

Федеральное агентство по образованию
Государственное образовательное учреждение высшего профессионального образования
«Нижегородский государственный архитектурно-строительный университет»

Кафедра информационных систем и технологий

**ОСНОВЫ ПРОГРАММИРОВАНИЯ
на языке Microsoft Visual Basic 2005**

ЛАБОРАТОРНЫЕ РАБОТЫ

УДК 681.3

Основы программирования на языке Microsoft Visual Basic 2005

Методические указания для выполнения лабораторных работ по курсу «Информатика» для студентов всех специальностей

Нижний Новгород: ННГАСУ, 2010

В методических указаниях представлены задания и уровни их исполнения для выполнения лабораторных работ в среде разработки Microsoft Visual Basic 2005, входящей в состав Microsoft Visual Studio 2005, изучаемого в рамках курса «Информатика» студентами всех специальностей. Уровень исполнения задания студент выбирает самостоятельно в соответствии со своей теоретической и практической подготовленностью.

Составители: к. т. н., доцент Д. И. Кислицын, ст. преподаватель Ю. А. Громов

Под редакцией д. ф.-м. н., профессора А. Н. Супруна

©Нижегородский государственный архитектурно-строительный университет, 2010 г.

Введение

Microsoft Visual Basic – средство разработки программного обеспечения, включающее *язык и среду программирования*. Язык Visual Basic (VB) унаследовал дух, стиль и отчасти синтаксис своего предка – языка Basic, у которого есть немало диалектов, например QBasic. В то же время Visual Basic сочетает в себе процедуры и элементы объектно-ориентированных и компонентно-ориентированных языков программирования. Среда разработки VB включает инструменты для визуального конструирования пользовательского интерфейса.

Visual Basic зарекомендовал себя как хорошее средство быстрой разработки программ и приложений баз данных, работающих под управлением операционных систем семейства Microsoft Windows.

Первое признание серьезными разработчиками Visual Basic получил после выхода версии 3 – VB3. Версия VB6, входящая в состав Microsoft Visual Studio 6.0, стала по-настоящему зрелым и функционально богатым продуктом. После этого разработчики из Microsoft существенно изменили направление развития данной технологии. Сегодня нужно говорить о классическом Visual Basic, его диалектах (Visual Basic for Applications (VBA) и Visual Basic Scripting Edition (VBScript)) и о языке для платформы .NET – Visual Basic .NET.

В 2005 году Microsoft выпустил новую версию Visual Basic – Visual Basic 2005 в комплекте Visual Studio. Язык основан на Visual Basic .NET. В 2007 году Microsoft выпустил версию Visual Basic 2008, а в 2010 году – Visual Basic 2010, которые также были основаны на Visual Basic .NET.

Язык Visual Basic 2005 является объектно-ориентированным языком и, соответственно, оперирует терминами объектно-ориентированного программирования, например, такими как класс, экземпляр класса, метод, событие, свойство.

Классом называют совокупность определённых правил, задающих одинаковое поведение для некоторых групп объектов. В качестве *объектов* в VB рассматриваются функциональные части приложений (программ). В VB подобными объектами, например, являются экранные формы, такие элементы управления, как кнопки, надписи, текстовые поля и т. д. Практически класс может пониматься как некий шаблон, по которому могут быть созданы эти объекты, называемые *экземплярами* данного класса. Например, кнопка как разновидность элементов управления – это *класс*, а каждая конкретная кнопка со своими геометрическими размерами, надписями и т. п., помещённая на форму, – это *экземпляр* класса.

Классы характеризуются *свойствами, методами и событиями*. Кратко рассмотрим эти понятия.

Свойства в VB представляют некоторые сведения об объекте и связаны с механизмом формального описания атрибутов объекта. Каждый объект может обладать специфическими свойствами. Например, если есть *объект* класса «Кнопка», то он может иметь следующие свойства: размер, цвет, местоположение, функциональное поведение (имитация нажатия) и т. д. Свойства могут иметь как фиксированный набор значений (например, выбор, представленный в виде выпадающего списка, цвета кнопки), так и произвольный (например, размеры кнопки).

Методы представляют собой процедуры и функции (схожие с процедурами и функциями в QBasic), объявляемые внутри *классов*. Они могут быть использованы для конкретного *экземпляра* класса или для самого *класса*. Т. е. методы представляют действия, которые может выполнить объект. Например, *методом* для *объекта* класса «Кнопка» может быть процедура «Скрыть кнопку» и т. п.

События – это уведомления, получаемые объектом от других объектов или приложений или передаваемые объектом в другие объекты или приложения. События позволяют объектам выполнять действия при появлении определенных обстоятельств. Примером события могут служить «Нажатие кнопки» или «Наведение курсора мыши на кнопку». Поскольку Microsoft Windows является управляемой событиями операционной системой, то события могут поступать от других объектов и приложений или от пользователя (например, при нажатии кнопки мыши или клавиши на клавиатуре).

Все функциональные части приложения рассматриваются как *объекты*, содержащие в себе некоторые *свойства* и способные выполнять определённые *методы* и генерировать *события*.

Одними из основных понятий в объектно-ориентированном программировании являются *наследование*, *полиморфизм* и *инкапсуляция*.

Наследование позволяет передавать свойства и методы объекта от предка к потомку. Это означает, что один объект может быть построен на базе другого. При этом могут быть унаследованы свойства, методы и события. Например, в программе VB имеется объект «Кнопка», предусматривающий прямоугольную форму кнопки. Требуется создать новый объект, отличающийся только формой кнопки. В таком случае создаётся новый класс кнопок на основе уже существующего класса прямоугольных кнопок. При этом в новом классе изменяются только геометрические параметры кнопки, а все остальные свойства, методы и события этого класса остаются неизменными.

Полиморфизмом называют способность объектов разных классов использовать один и тот же метод. При этом в зависимости от объекта, использующего данный метод, будет выполняться различный набор действий. Например, при использовании метода с именем «Выбрать объект» в разных случаях будут выполняться разные действия в зависимости от того, к какому классу объект принадлежит. Так, применение метода «Выбрать объект» для объекта «Кнопка» может привести к появлению рамки вокруг этой кнопки. Применение метода с тем же именем («Выбрать объект»), используемого для объекта «Текстовое поле», приведёт к появлению мигающего курсора в этом текстовом поле.

Инкапсуляция – это свойство языка программирования, позволяющее объединить и защитить данные и код в объекте и скрыть реализацию объекта от пользователя (прикладного программиста). При этом программисту предоставляется только спецификация (интерфейс) объекта. Например, для построения кнопки четырёхугольной формы используются определённые процедуры и функции, спрятанные (инкапсулированные) в классе «Кнопка», и программисту они на прямую не доступны, таким образом, он не может «сломать» кнопку, случайно изменив её геометрию. В то же время программисту доступны все открытые свойства, например ширина и высота кнопки.

Требования к выполнению работ

При подготовке к лабораторной работе студент должен изучить соответствующие разделы лекционного курса. В ходе выполнения лабораторной работы студент должен подготовить письменный отчет, включающий:

- номер, тему, цель лабораторной работы;
- номер варианта;
- задание;
- входные и выходные данные для каждого задания;
- эскиз проектируемой формы;
- блок-схему программы;
- текст программы с комментариями.

ЛАБОРАТОРНАЯ РАБОТА № 1

Тема: Интерактивные приложения в Visual Basic.

Цель работы: Создание Windows-приложений в среде Visual Basic 2005. Знакомство со средой Visual Basic 2005. Приобретение навыков проектирования графического интерфейса.

Задание

1. Создать новый проект в среде Visual Basic 2005.
2. Составить эскиз интерактивной формы (Рис. 1).
3. Составить программу вычисления заданного выражения (Табл. 4), предусмотрев ввод исходных данных через текстовые поля интерактивной формы и отображение результирующего значения посредством поля надписи той же интерактивной формы.
4. Выполнить сборку и компиляцию программы.
5. Запустить программу на выполнение, ввести исходные данные и получить результат вычисления заданного выражения.
6. Записать полученные результаты и оформить отчёт о проделанной работе.

The sketch shows a rectangular form with a blue header bar containing the text "ВАРИАНТ № 0". Below the header, there are five input fields arranged in two rows. The first row contains "a =" and "b =", each followed by a white rectangular input box. The second row contains "x =" and "y =", each followed by a white rectangular input box. Below these, there is a "Z=" label followed by a white rectangular input box. At the bottom left, there is a blue button labeled "ВЫЧИСЛ". At the bottom right, there is a red button labeled "ВЫХОД".

Рис. 1. Эскиз интерактивной формы

П р и м е р

Требуется создать Windows-приложение, позволяющее посредством спроектированного интерфейса в виде формы обеспечить ввод исходных данных и отображение результата вычисления следующего выражения:

$$Z = \frac{\sqrt{a/b^2 + 5.68}}{(\sin x + b \cos y)^2},$$

где $a = 114.6$; $b = 53.47 \cdot 10^3$; $x = 36^0$; $y = 0.87$.

Порядок выполнения работы

1. Создание нового проекта.
 - 1.1. Запустить среду разработки MS Visual Studio 2005.
 - 1.2. Создать новый проект, используя следующие меню: *File* → *New Project*.
 - 1.3. Указать тип нового проекта: Visual Basic (Рис. 2.).

- 1.4. Указать шаблон, на основе которого создаётся проект: Windows Application.
- 1.5. Ввести имя (Name) создаваемого проекта: `WindowsApplication_N`, где N – идентификатор пользователя (например, фамилия). При необходимости кнопкой *Browse* можно изменить расположение (Location) файлов проекта на диске. Рекомендуется сохранять в папку с номером Вашей группы.
- 1.6. Для подтверждения создания проекта нажать кнопку *OK*.

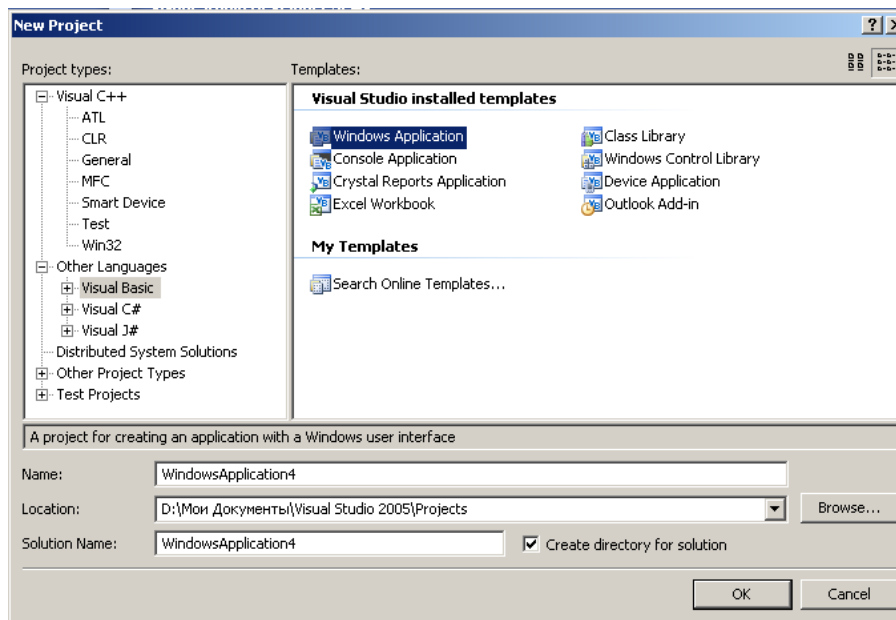


Рис. 2. Создание нового проекта

- 1.7. В результате выполнения вышеописанных действий будет создана пустая форма (Рис. 3).

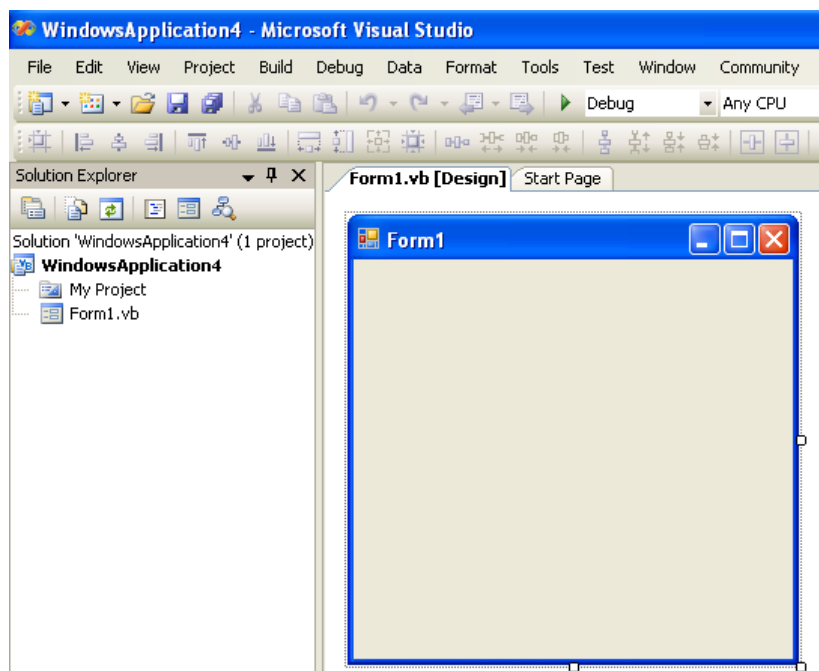


Рис. 3. Пустая форма

2. Создание эскиза интерактивной формы.

2.1. Для ввода исходных данных: a , b , x , y и вывода результата вычисления выражения $Z = \frac{\sqrt{a/b^2 + 5.68}}{(\sin x + b \cos y)^2}$ необходимо разместить на форме элементы управления, как показано на Рис. 4. Заметим, что если в Вашем варианте используется менее четырёх переменных, то и на проектируемой форме их должно быть меньше.

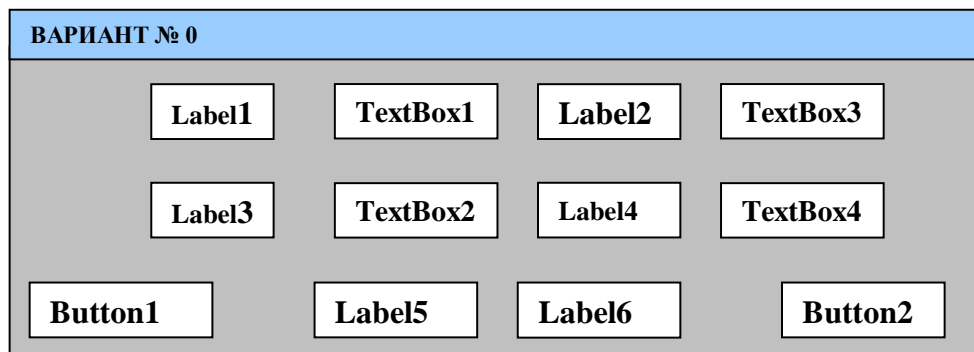


Рис. 4. Размещение элементов управления

Элементы управления *Label* (*Label1* – *Label6*) представляют собой надписи и служат для отображения текстовой информации на форме и вывода результата вычисления ($a=$, $b=$, $x=$, $y=$, $z=$, z).

Элементы управления *TextBox* (*TextBox1* – *TextBox4*) представляют собой текстовые поля и служат для ввода информации на форме (a , b , x , y).

Элементы управления *Button* (*Button1* – *Button2*) представляют собой кнопки и служат для выполнения расчёта (*Button1*) и выхода из программы (*Button2*).

Для размещения элементов управления необходимо воспользоваться панелью *Toolbox* (Рис. 5) из меню *View* → *Toolbox*, перенося мышкой необходимые элементы с панели *Toolbox* на форму.

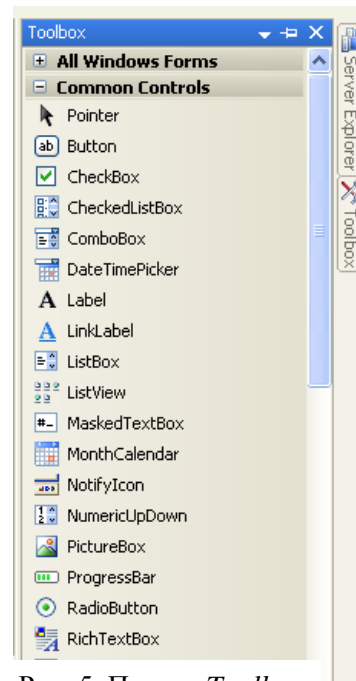


Рис. 5. Панель *Toolbox*

После размещения всех необходимых элементов управления на форме необходимо задать их свойства через панель *Properties* (Рис. 6), которая появляется после одинарного щелчка мышью на нужном элементе управления, расположенном на форме. Каждый элемент управления имеет свой набор свойств. Свойства можно назначать не только элементам управления, но и форме.

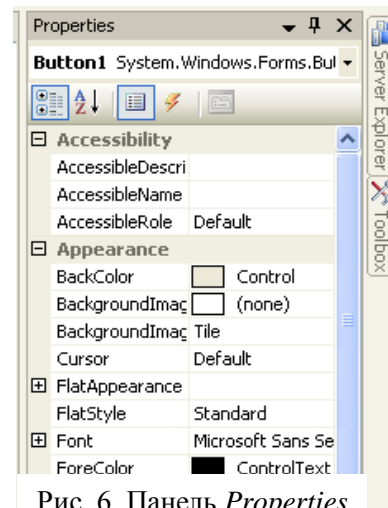


Рис. 6. Панель *Properties*

2.2. Установите значения свойств *BackColor* и *Text* объекта *Form1*, как показано на Рис.

7.

- **BackColor** = ЗЕЛЕНЬЙ

Пояснение: необходимо щелкнуть на кнопку в правом поле, затем перейти на вкладку *Custom* и выбрать из палитры цветов нужный цвет, например зеленый.

- **Text** = ВАРИАНТ № 0

Пояснение: для изменения заголовка элемента *Form1* необходимо в этом же меню *Properties* щелкнуть в правом поле свойства *Text* и вместо надписи *Form1* ввести другой текст, например «ВАРИАНТ № 0». Введенный текст будет отображаться в заголовке объекта *Form1*.

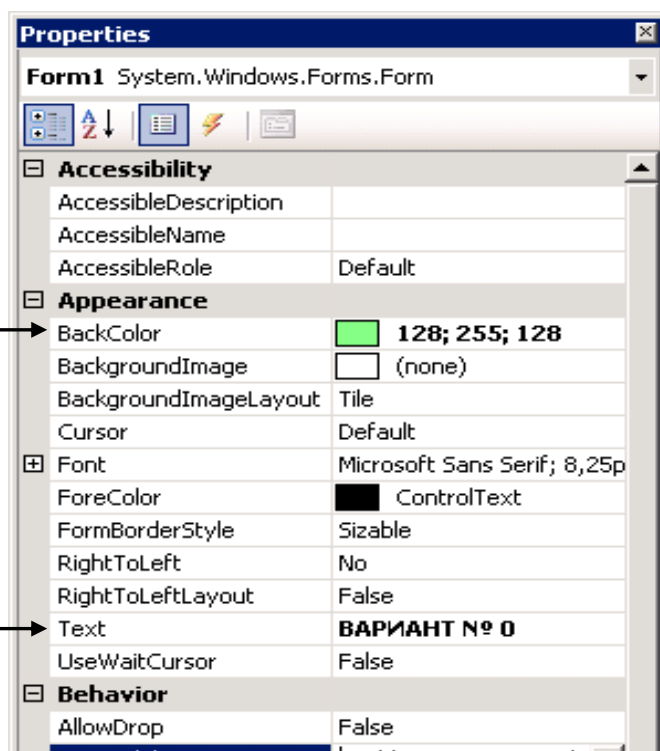


Рис. 7. Свойства *Form1*

2.3. Установите значения свойств *BackColor*, *Font* и *Text* элемента – надписи *Label1*, размещенного на форме (Рис. 8).

- **BackColor** = СВЕТЛО-ЖЕЛТЫЙ

Пояснение: необходимо щелкнуть на кнопку в правом поле, затем перейти на вкладку *Custom* и выбрать из палитры цветов нужный цвет, например светло-желтый.

- **Font** = жирный, 16 пт.

Пояснение: необходимо щелкнуть на кнопку в правом поле, затем с помощью окна настройки установить размер и шрифт текста, отображаемого в текстовом поле элемента *Label1*.

- **Text** = “a=”

Пояснение: необходимо в правом поле ввести текст надписи (например, “ a = “), размещаемой в текстовом поле элемента *Label1*.

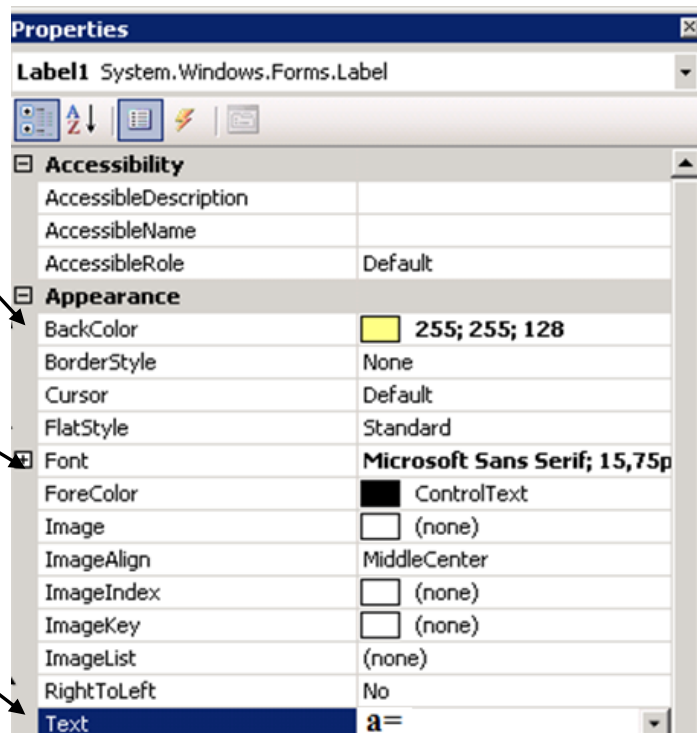


Рис. 8. Свойства *Label1*

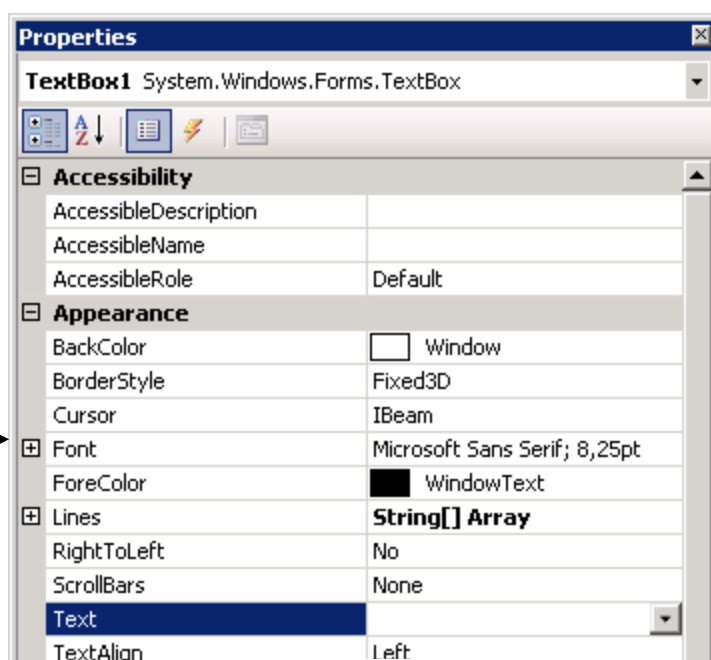
2.4. Аналогично установите для элементов *Label2*, *Label3*, *Label4*, *Label5* и *Label6* значения свойств, приведённые в Табл. 1.

Табл. 1.

Свойство	Значение
Label2.Text	b=
Label2.Font	жирный, 16 пт
Label2.BackColor	СВЕТЛО-ЖЕЛТЫЙ
Label3.Text	x=
Label3.Font	жирный, 16 пт.
Label3.BackColor	СВЕТЛО-ЖЕЛТЫЙ
Label4.Text	y=
Label4.Font	жирный, 16 пт.
Label4.BackColor	СВЕТЛО-ЖЕЛТЫЙ
Label5.Text	Z=
Label5.Font	жирный, 16 пт.
Label5.BackColor	СВЕТЛО-ЖЕЛТЫЙ
Label6.Font	жирный, 16 пт.
Label6.BackColor	СВЕТЛО-ЖЕЛТЫЙ

2.5. Установите значения свойства *Font* для элемента *TextBox1*, как показано на Рис. 9.

- **Font** = жирный, 16пт. →
- Пояснение: необходимо щелкнуть на кнопку в правом поле, затем с помощью окна настройки установить размер и шрифт текста, отображаемого в текстовом поле элемента *TextBox1*.

Рис. 9. Свойства *TextBox1*

2.6. Аналогично установите для элементов *TextBox2*, *TextBox3*, *TextBox4* значения свойств, приведённые в Табл. 2.

Табл. 2.

Свойство	Значение
TextBox2.Font	жирный, 16 пт.
TextBox3.Font	жирный, 16 пт.
TextBox4.Font	жирный, 16 пт.

2.7. Установите значения свойств *BackColor*, *Font* и *Text* для элемента *Button1*, как показано на Рис. 10.

- **BackColor = СИНИЙ** —————→
Пояснение: необходимо щелкнуть на кнопку в правом поле, затем перейти на вкладку *Custom* и выбрать из палитры цветов нужный цвет, например синий.
- **Font = жирный, 16 пт.** —————→
Пояснение: необходимо щелкнуть на кнопку в правом поле, затем с помощью окна настройки установить размер и шрифт текста, отображаемого на кнопке *Button1*.
- **Text = “ВЫЧИСЛ”** —————→
Пояснение: необходимо в правом поле ввести текст надписи (например, “ВЫЧИСЛ”), размещаемой на кнопке *Button1*.

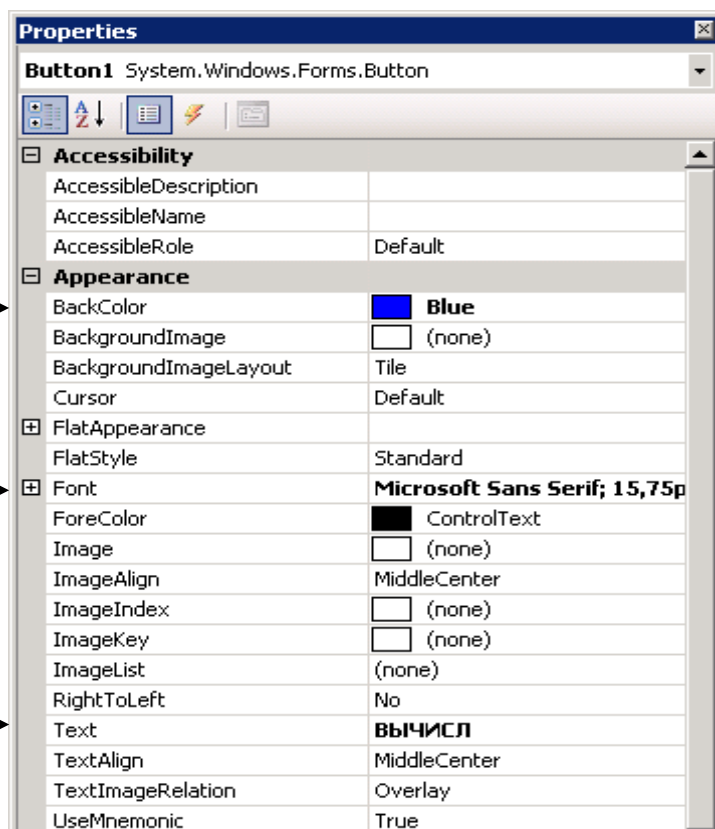


Рис. 10. Свойства *Button1*

2.8. Аналогично установите для элемента *Button2* значения свойств, приведённые в Табл. 3.

Табл. 3.

Свойство	Значение
Button2.Text	“ВЫХОД”
Button2.Font	жирный, 16 пт.
Button2.BackColor	КРАСНЫЙ

В результате изменения свойств вышеперечисленных объектов форма *Form1* примет вид, указанный на Рис. 1.

3. Написание программы (кода) включает в себя разработку кода для обработки события загрузки формы и нажатия кнопок «ВЫЧИСЛ» и «ВЫХОД».

3.1. Выполните двойной щелчок левой кнопкой мыши по пустому месту формы. В появившемся окне головного модуля *Form1.vb* будут присутствовать заголовок и концевик программы, обрабатывающей событие «ЗАГРУЗКА ФОРМЫ».

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles MyBase.Load

End Sub
```

Введите между этими строками код программы для обработки события «ЗАГРУЗКА ФОРМЫ»:

```

TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
Label6.Text = ""

```

Пояснение: с помощью введенных операторов осуществляется чистка текстовых полей в элементах *TextBox* и *Label6*.

3.2. Выполните двойной щелчок левой кнопкой мыши по кнопке «ВЫЧИСЛ». В появившемся окне головного модуля Form1.vb будут присутствовать заголовок и концевик программы, обрабатывающей событие «НАЖАТИЕ КНОПКИ «ВЫЧИСЛ».

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click

```

```

End Sub

```

Введите между этими строками код программы для обработки события:

```

Dim A, B, X, Y, X1, Z2, Z1, Z As Double
A = Val(TextBox1.Text)
B = Val(TextBox2.Text)
X = Val(TextBox3.Text)
Y = Val(TextBox4.Text)
X1 = X * 3.14159 / 180
Z2 = ( Math.Sin(X1) + B * Math.Cos(Y) ) ^ 2
Z1 = Math.Sqrt(A / B) + 5.68
Z = (Z1 / Z2)
Label6.Text = CStr(Z)

```

Пояснение:

Оператор *Dim* объявляет переменные *A, B, X, Y, X1, Z2, Z1, Z* как числовые переменные, имеющие тип числа с плавающей запятой двойной точности.

Функция *Val()* осуществляет преобразование аргумента строкового типа в числовое значение.

Функция *CStr()* осуществляет преобразование аргумента числового типа в строковое значение.

Использование в вычислениях стандартных математических функций реализуется посредством выражений, в которых указано имя класса *Math* и следующее за ним через точку имя стандартной функции (например, *Sin*).

3.3. Выполните двойной щелчок левой кнопкой мыши на кнопку «ВЫХОД». В появившемся окне головного модуля Form1.vb будут присутствовать заголовок и концевик программы, обрабатывающей событие «НАЖАТИЕ КНОПКИ «ВЫХОД».

Введите между этими строками код программы для обработки события.

```

Application.Exit()

```

Пояснение: для класса *Application* осуществляется вызов встроенного метода *Exit()*.

4. Сборка и компиляция модулей проекта выполняется командой **Build WindowsApplication** из меню *Build* (Рис. 11).

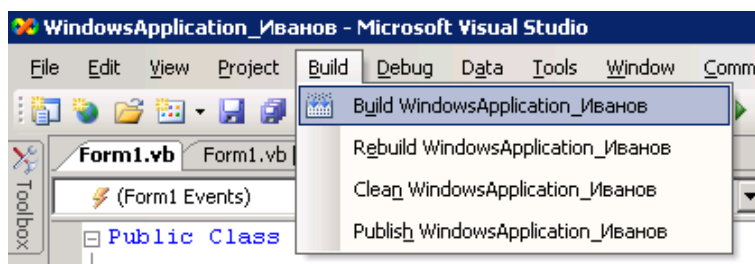


Рис. 11. Сборка и компиляция приложения

Убедитесь, что приложение откомпилировалось без ошибок (с сообщением **Build: 1 succeeded or up-to-date, 0 failed, 0 skipped**), в противном случае проверьте правильность написания кода.

Сохраните Ваш проект командой *Save All* из меню *File*.

5. Запустить приложение на выполнение можно командой *Start Debugging* из меню *Debug* (Рис. 12).

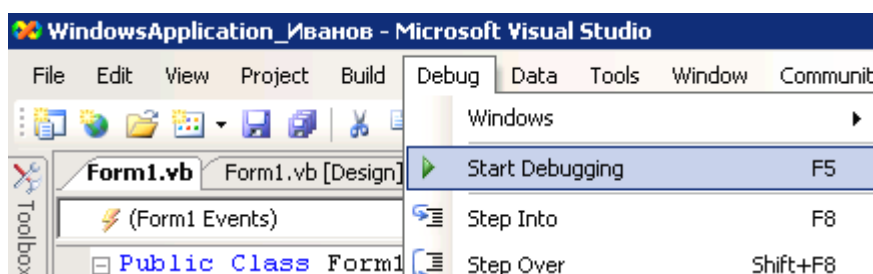


Рис. 12. Запуск программы

После запуска приложения на выполнение автоматически появляется интерактивная форма с размещенными на ней элементами управления (Рис. 1.).

5.1. Введите с клавиатуры в соответствующие поля на форме значения исходных данных, указанных в Вашем варианте ($a = 114.6$; $b = 53\,470$; $x = 36$; $y = 0.87$).

5.2. Для выполнения вычислений нажмите кнопку «ВЫЧИСЛ».

Результат расчета появится на форме в поле надписи *Label6*. Сеанс вычислений можно повторять многократно для различных значений исходных данных.

Для завершения работы приложения необходимо нажать кнопку «ВЫХОД».

Для выхода из программного комплекса Microsoft Visual Studio 2005 необходимо использовать команду *Exit* из меню *File*.

Табл. 4. Варианты к лабораторной работе № 1

№ задания	Выражение	Значения переменных
1	$Y = \sqrt[4]{a^3} + \frac{3 \cos 35^\circ + \operatorname{arctg} 3^\circ}{\ln(a+10) - (b-x)^3} + \sqrt{b-x} - 7.128$	$a=3.17$ $b=5.34$ $x=4.1$
2	$Y = \frac{4 \sin 35^\circ + \arccos 0.2}{(a-b)^2 + \ln(x+9)} + \sqrt{x} - \sqrt{a-b} + 0.29$	$a=8.31$ $b=4.50$ $x=2.71$
3	$Y = \sqrt[5]{x^3} + \frac{5 \sin 34^\circ + \operatorname{tg} 34^\circ}{\ln(x+10) - (a-b)^3} + \sqrt{a-b} - 4.023$	$a=8.31$ $b=4.50$ $x=4.412$
4	$Y = \sqrt[4]{(a-b)^3} + \frac{2 \arcsin 0.9 + 5 \cos 28^\circ}{\ln(a-b) + \sqrt{0.9}} - 6.493$	$a=5.34$ $b=4.10$
5	$Y = \frac{2 \arccos 0.1 - \sin 33^\circ}{\sqrt{a} + \ln a} \sqrt[6]{a^5} + \operatorname{tg} 33^\circ - 2.201$	$a=2.412$
6	$Y = \sqrt[3]{a^2} + \frac{2 \cos x + \operatorname{arctg} 10^\circ}{\ln(a^4 + 10) + \sqrt{a}} - 1.671$	$a=2.412$ $x=29^\circ$
7	$Y = \sqrt[4]{a^3} + \frac{4 \cos x + \operatorname{arctg} 3}{\ln(a+10) - (b-4.1)^3} + \sqrt{b-4.1} - 9.767$	$a=2.71$ $b=5.34$ $x=33^\circ$
8	$Y = \frac{2 \sin x + \arccos 0.2}{(a-4.5)^2 + \ln(b+9)} \sqrt[3]{b} - \sqrt{a-4.5} + 2.214$	$a=7.31$ $b=2.17$ $x=21^\circ$
9	$Y = \sqrt[5]{a^2} + \frac{2 \arccos 0.8 + \sin x}{\ln(a^2 + 10) - \sqrt{0.8}} - 1.665$	$a=3.115$ $x=51^\circ$
10	$Y = \frac{2 \arcsin 0.8 - \cos x}{\sqrt{a} + \ln(a^2 + 1)} - \sqrt[3]{a^3} + \operatorname{tg} 44^\circ + 1.83$	$a=4.115$ $x=44^\circ$
11	$Y = \sqrt[3]{a^2} + \frac{2 \sin 73^\circ + \operatorname{arctg} 9}{\ln(a^3 + 5) + \sqrt{a}} - 3.313$	$a=8.133$
12	$Y = \sqrt[3]{a^2} + \frac{4 \sin x + \operatorname{arctg} 0.3}{\ln(a^2 + 10) - \sqrt{a-4}} - 4.875$	$a=8.71$ $x=78^\circ$
13	$Y = \frac{2 \cos 37^\circ + \arcsin 0.8}{(a-2.7)^2 - \ln(b+10)} \sqrt[3]{b} - \sqrt{a-2.7} + 3.67$	$a=8.735$ $b=2.41$
14	$Y = \sqrt[6]{a^5} + \frac{2 \arcsin 0.2 - 5 \cos x}{\ln(a^2 + 6) - \sqrt{0.2}} - 1.015$	$a=3.891$ $x=64^\circ$
15	$Y = \frac{4 \operatorname{arctg} 8 + 5 \sin x}{\ln(a^2 + 1) - \sqrt{8}} + \sqrt[3]{a} - 5.601$	$a=8.735$ $x=22^\circ$
16	$Y = \sqrt[4]{a^3} - \frac{2 \arcsin 0.13 + \sqrt{a}}{5 \operatorname{tg} x - \ln(a^2 + 10)} - 5.882$	$a=12.13$ $x=14^\circ$

17	$V = \frac{U^4 \sqrt{ 3x-y } + 2.8 \operatorname{arctg} y^{\frac{3}{2}} - \cos 3x}{(3x-y^3)^4 - U \cdot \sin 51.3^\circ} + 1.7$	$u=2.95 \cdot 10^4$ $y=0.88$ $x=1.22 \cdot 10^3$
18	$Y = \frac{a \ln 3x^{\frac{2}{3}} + b^3 \sin 2x - \sqrt[6]{3x^{\frac{2}{3}}}}{\operatorname{atg} 2x + 3x^{\frac{2}{3}} - b^4 } + 1.794$	$a=6.35$ $b=1.5 \cdot 10^2$ $x=0.56$
19	$Z = \frac{l \operatorname{tg}(2x^2 - 0.25) + m \sin^2 52.1^\circ}{(2x^2 - 0.25)^{\frac{4}{3}} + lmx} + 1.538$	$l=2.7 \cdot 10^3$ $m=6.54$ $x=0.82$
20	$R = \frac{a \sqrt{b^2 - 0.8 \sin 3.2x} + \operatorname{tg}(0.8 \sin 3.2x)}{\ln(b^2 - 0.8 \sin 3.2x) + e^a} - 1041.29$	$a=3.42$ $b=152 \cdot 10^2$ $x=14.5^\circ$
21	$P = \frac{a \sin^2(2x) + \sqrt{a^2 2b}}{\ln(a^2 + 2b) + a^{-2b}} + a^2 - 39.248$	$a=4.78$ $b=2.83 \cdot 10^2$ $x=35.5^\circ$
22	$M = \frac{le^{-0.45x} + \sqrt{0.45x} + n \cos(3y + 15^\circ)}{e \operatorname{tg}(3y + 15^\circ) - n \ln 0.45x} - 4595.752$	$l=2.8 \cdot 10^4$ $n=1.45$ $x=2.2$ $y=8.2^\circ$
23	$T = \frac{2p \sin^3(x/4) - \sqrt{4p^2 + q} \cdot \operatorname{tg}(x/4)}{\ln(4p^2 + q) + q \cos x - 2.78} + 125.833$	$p=3.7 \cdot 10^3$ $q=-21.2$ $b=2.17$ $x=132^\circ$
24	$Z = \frac{a^2 \sqrt{b + 0.45 \cos 2.7x} + \operatorname{tg}(0.45 \cos 2.7x)}{(b + 0.45 \cos 2.7x)^{\frac{4}{3}} + e^{-b}} - 14179.162$	$a=-1.5 \cdot 10^2$ $b=0.98$ $x=15.2^\circ$
25	$W = \frac{u^3 e^{2x-1} - v \operatorname{tg}(2x-1)}{u^3 \ln 2x + v \cos 41.2^\circ} + 1.709$	$u=2.72$ $v=1.045 \cdot 10^3$ $x=0.24$
26	$P = \frac{\sqrt{m^2 - 2.83 \cdot 10^{-2} \cdot n^{\frac{3}{2}} + 0.48m^2}}{\sin(2.83 \cdot 10^{-2} \cdot n^{\frac{3}{2}}) + m^2 \cdot \cos 28.3^\circ} + 2.404$	$m=5.48$ $n=0.75 \cdot 10^2$ $q=2.8$
27	$S = \frac{0.52p^2 + 4.2 \cdot 10^{-3} e^{2q} - (\cos 0.52m)^2}{\sqrt{0.52p^2 + 4.2 \cdot 10^{-3} e^{2q} + 753 \cdot e^{2q}}} - 195.553$	$p=2.8 \cdot 10^2$ $q=0.35$ $m=37.2^\circ$
28	$W = \frac{473.2U^2 - 0.82V^{-0.54} + \cos U^3}{\sqrt{473.2U^2 - 0.82V^{-0.54} + \sin U^3}} - 5979.319$	$U=2.75 \cdot 10^2$ $V=1.54$
29	$r = \frac{\sqrt{p^{0.32} \cdot \operatorname{tg}^2 0.75 \cdot q + e^{0.75q}}}{622.7 \cdot 10^2 \cdot p^{0.32} + 0.75 \cdot q} + 2.9$	$p=5.75$ $q=1.3$
30	$q = \frac{pe^{0.25x} + \sqrt{0.25x} - \operatorname{tg}^2 2y}{p^{\frac{3}{4}} \ln(0.25x + 7.3) + 2.2p^{\frac{3}{4}}} - 1.538$	$p=1.3 \cdot 10^3$ $x=4.75$ $y=23.2^\circ$

ЛАБОРАТОРНАЯ РАБОТА № 2

Тема: Windows приложение в Visual Basic: опрос-шутка.

Цель работы: познакомиться с некоторыми возможностями работы с мышью и диалоговыми окнами.

Задание 1-го уровня

1. Создать новый проект.
2. Составить эскиз интерактивной формы - опроса (Рис. 1).
3. Задать значения свойств элементов управления, размещенных на интерактивной форме.
4. Для каждого элемента управления написать программный код, соответствующий событию активизации (нажатия) элемента управления, так, чтобы при наведении курсора мыши на кнопку «ДА» кнопка «убегала» (меняла своё местоположение на форме), а при выборе кнопки «НЕТ» появлялось сообщение: «Мы рады, что Вы считаете, что в повышении размера стипендии необходимости нет!»
5. Осуществить сборку и компиляцию модулей проекта.
6. Проверить работоспособность приложения.

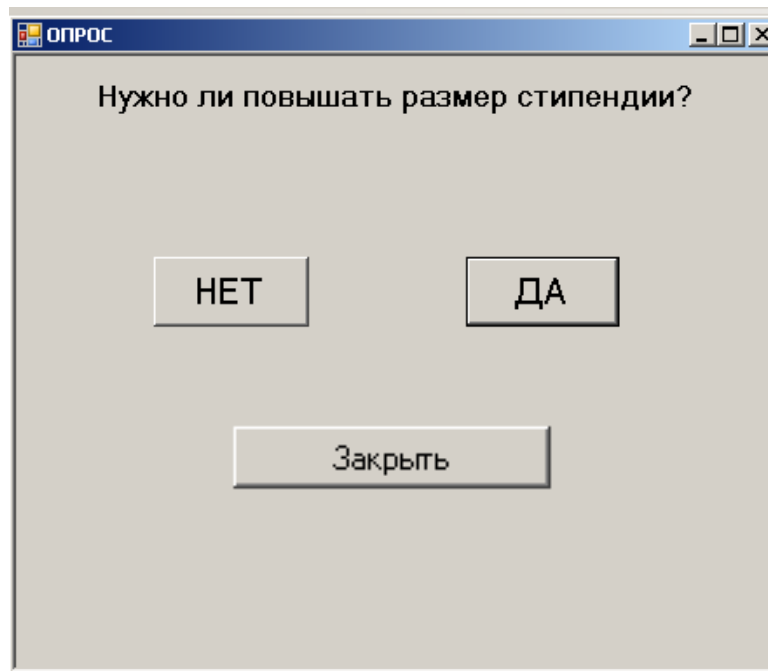


Рис. 1. Эскиз интерактивной формы

Задание 2-го уровня

Доработайте приложение из первого задания так, чтобы при наведении курсора мыши на кнопку «НЕТ» эта кнопка становилась в 2 раза больше, а при «снятии» курсора мыши с кнопки возвращалась к прежним размерам, а также чтобы при «снятии» курсора мыши с кнопки «ДА» эта кнопка возвращалась на прежнее место на форме.

Задание 3-го уровня

Доработайте приложение из первого и второго заданий таким образом, чтобы при каждом десятом наведении курсора мыши на кнопку «ДА» появлялось сообщение: «Поздравляем, Вы достойны повышения стипендии!».

Порядок выполнения работы (1-й уровень)

1. Создать новый проект командой *New Project* из меню *File* (порядок создания нового проекта подробно описан в лабораторной работе № 1).
2. Создать эскиз интерактивной формы.
 - 2.1. Используя панель инструментов *ToolBox*, разместить на форме элементы управления (кнопки - *Button1* ÷ *Button3* и надпись – *Label1*), как показано на Рис. 2.

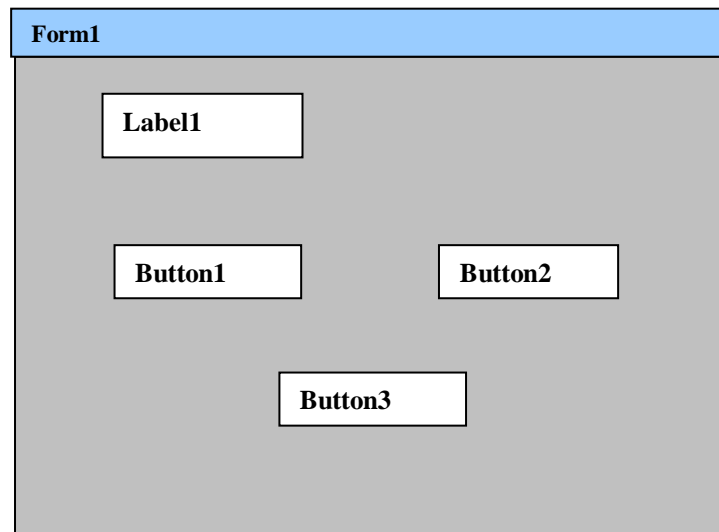


Рис. 2. Размещение элементов управления на форме

3. После размещения всех необходимых элементов управления на форме необходимо задать их свойства через панель *Properties*, которая появляется после одинарного щелчка мышью по нужному элементу управления, расположенному на форме. Каждый элемент управления имеет свой набор свойств. Свойства можно назначать не только элементам управления, но и форме.

3.1. Установите значения свойств *Cursor*, *MaximizeBox*, *Size* и *Text* объекта *Form1*, как показано на Рис. 3.

- **Form1.Cursor = Cross**
Пояснение: курсор мыши в пределах данной формы изменит свой вид со стрелки на перекрестие.
- **Form1.MaximizeBox = False**
- **Form1.Size = 450; 380**
- **Form1.Text = ОПРОС**
Пояснение: этот текст будет отображаться в заголовке формы.

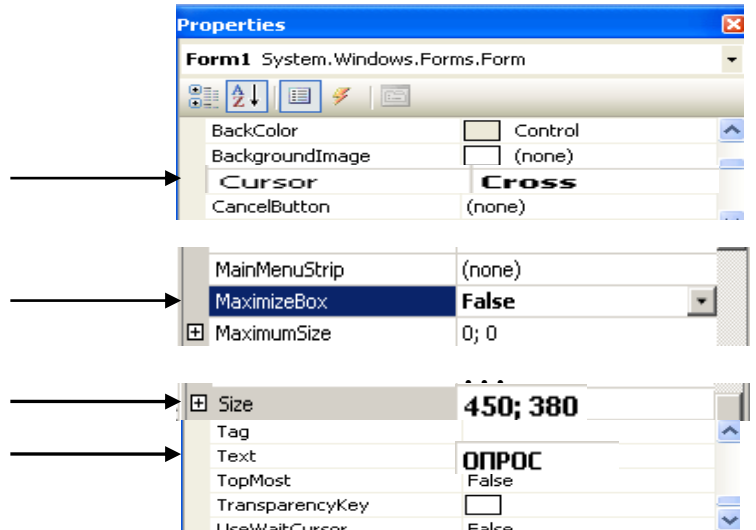


Рис. 3. Свойства Form1

3.2. Установите значения свойств элемента – надпись (*Label*), как указано в Табл. 1.

Табл. 1

Свойство	Значение
Label1.Text	Нужно ли повышать размер стипендии?
Label1.Font.Size	12
Label1.Font.Bold	TRUE
Label1.TextAlign	MiddleCenter

3.3. Установите значения свойств элементов – кнопок (*Button*), как указано в Табл. 2.

Табл. 2

Свойство	Значение
Button1.Text	НЕТ
Button1.Font.Size	15
Button2.Text	ДА
Button2.Font.Size	15
Button3.Text	Закреть

В результате изменения свойств вышеперечисленных объектов форма *Form1* примет вид, указанный на Рис. 1.

4. Написание программы (кода) включает в себя разработку кода обработки событий для кнопок «НЕТ», «ДА» и «Закреть».

4.1. Обработка события - нажатие кнопки «НЕТ». Для этого необходимо выполнить двойной щелчок левой кнопкой мыши по кнопке *Button1* и ввести код:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
```

```
    MessageBox.Show("Мы рады, что Вы считаете, что в повышении размера стипендии необходимости нет!", "Благодарим Вас за участие в опросе")
```

```
End Sub
```

Пояснение: *MessageBox* служит для вывода сообщений. В скобках указываются текст сообщения и заголовок сообщения. Также можно указать тип сообщения, иконку сообщения и ряд других параметров.

4.2. Обработка события – *наведение* на кнопку «ДА». Для этого необходимо выполнить двойной щелчок левой кнопкой мыши по кнопке *Button2* и в редакторе кода выбрать метод *MouseMove* (Рис. 4). Этот метод в отличие от метода *Click* срабатывает не на нажатие, а на наведение курсора мыши.

Введите код, описывающий реакцию приложения на наведение курсора мыши на кнопку «ДА»:

```
Private Sub Button2_MouseMove(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs) Handles Button1.MouseMove  
    Dim x, y As Single
```

```

x = Rnd()
y = Rnd()
Button1.Location = New Point(x * (Me.Width * 0.9 -
Button1.Width), y * (Me.Height * 0.9 - Button1.Height))

```

End Sub

Пояснение: свойство *Location* задаёт координаты положения кнопки *Button1* относительно верхнего левого угла формы. Чтобы кнопка не оказалась за пределами формы, нужно контролировать размеры формы по ширине (*Me.Width*) и по высоте (*Me.Height*). Из показателей размеров формы, умноженных на понижающий коэффициент 0.9, вычитаем соответствующие показатели размеров кнопки (*Button1.Width* и *Button1.Height*). Для того, чтобы кнопка перемещалась на случайную позицию, генерируем две псевдослучайных величины (X, Y), изменяющиеся в пределах от 0 до 1, и домножаем на них координаты кнопки *Button1*.

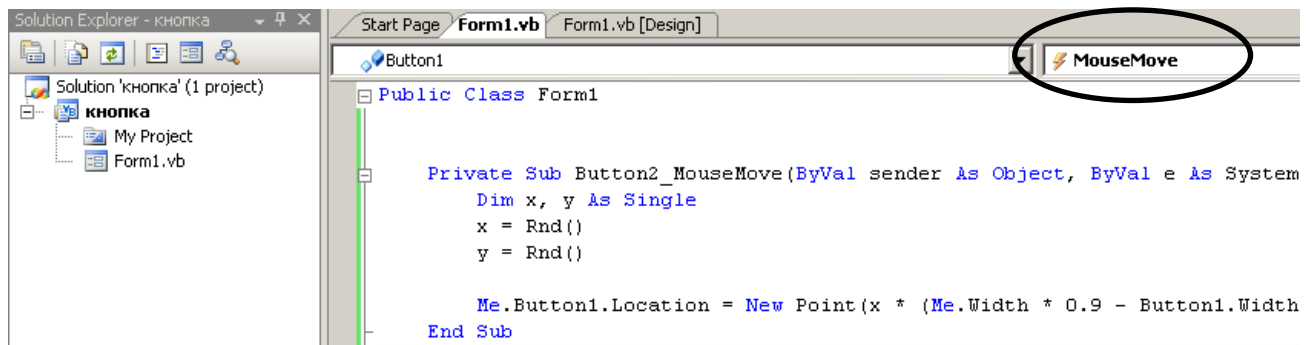


Рис. 4. Выбор метода MouseMove для Button1

4.3. Кнопка «Закреть» должна закрывать приложение. Код для обработки события – нажатия кнопки «Закреть» напишите самостоятельно.

5. Сборка и компиляция модулей проекта выполняется командой *Build WindowsApplication* из меню *Build*. Убедитесь, что приложение откомпилировалось без ошибок, в противном случае проверьте правильность написания кода.

Сохраните Ваш проект командой *Save All* из меню *File*.

6. Запустить приложение на выполнение можно командой *Start Debugging* из меню *Debug*.

Проверьте приложение на работоспособность. Покажите преподавателю результаты Вашей работы.

Для выхода из программного комплекса Microsoft Visual Studio 2005 необходимо использовать маршрут главного меню: *File* → *Exit*.

Пояснения для выполнения задания 2-го уровня

1. Для обработки событий: для наведения и снятия курсора используйте методы *MouseHover* и *MouseLeave*.

2. Размеры кнопки можно определить через свойства *.Size.Width* и *.Size.Height*.

Задать новые размеры кнопки можно используя конструкцию: *Button.Size = New Size(a, b)*, где *a* и *b* новые размеры кнопки (ширина и высота).

ЛАБОРАТОРНАЯ РАБОТА № 3

Тема: Windows приложение в Visual Basic: калькулятор.

Цель работы: создание Windows-приложения, аналогичного стандартному калькулятору Windows.

Задание 1-го уровня

7. Создать новый проект.
8. Составить эскиз интерактивной формы - калькулятора, способной выполнять простые арифметические действия (Рис. 1).
9. Задать значения свойств элементов управления, размещенных на интерактивной форме.
10. Для каждого элемента управления написать программный код, соответствующий событию активизации (нажатию) элемента управления.
11. Осуществить сборку и компиляцию модулей проекта.
12. Выполнить вычисления с помощью созданного калькулятора.

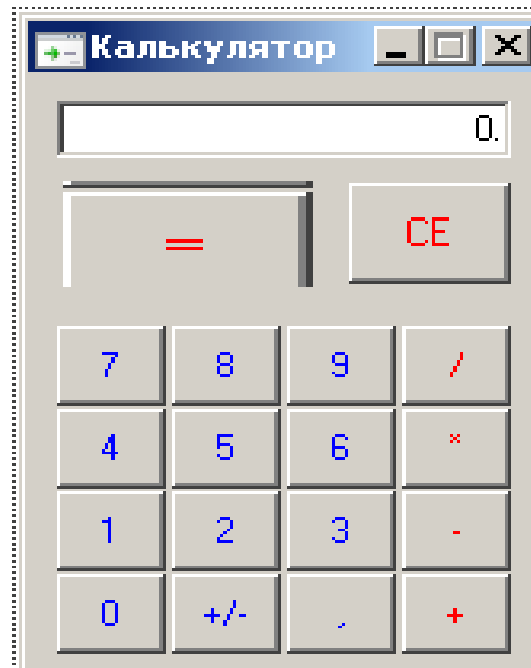


Рис. 1. Эскиз интерактивной формы

Задание 2-го уровня

Доработайте калькулятор из первого задания, добавив следующие возможности: вычисление квадратного корня, процентов, обратного числа, стирание одного символа, стирание числа (Рис. 2).

Задание 3-го уровня

Доработайте калькулятор из первого и второго заданий, добавив возможности работы с памятью: стереть память, вывести из памяти, записать в память, добавить в память (Рис. 3).



Рис. 2. Эскиз интерактивной формы (2-й уровень)

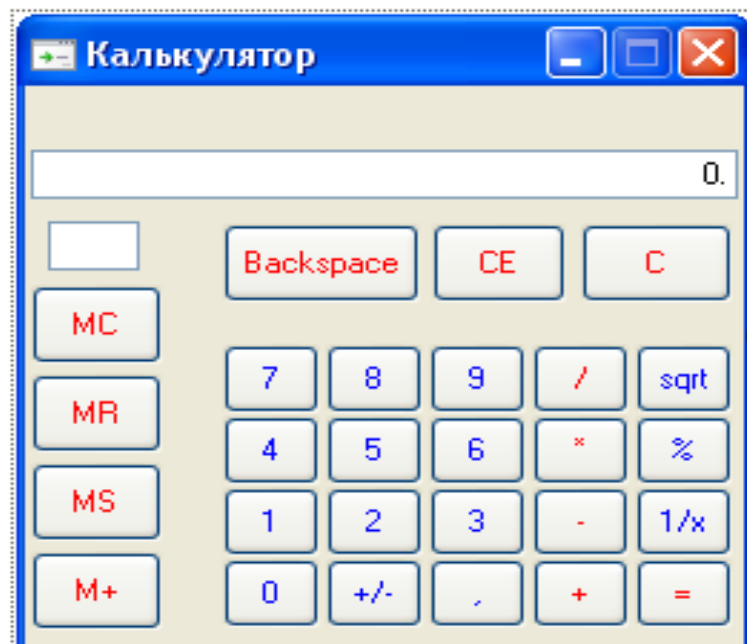


Рис. 3. Эскиз интерактивной формы (3-й уровень)

Порядок выполнения работы (1-й уровень)

4. Создать новый проект командой *New Project* из меню *File* (порядок создания нового проекта подробно описан в лабораторной работе № 1).

5. Создать эскиз интерактивной формы.

2.1. Используя панель инструментов *ToolBox*, разместить на форме элементы управления (кнопки - *Button1 ÷ Button18* и текстовое поле - *TextBox1*), как показано на Рис. 4.

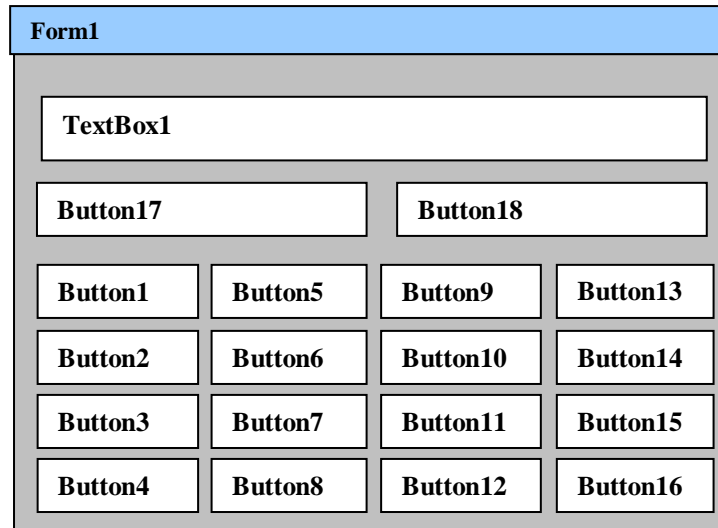


Рис. 4. Размещение элементов управления на форме

6. После размещения всех необходимых элементов управления на форме необходимо задать их свойства через панель *Properties*, которая появляется после одинарного щелчка мышью по нужному элементу управления, расположенному на форме. Каждый элемент управления имеет свой набор свойств. Свойства можно назначать не только элементам управления, но и форме.

3.1. Установите значения свойств *MaximizeBox*, *Size* и *Text* объекта *Form1*, как показано на Рис. 5.

- **Form1.MaximizeBox = False** →
- **Form1.Size = 180; 250** →
- **Form1.Text = Калькулятор** →
Пояснение: этот текст будет отображаться в заголовке формы.

Рис. 5. Свойства Form1

3.2. Установите значения свойств элемента – текстовое поле (*TextBox*), как указано в Табл. 1.

Табл. 1

Свойство	Значение
TextBox1.Name	TextBox1s
TextBox1.Text	0.
TextBox1.BorderStyle	Fixed3D

	Пояснение: щелкнуть на кнопку в правом поле, затем с помощью окна настройки установить значение данного свойства.
TextBox1.TextAlign	Right

3.3. Установите значения свойств элементов – кнопок (*Button*) как указано в Табл. 2.

Табл. 2

Свойство	Значение
Button1.Nam	bt7
Button1.ForeColor	Голубой
Button1.Text	7
Button2.Name	bt4
Button2.ForeColor	Голубой
Button2.Text	4
Button3.Name	bt1
Button3.ForeColor	Голубой
Button3.Text	1
Button4.Name	bt0
Button4.ForeColor	Голубой
Button4.Text	0
Button5.Name	bt8
Button5.ForeColor	Голубой
Button5.Text	8
Button6.Name	bt5
Button6.ForeColor	Голубой
Button6.Text	5
Button7.Name	bt2
Button7.ForeColor	Голубой
Button7.Text	2
Button8.Name	btprn
Button8.ForeColor	Голубой
Button8.Text	+/-
Button9.Name	bt9
Button9.ForeColor	Голубой
Button9.Text	9
Button10.Name	bt6
Button10.ForeColor	Голубой
Button10.Text	6
Button11.Name	bt3
Button11.ForeColor	Голубой
Button11.Text	3
Button12.Name	btpoint
Button12.ForeColor	Голубой
Button12.Text	,
Button13.Name	bt del
Button13.ForeColor	Красный
Button13.Text	/
Button14.Name	btpr
Button14.ForeColor	Красный

Button14.Text	*
Button15.Name	btmin
Button15.ForeColor	Красный
Button15.Text	-
Button16.Name	btplus
Button16.ForeColor	Красный
Button16.Text	+
Button17.Name	btis
Button17.ForeColor	Красный
Button17.Text	=
Button18.Name	btce
Button18.ForeColor	Красный
Button18.Text	CE

В результате изменения свойств вышеперечисленных объектов форма *Form1* примет вид, указанный на Рис. 1.

4. Написание программы (кода) включает в себя разработку кода для обработки событий нажатия всех кнопок.

4.1. Выполните двойной щелчок левой кнопкой мыши на пустом месте формы. В появившемся окне головного модуля *Form1.vb* выберете блок *Declarations* (Объявление), как показано на Рис. 6, и введите программный код, объявляющий переменные:

- *IsText* (для хранения содержимого текстовой строки);
- *IsNumber* (для хранения числа);
- *Point* (для указания разделителя дробной части);
- *op* (для хранения номера арифметической операции).

```
Private IsText As String
Private IsNumber As Double
Private Point As Boolean
Private op As Integer
```

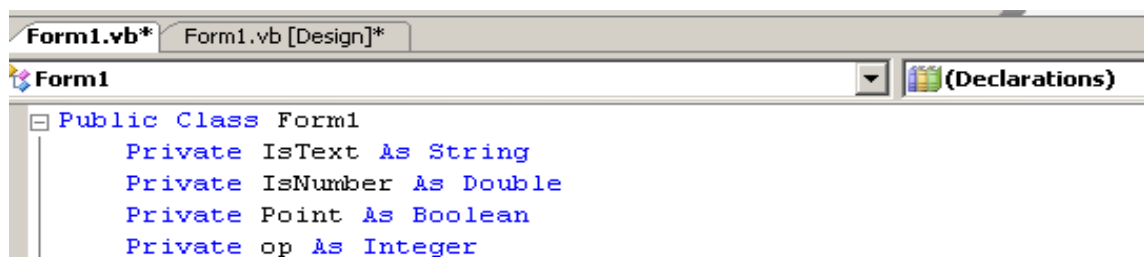


Рис. 6. Объявление переменных в блоке Form1 - Declaration

4.2. Обработка нажатия цифровых клавиш: 1, 2 ... 9, 0.

4.2.1. Введите программный код для обработки события – нажатия кнопки «1» (*bt1_Click*). Для этого необходимо выполнить двойной щелчок левой кнопкой мыши по кнопке *bt1* и ввести код:

```
Private Sub bt1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles bt1.Click

    If IsText = "0" Or IsText = "+" Or IsText = "-" Or IsText
= "/" Or IsText = "*" Then IsText = "1" Else IsText = IsText + "1"
    TextBoxIs.Text = IsText

End Sub
```


Пояснение: данный фрагмент кода сначала проверяет, не является ли вводимая цифра первой в числе и не была ли нажата клавиша арифметической операции (+, -, /, *), в этом случае вводимая цифра заменяет содержимое текстового поля (`TextBoxIs.Text`). В противном случае вводимая цифра добавляется к содержимому текстового поля (`TextBoxIs.Text`).

4.2.2. Введите программный код для обработки события – нажатия кнопки «2» (`bt2_Click`). Для этого необходимо выполнить двойной щелчок левой кнопкой мыши по кнопке `bt2` и ввести код аналогичный коду п. 4.2.1:

```
Private Sub bt2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles bt2.Click

    If IsText = "0" Or IsText = "+" Or IsText = "-" Or IsText
= "/" Or IsText = "*" Then IsText = "2" Else IsText = IsText + "2"
    TextBoxIs.Text = IsText

End Sub
```

4.2.3. По аналогии с п. 4.2.1, 4.2.2 введите программный код для обработки нажатия оставшихся цифровых кнопок (`bt3 – bt9`, `bt0`). Для экономии времени можете копировать повторяющиеся фрагменты кода.

4.3. Введите программный код для обработки события – нажатия кнопки «смена знака числа (+/-)» (`btpm_Click`). Для этого необходимо выполнить двойной щелчок левой кнопкой мыши по кнопке `btpm` и ввести код:

```
Private Sub btpm_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btpm.Click

    IsText = CStr(Val(IsText) * (-1))
    TextBoxIs.Text = IsText

End Sub
```

Пояснение: функция `Val()` преобразует текстовый тип в числовой; функция `CStr()` преобразует числовой тип в текстовый.

4.4. Введите программный код для обработки события – нажатия кнопки «запятая, отделяющая целую часть от дробной (,)» (`btpoint_Click`). Для этого необходимо выполнить двойной щелчок левой кнопкой мыши по кнопке `btpoint` и ввести код:

```
Private Sub btpoint_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btpoint.Click

    If Point = False Then
        IsText = IsText + "."
        TextBoxIs.Text = IsText
        Point = True
    End If

End Sub
```

Пояснение: данный фрагмент кода через переменную `Point` сначала проверяет, не была ли кнопка «запятая» нажата ранее при вводе текущего числа.

4.5. Обработка нажатия кнопок арифметических действий: /, *, -, +.

4.5.1. Введите программный код для обработки события – нажатия кнопки «деление (/)» (*btidel_Click*). Для этого необходимо выполнить двойной щелчок левой кнопкой мыши по кнопке *btidel* и ввести код:

```
Private Sub btidel_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btidel.Click
```

```
    IsNumber = Val(TextBoxIs.Text)
    op = 1
    IsText = "/"
    TextBoxIs.Text = IsText
    Point = False
```

```
End Sub
```

Пояснение: в данном фрагменте кода переменной *op* присваивается номер арифметической операции. При этом деление соответствует первому номеру, умножение – второму, вычитание – третьему, сложение – четвертому. Значение переменной *op* будет использоваться при вычислении результата (нажатие кнопки *btis*).

4.5.2. Введите программный код для обработки события – нажатия кнопки «умножение (*)» (*btpr_Click*). Для этого необходимо выполнить двойной щелчок левой кнопкой мыши по кнопке *btpr* и ввести код, аналогичный коду п. 4.5.1, изменив номер операции на второй:

```
Private Sub btpr_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btpr.Click
```

```
    IsNumber = Val(TextBoxIs.Text)
    op = 2
    IsText = "*"
    TextBoxIs.Text = IsText
    Point = False
```

```
End Sub
```

4.5.3. По аналогии с п. 4.5.1, 4.5.2 введите программный код для обработки событий – нажатия кнопок «вычитание (-)» (*btmin_Click*) и «сложение (+)» (*btplus_Click*), изменив соответственно номера операций и символы, их отображающие.

4.6. Введите программный код для обработки события – нажатия кнопки «=>» (*btis_Click*). Для этого необходимо выполнить двойной щелчок левой кнопкой мыши по кнопке *btis* и ввести код:

```
Private Sub btis_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btis.Click
```

```
    Select Case op
        Case 1
            IsNumber = IsNumber / Val(TextBoxIs.Text)
        Case 2
            IsNumber = IsNumber * Val(TextBoxIs.Text)
        Case 3
            IsNumber = IsNumber - Val(TextBoxIs.Text)
        Case 4
```

```

        IsNumber = IsNumber + Val(TextBoxIs.Text)
    End Select
    IsText = CStr(IsNumber)
    TextBoxIs.Text = IsText
    IsText = ""
    Point = False

```

End Sub

Пояснение: в данном фрагменте кода используется конструкция **Select Case**, позволяющая выбирать выполняемую часть кода в зависимости от значения переменной *op*.

4.7. Введите программный код для обработки события – нажатия кнопки «CE» (*btce_Click*). Для этого необходимо выполнить двойной щелчок левой кнопкой мыши по кнопке *btce* и ввести код:

```

Private Sub btce_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btce.Click

```

```

    IsNumber = 0
    IsText = "0"
    TextBoxIs.Text = IsText
    Point = False

```

End Sub

5. Сборка и компиляция модулей проекта выполняется командой *Build WindowsApplication* из меню *Build*. Убедитесь, что приложение откомпилировалось без ошибок, в противном случае проверьте правильность написания кода.

Сохраните Ваш проект командой *Save All* из меню *File*.

6. Запустить приложение на выполнение можно командой *Start Debugging* из меню *Debug*.

Выполните несколько вычислений, используя спроектированное приложение.

Покажите преподавателю результаты Вашей работы.

Для выхода из программного комплекса Microsoft Visual Studio 2005 необходимо использовать маршрут главного меню: *File* → *Exit*.

Пояснения для выполнения задания 2-го уровня

1. Для вычисления квадратного корня используйте функцию **Math.Sqrt()**.

2. Для обработки события – нажатия кнопки «Backspace» (удаление последнего введённого символа) можно использовать функцию **Remove()**, удаляющую указанное число символов в текстовой переменной начиная с указанной позиции, и свойство **Length**, возвращающее число символов в значении переменной. Например, так:

```

IsText = IsText.Remove(IsText.Length - 1, 1).

```

3. Нажатие кнопки «C» должно привести к стиранию только текущего набираемого числа, например если Вы ошиблись при вводе и хотите стереть число не посимвольно кнопкой «Backspace», а сразу целиком. Отличается от кнопки «CE» тем, что не стирает предыдущие набранные числа и операции.

Пояснения для выполнения задания 3-го уровня

В текстовом поле над кнопками по работе с памятью (M+, MS, MR, MC) должен отображаться символ «M», если в памяти содержится какое-либо число.

ЛАБОРАТОРНАЯ РАБОТА № 4

Тема: Windows приложение в Visual Basic: решение уравнения и построение графика функции.

Цель работы: создание Windows-приложения, обеспечивающего возможность решения уравнения и построения графика функции. Приобретение навыков использования графического элемента управления **PictureBox** при проектировании интерфейса Windows-приложения.

Задание 1-го уровня

1. Создать новый проект.
2. Составить эскиз интерактивной формы (Рис.1).
3. Задать значения свойств элементов управления, размещенных на интерактивной форме.
4. Составить программу для нахождения корней функции $f(x)$ на интервале $[A, B]$ с шагом E , предусмотрев ввод исходных данных через текстовые поля интерактивной формы. Функцию $f(x)$ выбрать из Табл. 6 в соответствии со своим вариантом.
5. Осуществить сборку и компиляцию модулей проекта.
6. Решить уравнение $f(x) = 0$.

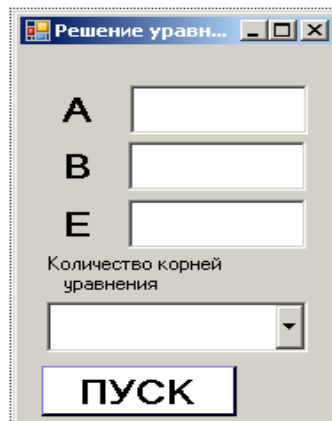


Рис. 1. Эскиз интерактивной формы

Задание 2-го уровня

Реализовать построение графика и отображение графика функции в элементе управления *PictureBox* (Рис. 2).

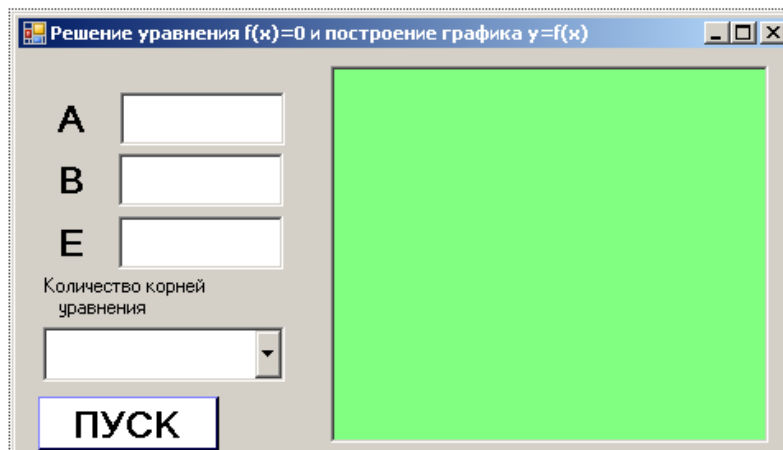


Рис. 2. Эскиз интерактивной формы с построением графика

Задание 3-го уровня

Реализовать возможность задавать пользователем функцию (полином до третьей степени), предусмотрев ввод параметров функции через текстовые поля интерактивной формы (Рис. 3).

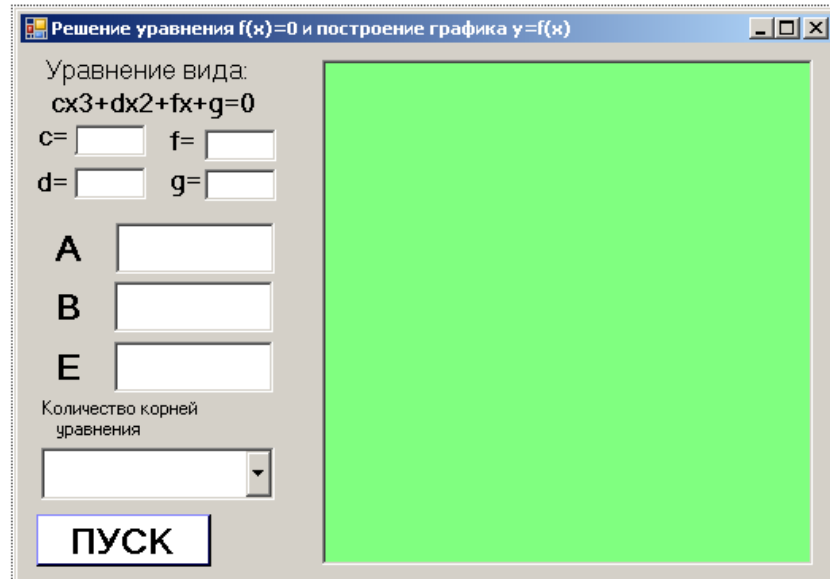


Рис. 3. Эскиз интерактивной формы с построением графика и пользовательским вводом функции

Порядок выполнения работы (1-й уровень)

1. Создать новый проект командой New Project из меню File (порядок создания нового проекта подробно описан в лабораторной работе № 1).

2. Создать эскиз интерактивной формы.

2.1. Используя панель инструментов *ToolBox*, разместить на форме элементы управления (кнопку - *Button1*, надписи – *Label1* – *Label6*, текстовые поля - *TextBox1* ÷ *TextBox3*, поле со списком - *ComboBox1* и графическое поле - *PictureBox1*), как показано на Рис. 4.

Элемент управления *ComboBox* - текстовое поле с predetermined списком значений, из которого можно выбрать одно из имеющихся значений. В данной работе в *ComboBox* будут отображаться значения вычисленных корней уравнения.

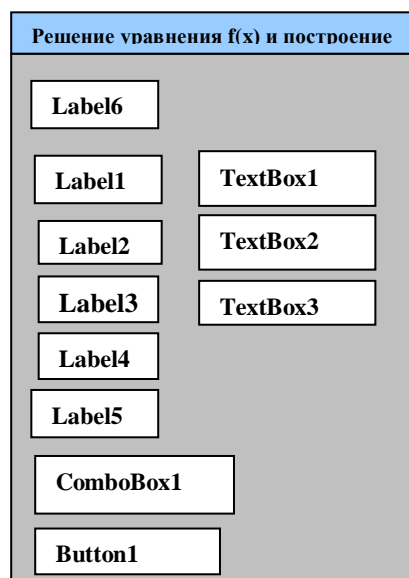


Рис. 4. Размещение элементов управления на форме

3. После размещения всех необходимых элементов управления на форме необходимо задать их свойства через панель *Properties*, которая появляется после одинарного щелчка мышью на нужном элементе управления, расположенном на форме. Каждый элемент управления имеет свой набор свойств. Свойства можно назначать не только элементам управления, но и форме.

3.1. Установите значения свойств *Text* и *WindowState* объекта *Form1*, как показано на Рис. 5.

- **Form1.Text** = Решение уравнения $f(x) = 0$ и построение графика $y = f(x)$
Пояснение: этот текст будет отображаться в заголовке формы.
- **Form1.WindowState** = Maximized
Пояснение: для отображения графика функции будет использован максимальный размер окна.

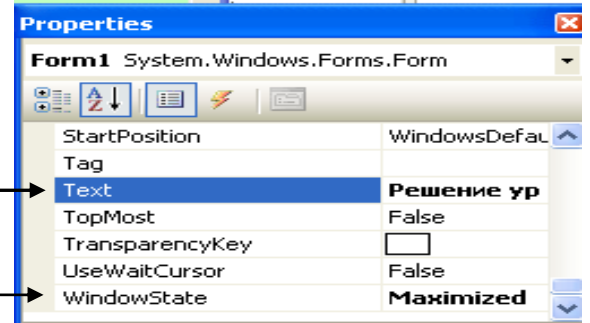


Рис. 5. Свойства Form1

3.2. Установите значения свойств элементов – надписей (*Label*), как указано в Табл. 1.

Табл. 1

Свойство	Значение
Label1.Text	A
Label2.Text	B
Label3.Text	E
Label1.Font	жирный, 16 пт.
Label2.Font	жирный, 16 пт.
Label3.Font	жирный, 16 пт.
Label4.Text	Количество корней
Label5.Text	Уравнения
Label6.Text	

3.3. Установите значения свойств элементов – текстовых полей (*TextBox*), как указано в Табл. 2.

Табл. 2

Свойство	Значение
TextBox1.Font	жирный, 16 пт.
TextBox2.Font	жирный, 16 пт.
TextBox3.Font	жирный, 16 пт.

3.4. Установите значения свойств элемента – кнопки (*Button*), как указано в Табл. 3.

Табл. 3

Свойство	Значение
Button1.BackColor	СИНИЙ
Button1.Font	жирный, 16 пт.
Button1.Text	ПУСК

Пояснение: для установки нужного цвета необходимо щелкнуть на кнопку в правом поле, перейти на вкладку Custom и выбрать из палитры цветов нужный цвет, например синий.

3.5. Установите значения свойств элемента – поля со списком (*ComboBox*), как указано в Табл. 4.

Табл. 4

Свойство	Значение
ComboBox1.DropDownStyle	DropDownList
ComboBox1.Font	жирный, 16 пт.

В результате изменения свойств вышеперечисленных объектов форма *Form1* примет вид, указанный на Рис. 1.

4. Написание программы (кода) включает в себя разработку кода для обработки событий формы и всех элементов управления. В качестве примера рассмотрим функцию $f(X) = X^2 - 2X - 10$.

4.1. Для объявления глобальных переменных выполните двойной щелчок левой кнопкой мыши на форме. В появившемся окне головного модуля *Form1.vb* выберете блок *Declarations* (Объявление), как показано на Рис. 6, и введите программный код, объявляющий переменные:

```
'Перечень глобальных переменных
Public Z As Boolean
Public A, B, Ep, MinF, MaxF As Double
'A - начальное значение аргумента X
'B - конечное значение аргумента X
'Ep - точность решения уравнения f(X)=0
'MinF - минимальное значение функции f(X)
'MaxF - максимальное значение функции f(X)
```

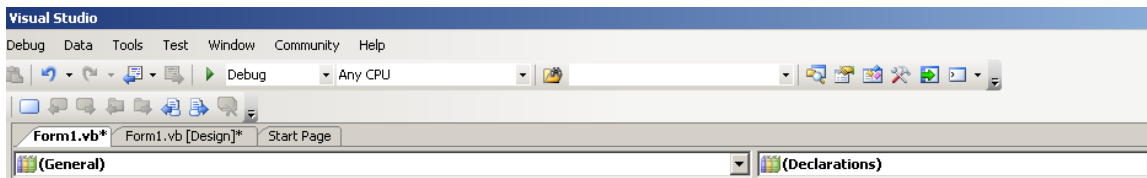


Рис. 6. Обработка события в блоке General - Declarations

4.2. Для обработки события – загрузки формы (**Form1_Load**) выберете блок Load (как показано на Рис. 7) и между строк

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
```

```
End Sub
```

введите программный код:

```
' Чистка текстовых полей исходных данных
TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
Z = 0
```

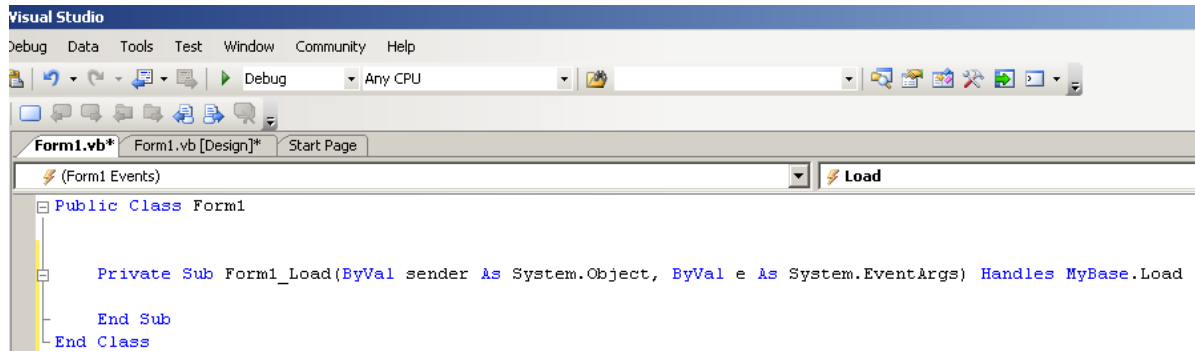


Рис. 7. Обработка события в блоке Form1 - Load

4.3. Написать программный код, обрабатывающий событие «НАЖАТИЕ КНОПКИ ПУСК» (Button1_Click).

```
Private Sub Button1_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles Button1.Click
```

```
    Dim X, X2, XC, Y1, Y2 As Double
    Dim KK As Integer
    Z = 0
    ' Проверка корректности исходных данных : если данные не
    корректны, следует выход из подпрограммы
    If TextBox1.Text = "" Or TextBox2.Text = "" Or
    TextBox3.Text = "" Then Exit Sub
    A = Val(TextBox1.Text)
    B = Val(TextBox2.Text)
    Ep = Val(TextBox3.Text)
    If A >= B Or Ep <= 0 Then Exit Sub
    If B - A < Ep Then Exit Sub
    ' Если исходные данные корректны, то устанавливается
    значение Z=1, то есть разрешается перерисовка графика
    функции f(X)
    Z = 1
    ' Устанавливаются начальные значения диапазона изменения f(X)
    MinF = func(A)
    MaxF = MinF
    ' Чистка открывающегося списка ComboBox1
    ComboBox1.Items.Clear()
    ' Чистка счетчика корней уравнения f(X)=0
    KK = 0
    ' В цикле для X от A до B с шагом Ep осуществляется
    анализ значений функции f(X)
    For X = A To B + Ep / 10 Step Ep
        Y1 = func(X)
        ' Уточняются значения диапазона изменения f(X)
        If Y1 < MinF Then MinF = Y1
        If Y1 > MaxF Then MaxF = Y1
        X2 = X + Ep
        Y2 = func(X2)
        If Y1 * Y2 <= 0 Then
            'Если выполнены условия существования корня уравнения,
            то
```



```

' уточняется значение очередного корня уравнения f(X)=0
  XC = (X + X2) / 2
' и уточненное значение корня добавляется в список
ComboBox1
  KK = KK + 1
  ComboBox1.Items.Add("X" & CStr(KK) & "= " &
  Format(XC, "0.#####"))
End If
Next X
'Значение счетчика (KK) корней отображается в поле элемента
Label5
  Label5.Text = "уравнения = " & CStr(KK)
  If MaxF < 0 Then MaxF = 0
  If MinF > 0 Then MinF = 0

End Sub

```

В качестве примера рассмотрим функцию $f(X) = X^2 - 2X - 10$.

```

' Процедура-функция, предназначенная для вычисления значения
f(X) при заданном значении аргумента X
Function func(ByVal X As Double)
  Label6.Text = "f(x)= X ^ 2 - 2 * X - 10"
  Return (X ^ 2 - 2 * X - 10)
End Function

```

5. Сборка и компиляция модулей проекта выполняется командой *Build WindowsApplication* из меню *Build*.

Сохраните Ваш проект командой *Save All* из меню *File*.

6. Запустить приложение на выполнение можно командой *Start Debugging* из меню *Debug*.

В появившейся форме (Рис. 1) ввести с клавиатуры значения исходных данных:

- **A** - начало интервала табулирования функции;
- **B** - конец интервала табулирования функции;
- **E** - шаг вычисления корней уравнения.

Для выполнения вычислений нажать кнопку ПУСК.

Покажите преподавателю результаты работы.

Для выхода из программного комплекса Microsoft Visual Studio 2005 необходимо использовать маршрут главного меню: *File* → *Exit*.

Пояснения для выполнения задания 2-го уровня

1. Для отображения графика функции можно использовать элемент управления *PictureBox*, позволяющий размещать графические примитивы (точку, отрезок, простые геометрические фигуры). Разместите элемент управления *PictureBox* на форме, как показано на Рис. 8.

2. Установите значения свойств элемента – графического поля (*PictureBox*), как указано в Табл. 5.

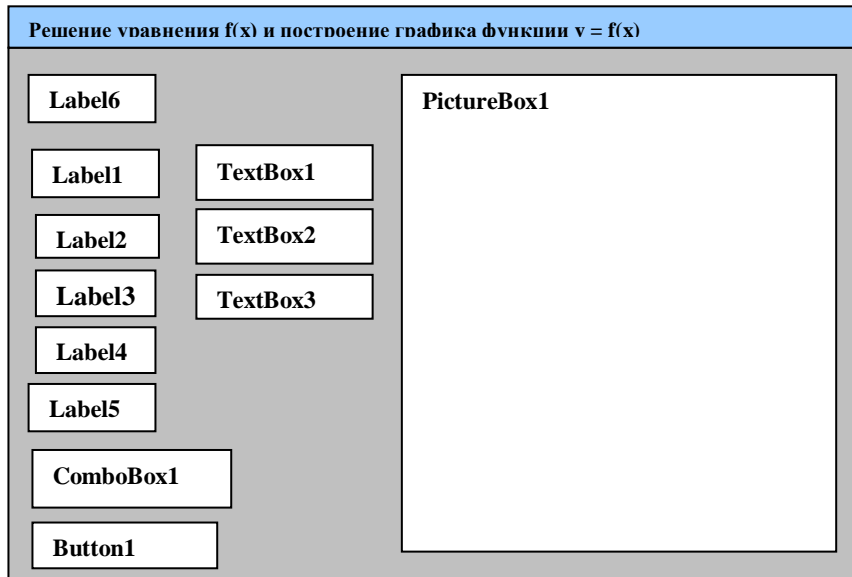


Рис. 8. Размещение элементов управления на форме

Табл. 5

Свойство	Значение
PictureBox1.BackColor	ЗЕЛЕНЬЙ
PictureBox1.BorderStyle	Fixed3D

3. Добавьте в обработчик события *Load* объекта *Form1* код, устанавливающий размеры *PictureBox*:

```
' Установка оптимальных размеров окна графического
элемента PictureBox1
PictureBox1.Width = Me.Width * 0.75
PictureBox1.Height = Me.Height * 0.9
```

4. Добавьте в обработчик события *Click* объекта *Button1* код, выполняющий перерисовку содержимого в *PictureBox*:

```
' Элементу PictureBox1 дается разрешение на перерисовку
графика функции f(X)
PictureBox1.Refresh()
```

5. Для обработки события **Paint**, возникающего при активизации графического элемента **PictureBox1**, необходимо выбрать блок **Paint** (Рис. 9).

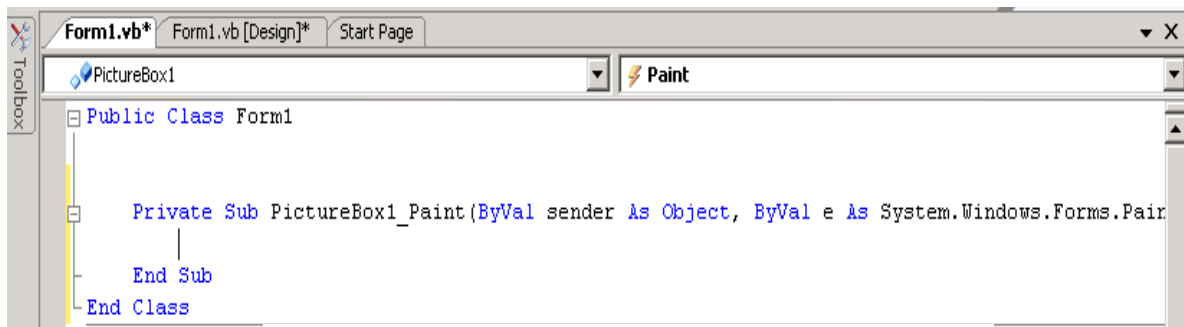


Рис. 9. Обработка события в блоке PictureBox1- Paint

И ввести код для подпрограммы **PictureBox1_Paint**:

```
Private Sub PictureBox1_Paint(ByVal sender As Object, ByVal e
As System.Windows.Forms.PaintEventArgs) Handles PictureBox1.Paint
```

```
    Dim G As Graphics = e.Graphics
    Dim X, DX, Y, DY As Double
    Dim AG, BG, NG, N, I, DW, H0, PX, PY As Integer
    ' Если есть запрет на прорисовку графика функции f(X), то
    выход из подпрограммы
    If Z = 0 Then Exit Sub
    DY = MaxF - MinF
    If DY <= 0 Then DY = 1
    ' N - количество точек для функции f(X)
    N = (B - A) / Ep
    AG = 1 : BG = PictureBox1.Width
    If A > 0 Then AG = 10
    If B < 0 Then BG = BG - 10
    ' NG - количество точек для графика функции f(X)
    NG = BG - AG
    If NG > N Then NG = N
    If NG < N Then N = NG
    ' DX - шаг изменения аргумента X для функции f(X)
    DX = (B - A) / N
    ' DW - шаг изменения аргумента X для графика функции f(X)
    DW = (BG - AG) / NG
    ' H0 - высота графика функции f(X)
    H0 = PictureBox1.Height
    ' Определяется перо для прорисовки графика функции f(X)
    Dim MyPen As New Pen(Color.Red, 3)
    ' Определяется и заполняется массив точек графика функции
    f(X)
    Dim Points(N) As Point
    For I = 0 To NG
        X = A + DX * I
        Y = func(X)
        PX = AG + (BG - AG) * (X - A) / (B - A)
        PY = (H0 - 5) * (MaxF - Y) / DY
        Points(I) = New Point(PX, PY)
    Next I
    ' Выполняется прорисовка графика функции f(X)
    G.DrawLine(MyPen, Points)
    ' Определяется перо для прорисовки координатных осей графика
    функции f(X)
    Dim MyPenXY As New Pen(Color.Blue, 3)
    ' Определяется и заполняется массив точек оси X
    Dim PointsX(2) As Point
    PX = 1
    PY = (H0 - 5) * (MaxF - 0) / DY
    PointsX(1) = New Point(PX, PY)
    PX = PictureBox1.Width
    PointsX(2) = New Point(PX, PY)
    ' Выполняется прорисовка оси X
```

```

G.DrawLine(MyPenXY, PointsX(1), PointsX(2))
' Определяется и заполняется массив точек оси Y
Dim PointsY(2) As Point
If A > 0 Then
    PX = 1
Else
    If B < 0 Then
        PX = PictureBox1.Width - 5
    Else
        PX = AG + (-A / DX) * DW
    End If
End If
PY = 0
PointsY(1) = New Point(PX, PY)
PY = H0 - 5
PointsY(2) = New Point(PX, PY)
' Выполняется прорисовка оси Y
G.DrawLine(MyPenXY, PointsY(1), PointsY(2))

```

End Sub

После сборки, компиляции и запуска приложения результат расчета в виде графика функции появится на форме в поле элемента **PictureBox1** (Рис. 10):

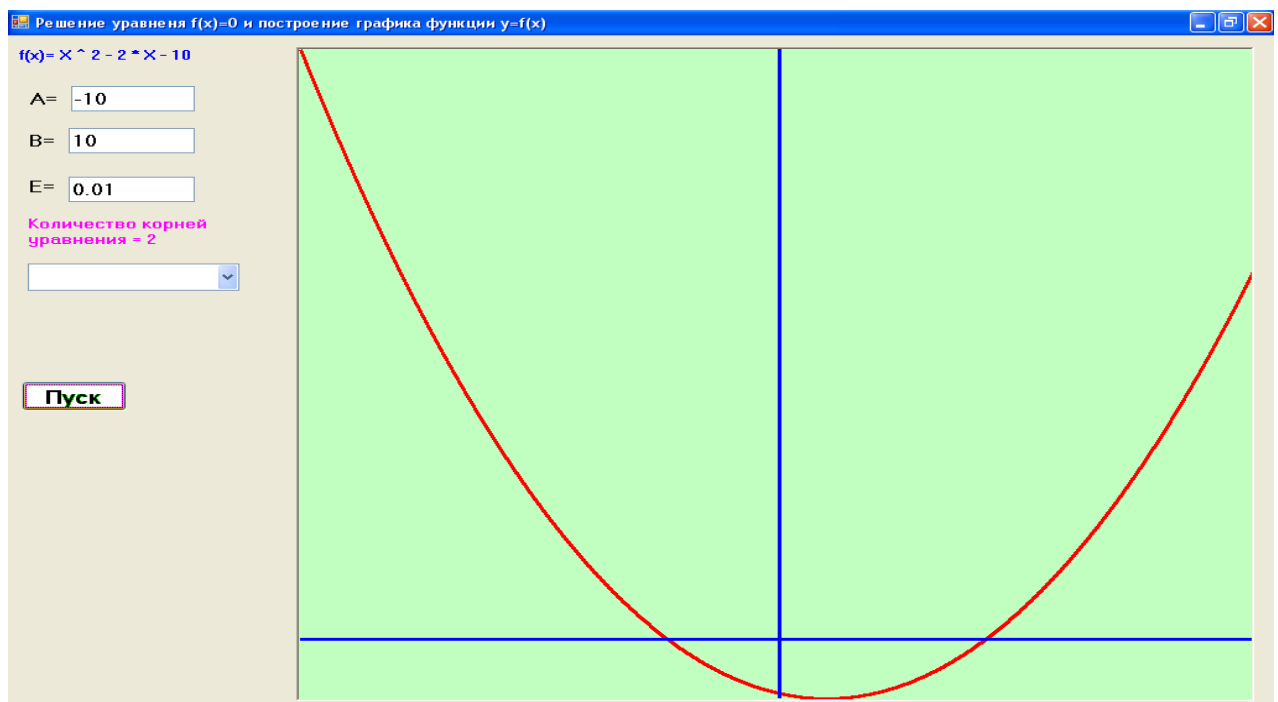


Рис. 10. Результат расчета функции $f(x) = x^2 - 2x - 10$

Выполните несколько вычислений (на разных отрезках $[A, B]$ с разным шагом E), используя спроектированное приложение.

Табл. 6. Варианты заданий к лабораторной работе №4

№ вар.	Уравнение	Отрезок [a, b]	№ вар.	Уравнение	Отрезок [a,b]
1	$e^x + x - 2 = 0$	[0;1]	16	$22x - 2^x = 0$	[0;1]
2	$\frac{1}{2} - 2\ln x = 0$	[1;2]	17	$e^x - 10x = 0$	[3;4]
3	$2^{-x} - \sqrt{x} = 0$	[0.1;1.1]	18	$e^{2x} = 2 - x^2$	[-1;0]
4	$\ln x + \sqrt{x} = 0$	[0.1;1.1]	19	$2 - x = \lg x$	[1;2]
5	$\frac{2}{x} - \ln x = 0$	[2;3]	20	$\sqrt{x} + 1 = \frac{1}{x}$	[0.1;1.1]
6	$3^{-x} - \sqrt{2x} = 0$	[0;1]	21	$2x + \ln(2x + 3) - 1 = 0$	[0;1]
7	$\sqrt{x+2} - 2^{-x} = 0$	[-2;-1]	22	$x2^x - 1 = 0$	[0;1]
8	$2\ln x - 2x = 0$	[0.1;1.1]	23	$x^3 - x - 2 = 0$	[1;2]
9	$2^{-x} - x^2 = 0$	[0;1]	24	$x^4 - 2x - 4 = 0$	[1;2]
10	$x^2 - 2\sqrt{x} = 0$	[0,1;0,5]	25	$x + 1^x = 0$	[-1;0]
11	$x^3 - \sqrt{2x} = 0$	[0,1;2]	26	$x + \ln x = 0$	[2;3]
12	$\sqrt{x} - x + 1 = 0$	[0;1.2]	27	$x^2 + \ln(l + x) - 3 = 0$	[0;1]
13	$\ln \frac{x}{2} + 2\sqrt{x} = 0$	[0,1;2]	28	$x^3 + 4\sin x = 0$	[-0,5;0,5]
14	$(x-1)^2 = 0,5e^x$	[0;1]	29	$\ln x - \sin x = 0$	[2;5]
15	$(x-1)^2 = e^{-x}$	[1;2]	30	$\sin x + 2x = 1$	[0;1]

СОДЕРЖАНИЕ

Введение.....	3
Требования к выполнению работ.....	5
Лабораторная работа № 1. Интерактивные приложения в Visual Basic.....	6
Лабораторная работа № 2. Windows приложение в Visual Basic: опрос-шутка.....	16
Лабораторная работа № 3. Windows приложение в Visual Basic: калькулятор.....	20
Лабораторная работа № 4. Windows приложение в Visual Basic: решение уравнения и построение графика функции.....	28

Дмитрий Игоревич Кислицын
Юрий Алексеевич Громов

Основы программирования на языке Microsoft Visual Basic 2005

Методические указания для выполнения лабораторных работ по курсу «Информатика» для студентов всех специальностей

Подписано к печати _____ . Бумага газетная
ф.60x84 1/16. Печать высокая. Гарнитура «Литературная». Уч.изд.л. _____ .
Усл.печ.л. _____ . Тираж _____ экз. Заказ № _____ .

Нижегородский государственный архитектурно-строительный университет
603950, Нижний Новгород, ул. Ильинская, д. 65
Полиграфический центр ННГАСУ, 603950, Н. Новгород, ул. Ильинская, д. 65