

УО «ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ЛАБОРАТОРНАЯ РАБОТА №1

**по дисциплине «Базы данных»
для специальности 1-40 01 01
«Программное обеспечение информационных технологий»**

ТЕМА: Ознакомление с интерфейсом среды программирования **Visual C#**
и основными принципами написания программ.

НОВОПОЛОЦК 2011

МАТЕРИАЛЫ ПОДГОТОВИЛА:

старший преподаватель кафедры технологий программирования
Бураченко Ирина Брониславовна

ТЕМА: Ознакомление с интерфейсом среды программирования Microsoft Visual C# и основными принципами написания программ.

ЦЕЛЬ: Научить пользователя основным принципам и приемам работы в системе программирования Microsoft Visual C#.

Результат обучения:

После успешного завершения занятия пользователь должен:

- получить общее представление о работе в системе программирования Microsoft Visual C#;
- уметь проектировать рабочую форму приложения и размещать на ней элементы управления;
- уметь сохранять элементы среды программирования.

Используемая программа: Microsoft Visual Studio 2008 (2010)

План занятия:

1. Знакомство с интерфейсом системы программирования Microsoft Visual C#.
2. Создание формы и элементов управления, ввод текста и запуск программы.
3. Задания для самоконтроля «СОЗДАНИЕ ФОРМ».
4. Практические задания для самостоятельной работы.

1. ЗНАКОМСТВО С ИНТЕРФЕЙСОМ СИСТЕМЫ ПРОГРАММИРОВАНИЯ VISUAL C#

Для создания нового проекта запустите **Microsoft Visual Studio** и выберите из меню **File** строку **New**, а затем из меню второго уровня – строку **Project**. В результате перед вами появится окно, представленное на рисунке 1.

Необходимо выбрать шаблон создаваемого проекта. Это можно сделать при помощи списка **Templates**. Мы будем создавать оконные приложения **Microsoft Windows**, поэтому выберите здесь шаблон **Windows Forms Application**. В поле **Name**, уже предложено стандартное название будущего проекта, при желании вы можете изменить название проекта. Стандартно изучение любого языка начинается с написания программы Hello World. Не будем отступать от этого правила. Изменим название проекта на "Hello World", после чего нажмем кнопку **OK**. Через некоторое время проект будет создан, а окно **Visual Studio** примет вид, показанный на рисунке 2.

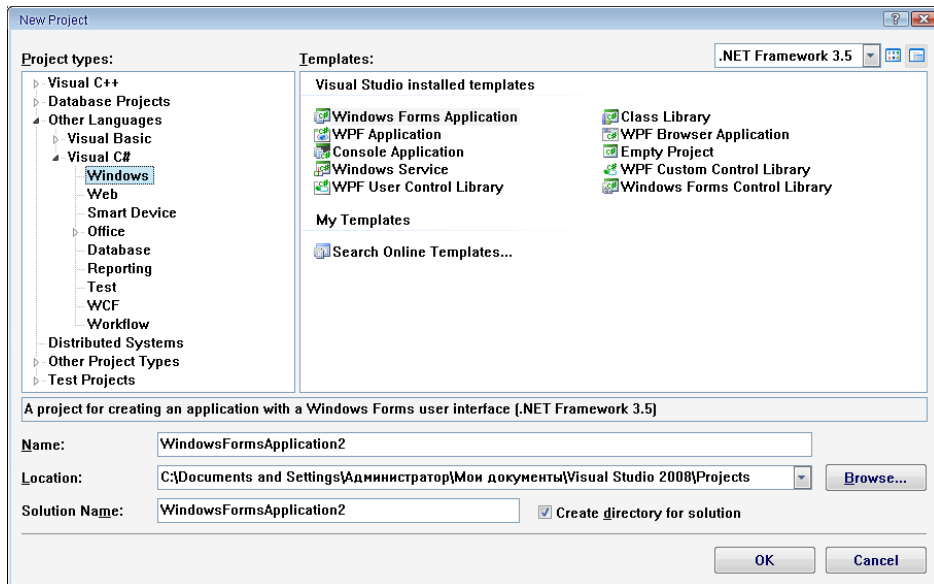


Рис. 1. Окно создания нового проекта

Обратите внимание на окно **Form1**, расположенное в центре рисунка. Это форма класса **System.Windows.Forms**, которая будет играть роль главного окна нашего приложения. Пока это окно пустое, но скоро мы добавим сюда элементы управления. Элементы управления – это то, что вы помещаете в форму. Они нужны для вывода информации, например текстовой (элемент управления **Label**) или графической (элемент управления **PictureBox**), либо для выполнения определенных действий, например, выбора значения или перехода к другой форме после нажатия кнопки. Все элементы управления помещаются на форму.

Понятие «форма», принятое в программировании, родственно понятию «форма анкеты» или «форма документа» из обычной жизни. На форме можно в определенном порядке расположить различные элементы. Когда нам дают готовую форму документа и просят ее заполнить, мы обычно читаем содержащуюся в ней типовую информацию, а затем вписываем недостающие данные в определенные строки.

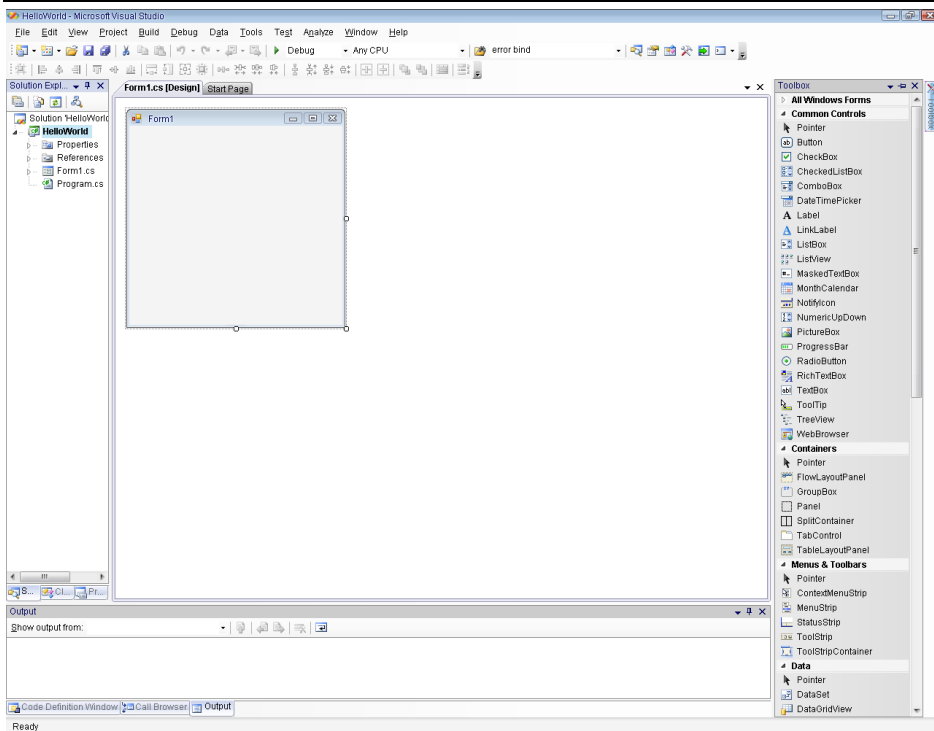


Рис. 2. Окно Visual Studio

ГЛАВНОЕ ОКНО MICROSOFT MICROSOFT VISUAL C#

Давайте подробнее ознакомимся с главным окном системы разработки **Microsoft Visual Studio**, что бы идентифицировать расположенные там наиболее важные элементы.

Рабочее окно

В центре главного окна Microsoft Visual Studio расположено рабочее окно с вкладками. На рисунке 2 видны вкладки **Form1.cs[Design]** и **StartPage**. По мере работы с проектом там будут появляться и другие вкладки. Вкладка **StartPage** содержит множество разнообразной полезной функциональной информации, однако останавливаться на ней мы не будем.

Вкладка проектирования формы

Визуальное проектирование формы осуществляется на вкладке **Form1.cs[Design]**. Эта вкладка показана на рисунке 2.

При необходимости вы можете изменить размеры создаваемой формы и цвет ее фона, разместить в форме текст, графические изображения и различные элементы управления.

Окно Solution Explorer

Окно **Solution Explorer**, показанное на рисунке 3 позволяет просматривать и редактировать файлы проекта. Для нашего проекта было создано одно решение. Ему соответствует корневой значок дерева файлов, обозначенный как **Solution 'HelloWorld'(1 project)**.

На следующем уровне иерархии находится значок проекта с надписью **HelloWorld**. Далее следуют папки входящие в проект.

Папка **Properties** содержит настройки проекта, так например содержащийся в ней **AssemblyInfo.cs** содержит описание сборки проекта.

Form1.cs и **Program.cs** содержат исходный код данного проекта.

Папка **References** содержит перечень классов, на которые имеются ссылки в файлах исходных текстов проекта.

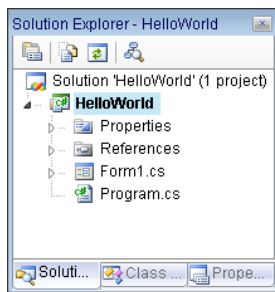


Рис. 3. *Окно Solution Explorer*

2. СОЗДАНИЕ ФОРМЫ И ЭЛЕМЕНТОВ УПРАВЛЕНИЯ, ВВОД ТЕКСТА И ЗАПУСК ПРОГРАММЫ

Создание любой программной системы (проекта) на Visual C# состоит из следующих этапов:

- создания интерфейса (создания окна диалога и размещения в нем элементов управления),
- установки параметров (свойств) элементов управления;
- ввода текста программ и их отладки;
- сохранения проекта в удобном для последующего использования виде.

Рассмотрим реализацию этих этапов на примере простого проекта – программы приветствия мира.

Программа будет содержать поле ввода имени пользователя и кнопку для приветствия.

Пользовательский интерфейс показан на рисунке 4.

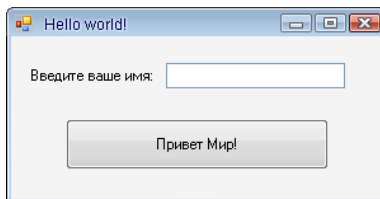


Рис. 4. *Интерфейс создаваемой программы*

При нажатии на кнопку проверяется поле ввода, если оно пусто, то заголовок окна будет меняться «Hello world!», иначе заголовок окна будет меняться на «Привет %имя%».

2.1. СОСТАВ ПРОЕКТА

Проект включает:

- формы (фон, Windows, диалоговые окна);
- элементы управления (графические объекты, помещенные на форму);
- программный код (процедуры и объявления).

Для создания данной программы на форме разместим элементы управления: **Текстовое окно (TextBox)**, **Надпись (Label)** и **Кнопку (Button)** используя **Панель инструментов**.

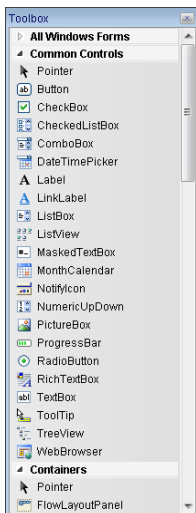


Рис. 5. Панель инструментов *Toolbox*


Создание формы с элементами управления является ответственным этапом, т.к. здесь определяется не только насколько удобна будет программа для пользователя, но и создаются объекты, которые впоследствии будут "оживать" в процессе программирования, являясь "каркасом" приложения.

Элементы управления добавляются с помощью вкладки *Toolbox*, показанной на рисунке 5.:

1. Чтобы создать элемент управления, его необходимо кликнуть на элемент, а затем в нужном месте на форме. Если необходимо разместить несколько одинаковых элементов управления, то следует зажать **Ctrl** и кликать в нужных местах на форме, это упростит работу.

Выделить элемент управления можно, щелкнув по нужному элементу мышью.

После выделения элемента управления или формы в окне свойств можно изменить его свойства.

Примечание: Перед выделением должна быть нажата кнопка  на панели инструментов. Если нажата другая кнопка, то будет нарисован новый элемент управления.

Удалить элемент управления можно, выделив его и нажав клавишу <Delete>.

2. Выберем указанным способом два текстовых окна и разместим их на форме.

3. Активный в данный момент элемент помечен маркерами (щелчком мыши элемент делается активным) и его можно перемещать мышью и изменять его размеры. Это позволяет отредактировать размер и положение управляющих элементов на форме соответственно желанию разработчика.
4. По умолчанию последовательно вводимым одинаковым элементам присваиваются одинаковые имена, отличающиеся последней цифрой. Например *Текстовым* окнам последовательно присваиваются имена **textBox1**, Надписям последовательно присваиваются имена **label1**, кнопкам **button1**.
5. Определим теперь свойства введенных элементов, используя **Окно свойств (Properties)**. Активируем элемент **label1**, при этом активируются свойства в окне свойств, относящиеся к этому элементу. Выберем свойство **Text (Текст)** и определим его как *Введите ваше имя*, используя строку ввода.
6. Определим свойства кнопки. Активируем **button1**, выберем свойство **Text** и определим его как *Привет Мир!* используя строку ввода.
7. Определим свойства основной формы **Form1**. Для этого необходимо активировать форму, щелкнув мышью на свободном пространстве формы. Определим свойство **Text** как *Hello World!*

2.2. ВВОД ТЕКСТА ПРОГРАММЫ

Для ввода текста необходимых программ дважды щелкните мышью на элементе **Кнопка button1**. Откроется окно *Form1.cs* (имя проектируемой формы по умолчанию *Form1*). Появившаяся функция **button1_click** уже связана с кнопкой **button1**, как функция обработчик нажатия на кнопку. В нее следует дописать код, который должен срабатывать при нажатии на кнопку. Посмотреть все возможные события, которые могут происходить с элементом управления можно посмотреть на вкладке **Свойств**, выбрав параметр **Events (События)**

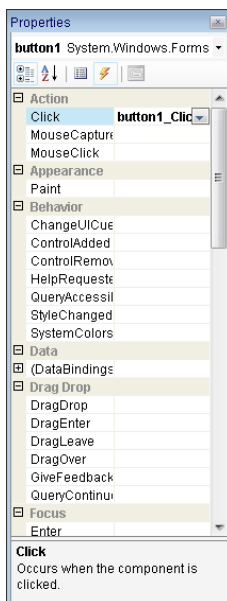


Рис. 6. События кнопки **button1**

Введем необходимый программный код и дадим ему небольшие пояснения. **textBox1.Text** содержит всю информацию о строке, введенной в **textBox1**, параметр **Length** означает длину этой строки. **this** – означает текущую форму, а ее параметр **Text** означает заголовок формы. Таким образом, мы явно его меняем.

```
private void button1_Click(object sender, EventArgs e)
{
    if (textBox1.Text.Length == 0)
        this.Text = "Hello world!";
    else
        this.Text = "Привет " + textBox1.Text;
}
```

Рис. 7. Программный код

2.3. ЗАПУСК ПРОГРАММЫ

Теперь, щелкнув мышью на кнопке **ЗАПУСК** панели инструментов главного меню, проект можно отправить на выполнение. Появляется спроектированная форма, в окнах которой можно вводить информацию (рисунок 8). Кроме ввода и вывода информации, форма соответствует стандарту Windows-приложений (реагирует на кнопки минимизации и максимизации, изменяет размеры и др.).

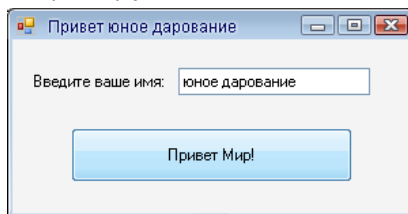


Рис. 8. Результат выполнения программы

3. ЗАДАНИЯ ДЛЯ САМОКОНТРОЛЯ

1. ЗАДАНИЕ 1

Пример программы.


Создадим новый проект с именем *Example1*, в котором открывается форма **Form1**.

Разместим на форме Надпись (**label**) в левом верхнем углу, очистив полностью свойство **Text**. В событиях формы выберем **Click** и дважды щелкнем на нем мышью, а также на событии **KeyPressed**.

В появившейся функции введем следующий код, представленный на рис. 9.

Программа работает следующим образом.

При щелчке кнопки мыши в любом месте формы исходное положение и размер формы (заданы по умолчанию) изменяются, задается курсивный шрифт и его размер для вывода на форму текста.

При нажатии клавиши изменяются исходное положение и размер формы, задается полужирный шрифт и его размер, которым выводится на форму нажатая клавиша. Запустив проект на выполнение кнопкой  **Главного меню**, увидим пустую форму. Щелкнув на ней мышью, увидим изменение положения и размеров текста. Нажав клавишу **<j>**, увидим измененную форму (рисунок. 11).

```
private void Form1_Click(object sender, EventArgs e)
{
    this.SuspendLayout();

    //
    // label1
    //
    label1.Font = new Font("Arial", 14F, System.Drawing.FontStyle.Italic);
    label1.Text = "Мышь";
    //
    // Form1
    //
    this.Size = new Size(200, 500);

    this.ResumeLayout(false);
    this.PerformLayout();
}

private void Form1_KeyPress(object sender, KeyPressEventArgs e)
{
    this.SuspendLayout();

    //
    // label1
    //
    label1.Font = new Font("Times New Roman", 24F, System.Drawing.FontStyle.Underline);
    label1.Text = "Нажатая кнопка " + e.KeyChar.ToString();
    //
    // Form1
    //
    this.Size = new Size(500, 200);

    this.ResumeLayout(false);
    this.PerformLayout();
}
```

Рис. 9. Код программы для обработки событий

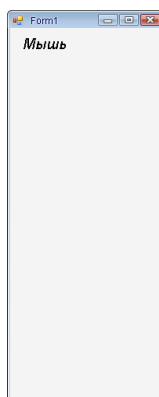


Рис. 10. Вид формы после события *Click*

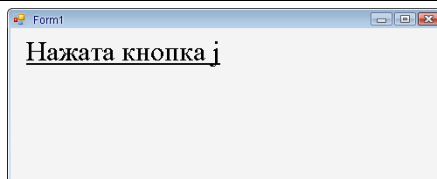


Рис. 11. Вид формы события **KeyPressed**

ЗАДАНИЕ 2

Пример программы.

Разместим на форме управляющие элементы: два текстовых окна **Text1** и **Text2** и метку **Label1** (рисунок 12).

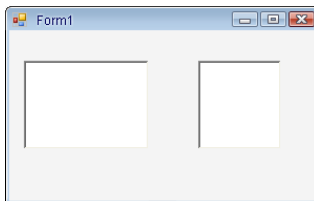


Рис. 12. Первоначальная форма окна

Свойства управляющих элементов заданы по умолчанию.

Введем для первого текстового окна (**Text1**) процедуры обработки событий получения и потери фокуса **Enter** и **Leave** (рисунок 13).


```
private void richTextBox1_Enter(object sender, EventArgs e)
{
    richTextBox1.BackColor = System.Drawing.Color.Aqua;
    label1.Text = "Фокус ввода получен";
}

private void richTextBox1_Leave(object sender, EventArgs e)
{
    richTextBox1.BackColor = System.Drawing.Color.Brown;
    label1.Text = "Фокус ввода потерян";
}
```

Рис. 13. Программный код процедуры обработки событий **Enter** и **Leave**

В первой и второй функциях первый оператор изменяет цвет текстового окна.

Второй параметр определяет содержание **Надписи**.

Щелкнув мышью по кнопке  Главного меню запустим программу. Если щелкнуть мышью по первому текстовому окну (активизировать данный управляющий элемент), то первоначальная форма примет другой вид (рис. 14).

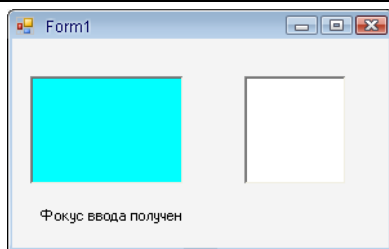


Рис. 14. *Окно в фокусе*

Щелчок мышью по второму окну (первое окно становится не активным) изменяет вид формы (рис. 15).

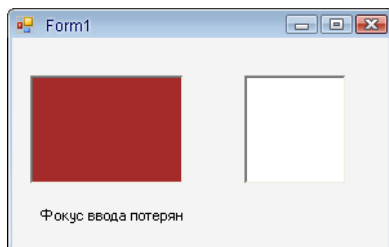


Рис. 15. *Окно НЕ в фокусе*

Продемонстрируйте Вашу работу преподавателю!

КОНТРОЛЬНЫЕ ВОПРОСЫ:

1. Расскажите, как создать новый проект в Microsoft Visual C#?
2. Перечислите основные элементы среды программирования Microsoft Visual C#.
3. Как в Microsoft Visual C# создать форму и разместить на ней элементы управления?
4. Дайте понятие пользовательский интерфейс.
5. Какие элементы составляют проект Microsoft Visual C#..?

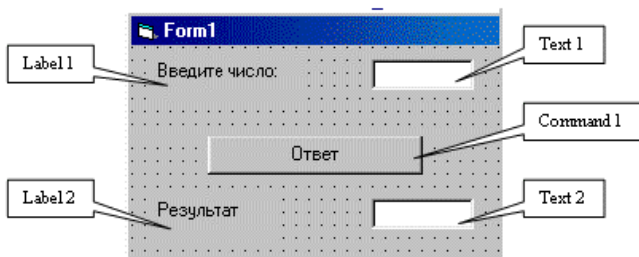
4. ПРАКТИЧЕСКИЕ ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

ПРАКТИЧЕСКОЕ ЗАДАНИЕ №1

1. Выберите в реальной жизни любой объект (холодильник, магнитофон, машина) и опишите его **свойства, методы, события**.
2. Выберите любой объект на экране компьютера (кнопка, окно, текстовое поле) и опишите его **свойства, методы, события**.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ №2

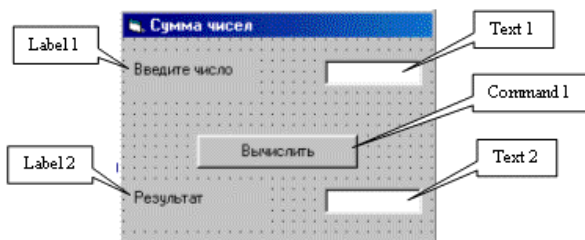
1. Разработайте программу "Угадайка", которая будет загадывать число, а пользователь попытаться его угадать:
2. Создайте следующую форму:

**Рис. 16.**

3. Дважды щелкните по кнопке Ответ и в процедуре обработки события создайте код, который сравнивает введенное пользователем число в **Text1.text** с любым фиксированным числом, например, 5. Если числа равны, то в **Text2.text** должно быть введено "Угадал", в противном случае "Не угадал".
4. Запустите программу, введите исходные данные, щелкните по кнопке Ответ, протестируйте программу, остановите, исправьте при необходимости ошибки в программе.
5. Доработайте программу, заданную в первом пункте так, чтобы она вычисляла объем параллелепипеда путем перемножения длины, ширины и высоты.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ №3

1. Разработайте программу, вычисляющую сумму чисел от 1 и по порядку до указанного пользователем числа.
2. Создайте следующую форму:

**Рис. 17.**

3. Дважды щелкните по кнопке **Вычислить** и в процедуре обработки события напишите код, который суммирует числа от 1 и до введенного пользователем в **Text1.text**, то есть если пользователь введет число 5, то программа суммирует: **1+2+3+4+5**. Суммирование выполняйте с помощью цикла, в теле которого накапливайте значение: **S = S + ...**
4. Результат выведите в **Text2.text**.

5. Запустите программу, введите исходные данные, щелкните по кнопке **Вычислить**, протестируйте программу, остановите, исправьте при необходимости ошибки в программе.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ №4

1. Разработайте программу для исследования математических функций.
2. Создайте следующую форму:

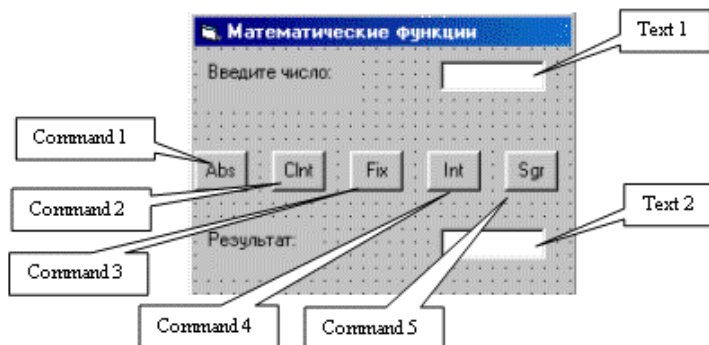


Рис. 18.

3. Дважды щелкните по кнопке **Abs** и в процедуре обработки события напишите код, который в качестве аргумента функции берет число из **Text1.text**, результат функции выводит в **Text2.text**.
4. Также дважды щелкайте по другим кнопкам и в соответствующие процедуры вводите аналогичный код.
5. Запустите программу, введите исходные данные, щелкните по кнопкам, протестируйте программу, остановите, исправьте при необходимости ошибки в программе.
6. Доработайте программу таким образом, чтобы перед вычислением квадратного корня число становилось неотрицательным.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ №5

1. Разработайте программу, которая будет вычислять площадь прямоугольника по заданным двум строкам, для этого:
2. Создайте следующую форму:

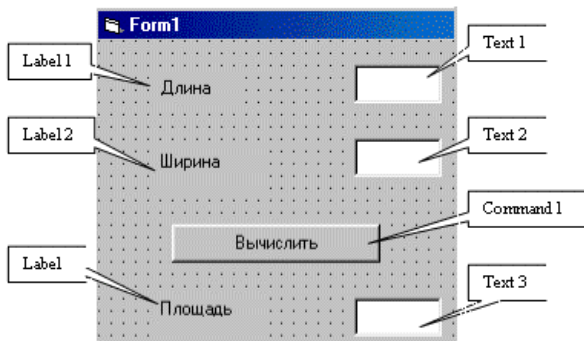


Рис. 19.

3. Дважды щелкните по кнопке **Вычислить** и в процедуре обработки события создайте код, вычисляющий площадь прямоугольника путем умножения его длины на ширину. Не забудьте, что длина располагается в **Text1.text**, ширина в **Text2.text**, а результат - площадь необходимо поместить в **Text3.text**.
4. Запустите программу, введите исходные данные, щелкните по кнопке **Вычислить**, протестируйте программу, остановите, исправьте при необходимости ошибки в программе.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ №6

1. Разработайте программу, которая подсчитывает, сколько символов ввел пользователь и выводит это число на форму.
2. Создайте следующую форму:

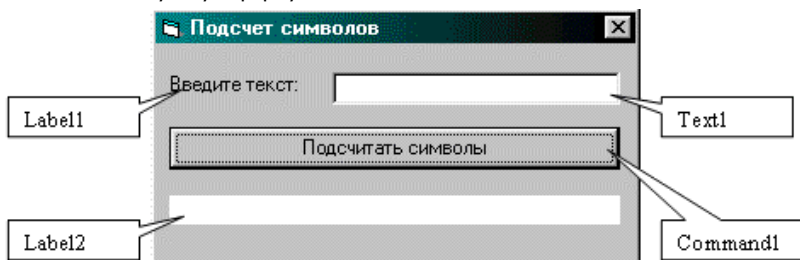
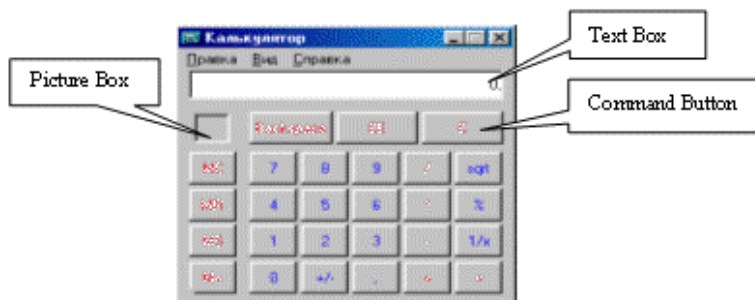


Рис. 20.

3. Дважды щелкнуть по кнопке **Command1** и в процедуре обработки события создайте код, который определяет количество символов, содержащихся в **Text1.text**.
4. Результаты нужно вывести в **Label2.Caption**, но в таком виде: Вы ввели 5 символов. Для этого воспользуйтесь приемом сложения строк, например: "Летели + i + вороны". Здесь переменная i содержит число – количество ворон. Не забывайте про пробелы, чтобы результат не был слитным.
5. Запустите программу, введите исходные данные, щелкните по кнопке **"Подсчитать символы"**, протестируйте программу, остановите, исправьте при необходимости ошибки в программе.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ №7

1. Откройте среду программирования Microsoft Visual C#. Создайте новый проект.
2. В окне среды программирования найдите все элементы, уберите их с экрана и выведите вновь, переместите элементы, подберите их оптимальные размеры.
3. Добавьте новую форму, обратите внимание на ее имя. Закройте и откройте окна форм, измените надписи в заголовках форм:
 - у первой "Обычный калькулятор"
 - у второй "Инженерный калькулятор"
4. Откройте и закройте окна кода первой формы, затем – второй.
5. Запустите программу. Какая форма появится? Сделайте так, чтобы при запуске появлялась другая форма.
6. Удалите вторую форму из проекта, не сохраняя ее.
7. Сохраните проект и форму в отдельной папке. Папку создайте внутри папки Мои документы. В качестве имени папки используйте свою фамилию.

**Рис. 21.**

8. Отрегулируйте размеры формы по содержимому с помощью мыши или свойств **Height** (высота) и **Width** (ширина).
9. Напишите программный код для выполнения расчетов.
10. Сохраните проект и форму в файлах с именами **Калькулятор**.