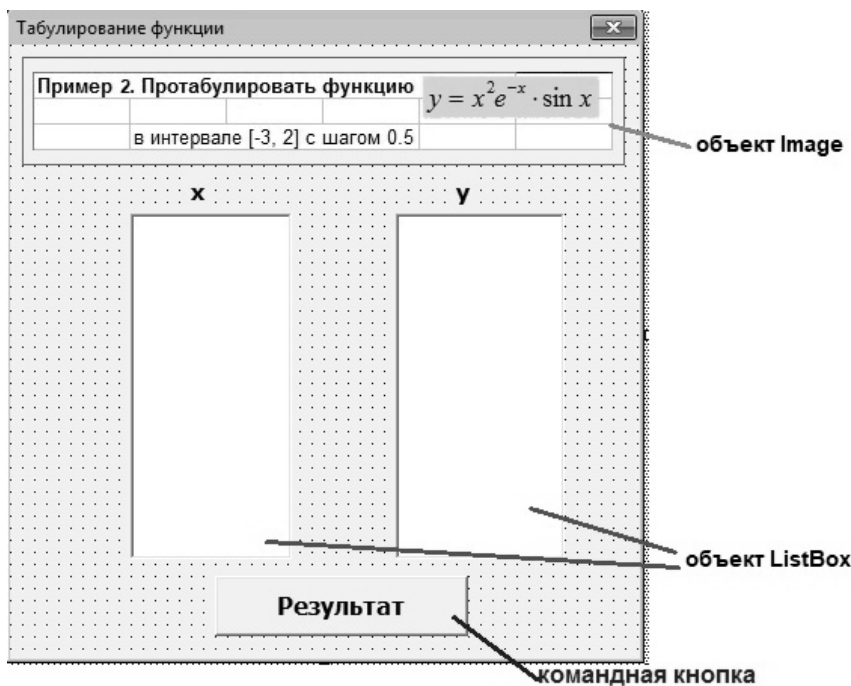


Рис. 3.4. Графический интерфейс для примера 2

3. Нанесите на форму две **Метки**, присвоив свойству **Caption** значения «x» и «y» соответственно; свойству **Font** – шрифт **Arial**, размер **12**, полужирный.

4. Нанесите на форму два объекта **ListBox**, измените их размер.

5. Нанесите на форму **Командную кнопку**, изменив свойство **Caption** на «Результат» (рис. 3.4).

Вторая часть – написание программного кода

1. Выполните двойное нажатие по командной кнопке «Результат» и введите программный код:

```

Private Sub CommandButton1_Click()
For x = -3 To 2 Step 0.5    'начало цикла
y = x ^ 2 * Exp(-x) * Sin(x)
'значения аргумента x будем выводить в объект ListBox1
ListBox1.AddItem x
'значения функции y будем выводить в объект ListBox2
ListBox2.AddItem y
Next x    'изменение значения аргумента x
End Sub

```

2. Запустите приложение на выполнение командой **Run - Start** и протестируйте его (рис. 3.5).

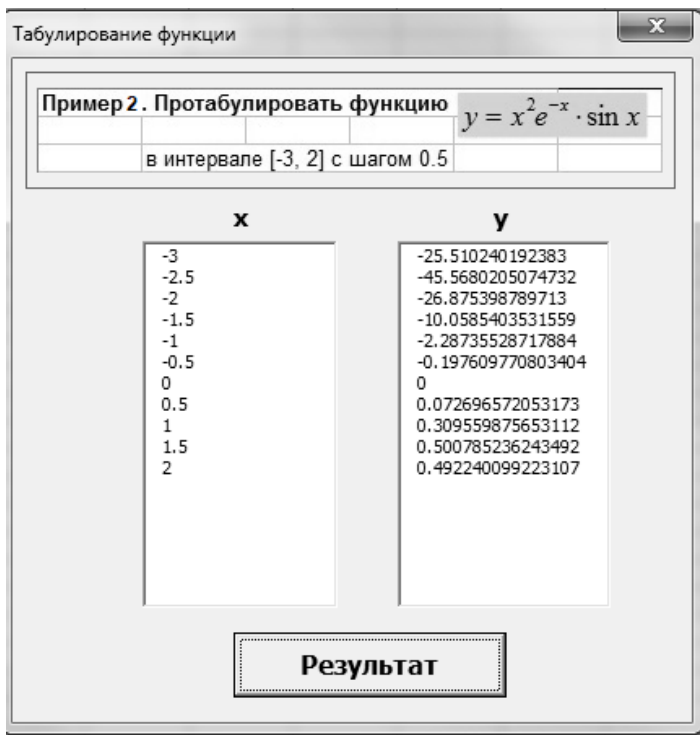


Рис. 3.5. Готовое приложение для примера 2

ВАРИАНТЫ ЗАДАНИЯ 2

Разработать интерактивное приложение для табулирования функции (таблица 3.1).

В отчете отразить основные этапы разработки интерактивного приложения с графическим интерфейсом:

- условие задания с расчетными формулами,
- пользовательскую форму с элементами управления,
- программный код,
- результат работы готового приложения.

Таблица 3.1

Вар-т	Функция	Интервал изменения аргумента $[a, b]$	Шаг изменения аргумента h
1	$y = x + \frac{4}{x + 0,5}$	[0, 6]	0,5
2	$y = 3 \cdot (x - \sin 2x)$	[-1, 4]	0,5
3	$y = (x + 2) \cdot \sin 3x$	[-2, 3]	0,5
4	$y = \frac{x - \sin 2x}{ x + 1}$	[-4, 2]	0,5
5	$y = (x + 0,5) \sin 2x$	[-2, 3]	0,5
6	$y = (x - 1) \cdot e^{-x}$	[0, 3]	0,25
7	$y = \sqrt{x} \cdot e^{-x}$	[0, 4]	0,25
8	$y = (x^2 - x) \cdot e^x$	[-3, 2]	0,5
9	$y = \frac{1 - \ln x}{1 + \ln x}$	[1, 6]	0,5
10	$y = \frac{0,5 \cdot x^2 - x + 2}{x^2 + 1}$	[-10, 12]	2
11	$y = \cos 2x - 3 \cdot \sin x$	[-2, 1]	0,25

Продолжение табл. 3.1

Вар-т	Функция	Интервал	Шаг
12	$y = \frac{x+1}{x^2+1} e^{-x}$	[-2, 3]	0,5
13	$y = 2^{- x } x$	[-1, 2]	0,25
14	$y = \sqrt[3]{ x } \sin x$	[-12, 10]	2
15	$y = \sqrt[3]{ x } \cos x$	[-5, 1]	0,5

ВАРИАНТЫ ЗАДАНИЯ 3

Разработать интерактивное приложение для табулирования функции с делением на три ветви, согласно варианту (таблица 3.2).

Интервал и шаг изменения аргумента выбрать в соответствии с условием задания.

Таблица 3.2

Вариант	Формулы для вычисления $y(x)$
1	$y = \begin{cases} e^{-x^2} - 2, & \text{если } x < -1 \\ \ln(1+x^2), & \text{если } -1 < x \leq 6 \\ \cos x^2 + 3, & \text{если } x > 6 \end{cases}$
2	$y = \begin{cases} 3, & \text{если } x \leq -4 \\ x^3 + 2x, & \text{если } -4 < x < 4 \\ 2 \cdot (x-2), & \text{если } x \geq 4 \end{cases}$
3	$y = \begin{cases} -(x+1)^3, & \text{если } x < -5 \\ 3+ x , & \text{если } -5 \leq x \leq 3 \\ (x+1)^2, & \text{если } x > 3 \end{cases}$

Продолжение табл. 3.2

Вариант	Формулы для вычисления $y(x)$
4	$y = \begin{cases} x+4, & \text{если } x < -2 \\ x^2+3, & \text{если } -2 \leq x < 1 \\ \frac{6+x}{x^3-2}, & \text{если } x \geq 1 \end{cases}$
5	$y = \begin{cases} x, & \text{если } x \leq -5 \\ -\sqrt{4+x^2}, & \text{если } -5 < x < 2 \\ \frac{x-4}{5}, & \text{если } x \geq 2 \end{cases}$
6	$y = \begin{cases} 0, & \text{если } x \leq -4 \\ -\sqrt{4+(x+2)^2}, & \text{если } -4 < x < 0 \\ \sqrt{4+(x-2)^2}, & \text{если } x \geq 0 \end{cases}$
7	$y = \begin{cases} e^x, & \text{если } x < 1 \\ x+1, & \text{если } 1 \leq x \leq 4 \\ \frac{5}{x^3}, & \text{если } x > 4 \end{cases}$
8	$y = \begin{cases} \frac{1}{x^2+1}, & \text{если } x < -2 \\ x-1, & \text{если } x \leq 2 \\ e^x, & \text{если } x > 2 \end{cases}$
9	$y = \begin{cases} 1/(x-2)^2 & \text{если } x < 1 \\ x^2+2x, & \text{если } 1 \leq x < 5 \\ 1/(x^2+3x), & \text{если } x \geq 5 \end{cases}$

Вариант	Формулы для вычисления $y(x)$
10	$y = \begin{cases} \frac{1}{x^2}, & \text{если } x < -3 \\ x^2 + 4, & \text{если } x \leq 3 \\ e^{x+1}, & \text{если } x > 3 \end{cases}$
11	$y = \begin{cases} \frac{4}{(x-2)^2}, & \text{если } x < -1 \\ x^2 + 1, & \text{если } -1 \leq x \leq 6 \\ \ln x, & \text{если } x > 6 \end{cases}$
12	$y = \begin{cases} \frac{2}{x}, & \text{если } x < -5 \\ x^2 + 3x, & \text{если } x \leq 5 \\ (x-4)^2, & \text{если } x > 5 \end{cases}$
13	$y = \begin{cases} 2 \cdot e^x + 3, & \text{если } x \leq -1 \\ \frac{4+x}{3}, & \text{если } -1 < x < 4 \\ \frac{x+1}{(x-2)^2}, & \text{если } x \geq 4 \end{cases}$
14	$y = \begin{cases} \frac{x^2+1}{x-2}, & \text{если } x < -2 \\ 2x^3+4, & \text{если } -2 \leq x \leq 4 \\ e^x+5x, & \text{если } x > 4 \end{cases}$
15	$y = \begin{cases} 2^x + 4, & \text{если } x < -4 \\ \frac{x+1}{x^2+1}, & \text{если } x \leq 4 \\ \sin x + 3, & \text{если } x > 4 \end{cases}$

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Агафонов Е.Д.* Прикладное программирование: учебное пособие / Е.Д. Агафонов, Г.В. Вашенко. – Красноярск: Сибирский федеральный университет, 2015. 112 с.
2. *Воробьева Ф.И.* Приемы программирования в среде Visual BASIC for applications MS Office / Е.С. Воробьев, Ф.И. Воробьева. – Казань: Изд-во Казан. гос. технол. ун-та, 2010. 104 с.
3. *Грошев А.С.* Информатика: учебник для вузов/ А.С. Грошев. – Москва, Берлин: Директ-Медиа, 2015. 484 с.
4. Информатика. Решение инженерных задач в среде программирования VBA: учебное пособие / Е.Н. Овчинникова, С.Ю. Кротова, Т.В. Сарапулова. – СПб.: Изд-во «ЛЕМА», 2020. 104 с.
5. *Кузьменко В.Г.* Visual Basic 6. – М.: «Бином-Пресс», 2011. 672 с.
6. *Назаров С.В.* Программирование на Visual Basic / С.В. Назаров, П.П. Мельников. – М.: «Финансы и статистика», 2010. 320 с.
7. *Слепцова Л.Д.* Программирование на VBA в Microsoft Office 2010. – М.: ООО «И.Д. Вильямс», 2010. 432 с.
8. *Ширева С.Н.* Практикум по VBA для Microsoft Excel: учебное пособие. – РГППУ, 2017. 128 с.

СОДЕРЖАНИЕ

Введение.....	3
ЛАБОРАТОРНАЯ РАБОТА №1	4
Разработка пользовательской формы для решения вычислительных задач	4
1.1. Общие сведения.....	4
1.2. Примеры разработки пользовательской формы для решения вычислительных задач	8
Варианты задания 1	11
Варианты задания 2.....	15
ЛАБОРАТОРНАЯ РАБОТА №2.....	18
Применение объекта Переключатель	18
2.1. Краткие теоретические сведения	18
2.2. Примеры приложений с использованием Переключателей ...	18
Варианты задания 1	21
Варианты задания 2.....	23
ЛАБОРАТОРНАЯ РАБОТА №3	24
Применение объекта Список.....	24
3.1. Краткие теоретические сведения	24
3.2. Примеры пользовательских форм с использованием списков.....	24
Варианты задания 1	26
Варианты задания 2.....	29
Варианты задания 3.....	30
Библиографический список.....	33