

Создание Windows-приложений

Цель работы: освоить приёмы разработки Windows-приложений в среде Microsoft Visual Studio.

Порядок выполнения работы:

1. Реализуйте Windows-приложение из примера.
2. Выполните для своего варианта индивидуальное задание на разработку Windows-приложения.

Процесс создания Windows-приложения состоит из двух основных этапов:

1. Визуальное проектирование, то есть задание внешнего облика приложения.
2. Определение поведения приложения путем написания процедур обработки событий.

Визуальное проектирование заключается в помещении на форму компонентов (элементов управления) и задании их свойств и свойств самой формы с помощью окна свойств.

Определение поведения программы начинается с принятия решений, какие действия должны выполняться при щелчке на кнопках, вводе текста, выборе пунктов меню ит. д., иными словами, по каким событиям будут выполняться действия, реализующие функциональность программы.

Пример: создать меню с командами Translate, Help, About, Exit. При выборе команды Exit приложение завершает работу. При выборе команды Translate открывается диалоговое окно, содержащее:

- поле ввода типа TextBox с меткой Binary number;
- поле ввода типа TextBox для вывода результата (read-only);
- группу из трех переключателей (8, 10, 16) типа RadioButton;
- кнопку Do типа Button.

Обеспечить возможность:

- ввода числа в двоичной системе в поле Binary number;
- выбора режима преобразования с помощью переключателей: перевод в восьмеричную, десятичную или шестнадцатеричную систему счисления.

При щелчке на кнопке Do должен появляться результат перевода.

1. СОЗДАНИЕ НОВОГО ПРОЕКТА

В меню «**File** > **New**» выберите команду «**Project**».

В появившемся окне диалога «**New Project**», в окне «**Project Type**» раскройте узел дерева под именем «**Visual C# Projects**» и выберите узел **Windows**.

В окне «**Visual Studio installed templates**» выберите тип проекта **Windows Forms Application**.

В окне «**Name**» задайте имя проекта.

После нажатия на кнопку **ОК** перед вами появится рабочая область Visual Studio 2008.

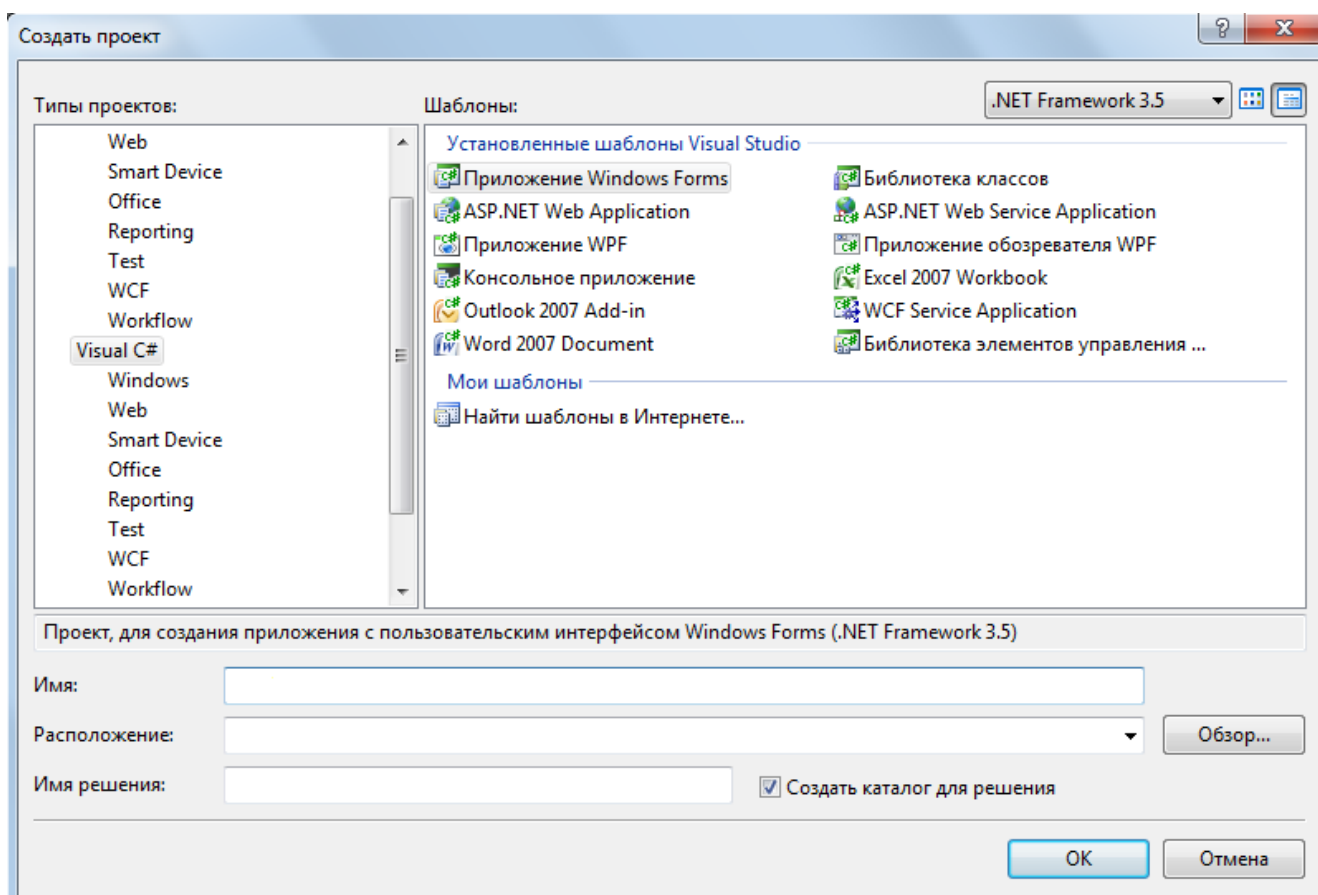


Рис. 1.1. Окно диалога New Project

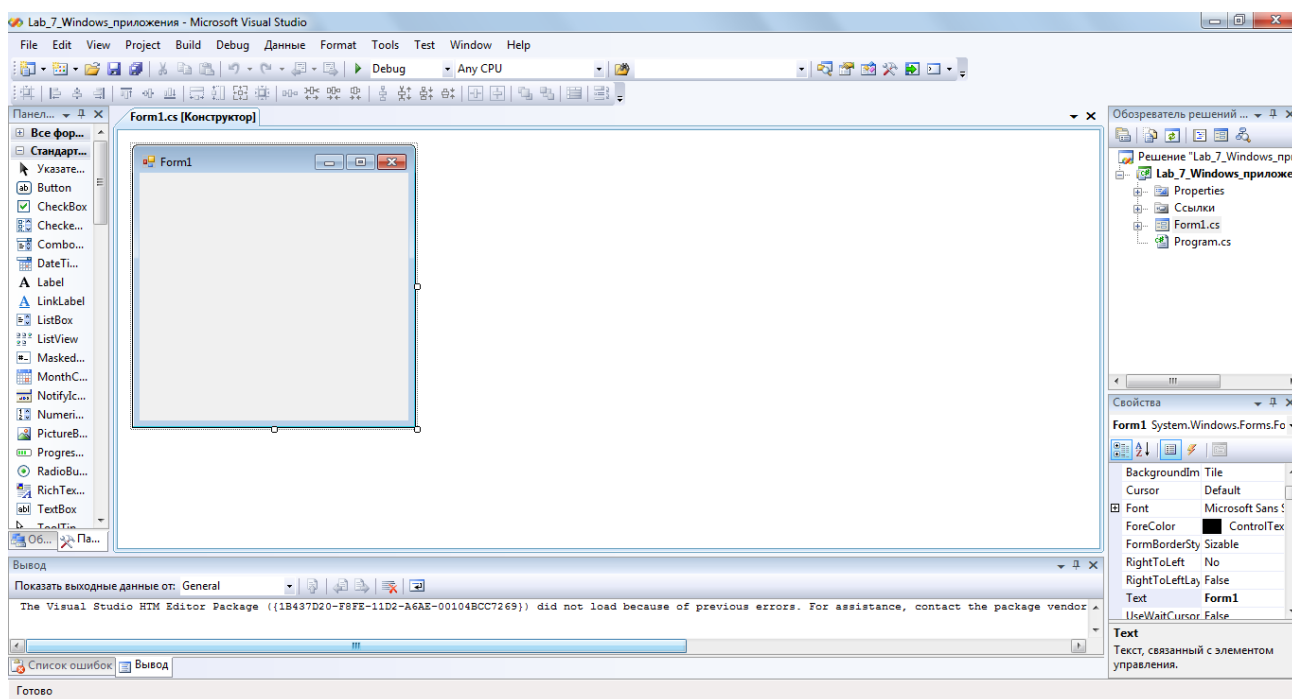


Рис. 1.2. Основное окно Visual Studio 2008

2. ПРИМЕР ПРОГРАММЫ

Разместим на форме одно меню (**Menu Strip**) для выбора необходимой пользователю функции.

В данном меню создадим четыре команды: Translate, Help, About, Exit. Форма примет вид:

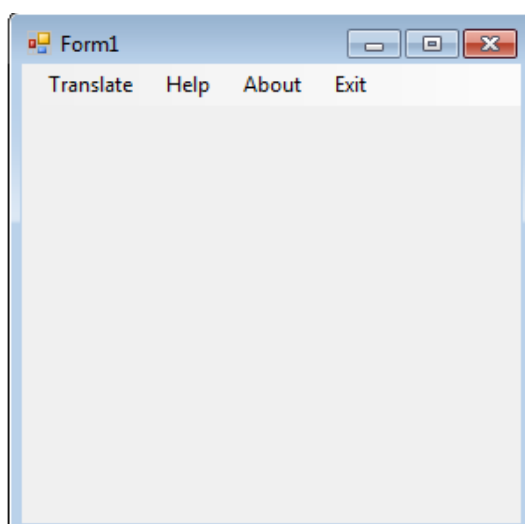


Рис. 1.3. Форма с размещенными компонентами

Теперь для того что бы создать события нажатия на команду «**Translate**», необходимо два раза кликнуть по ней. Автоматически сгенерируется код:

```
private void translateToolStripMenuItem_Click(object sender, EventArgs e)
{

}
```

Теперь между скобок напишем код, который будет выполняться при нажатии на команду на форме «**Translate**».

```
Form2 dial = new Form2(); //создание экземпляра класса окна
dial.ShowDialog();      //отображение диалогового окна
```

Таким образом, при нажатии команды «**Translate**» будет вызываться новое диалоговое окно. Именно поэтому мы для отображения диалогового окна создали экземпляр объекта соответствующей формы, а затем вызвали для этого объекта метод ShowDialog.

Аналогично создадим события нажатия на команду «**Help**». Автоматически сгенерируется код:

```
private void helpToolStripMenuItem_Click(object sender, EventArgs e)
{

}
```

Теперь между скобок напишем код, который будет выполняться при нажатии на команду на форме «**Help**». Пусть при нажатии на кнопку «**Help**», пользователю будет выводиться информация о действиях, которые ему необходимо совершить, для демонстрации работы программы.

Таким образом, код, выполняющийся при нажатии на команду «**Help**» получил вид:

```
private void helpToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("Введите число в двоичной СС в поле Binary number,
выберите режим преобразования с помощью переключателей: 8(перевод в
восьмеричную), 10(в десятичную) или 16(в шестнадцатеричную)СС. И нажмите
кнопку DO, для вывода результата.");
}
```

Аналогично создадим события нажатия на команду «**About**». Автоматически сгенерируется код:

```
private void aboutToolStripMenuItem_Click(object sender, EventArgs e)
{

}
```

Теперь между скобок напишем код, который будет выполняться при нажатии на команду на форме «**About**». Пусть при нажатии на кнопку «**About**», пользователю будет выводиться информация о назначении программы и создателе.

Таким образом, код, выполняющийся при нажатии на команду «**About**» будет иметь вид:

```
private void aboutToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("Программа для перевода чисел из двоичной СС в
    восьмеричную, десятичную и шестнадцатичную. Создатель: указать ФИО.");
}
```

Аналогично создадим события нажатия на команду «**Exit**». Автоматически сгенерируется код:

```
private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{

}
```

Теперь между скобок напишем код, который будет выполняться при нажатии на команду на форме «**Exit**». Пусть при нажатии на кнопку «**Exit**» программа будет закрываться.

Таким образом, код, выполняющийся при нажатии на команду «**Exit**» будет иметь вид:

```
private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
    Close();
}
```

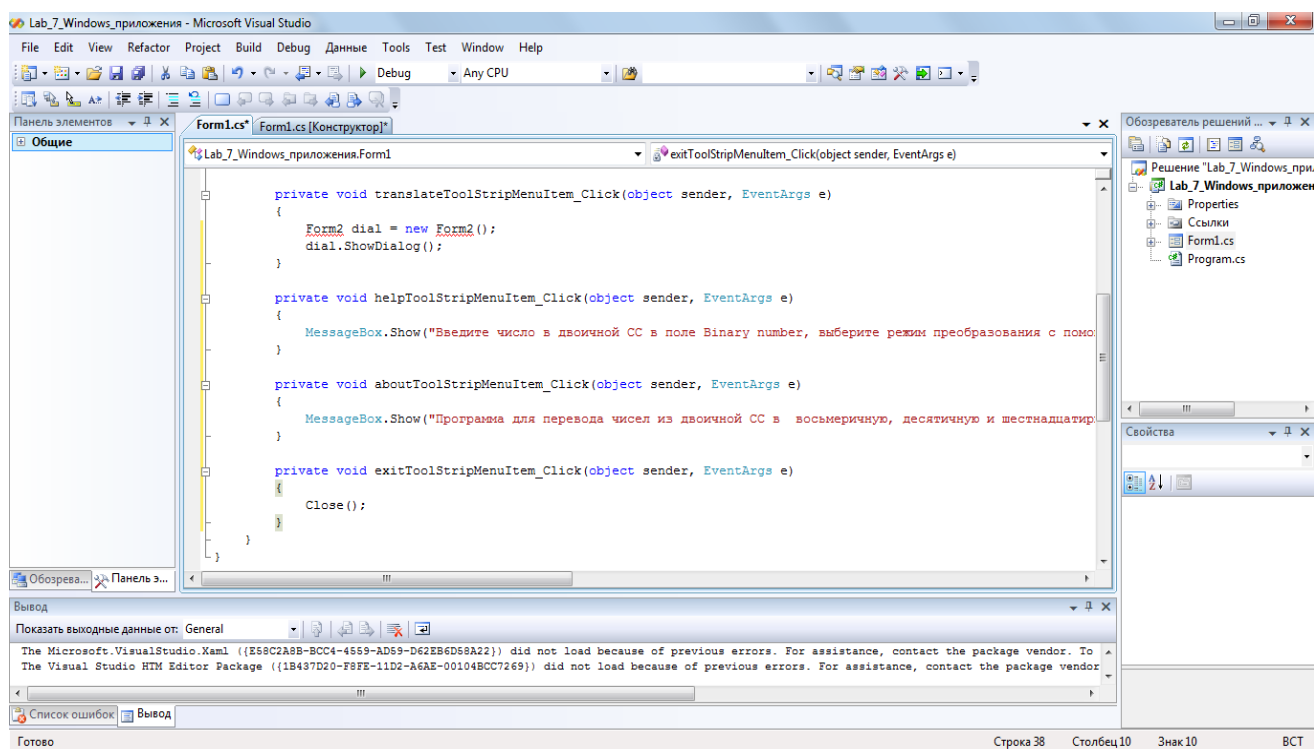


Рис. 1.4. Код программы(команды меню «Menu Strip»).

Теперь в меню, которое находится сверху в открытом окне **Microsoft Visual Studio**, находим пункт «**Project**». Левой кнопкой мыши щелкаем один раз, и в вызванном контекстном меню выбираем пункт «**Добавить форму Windows**».

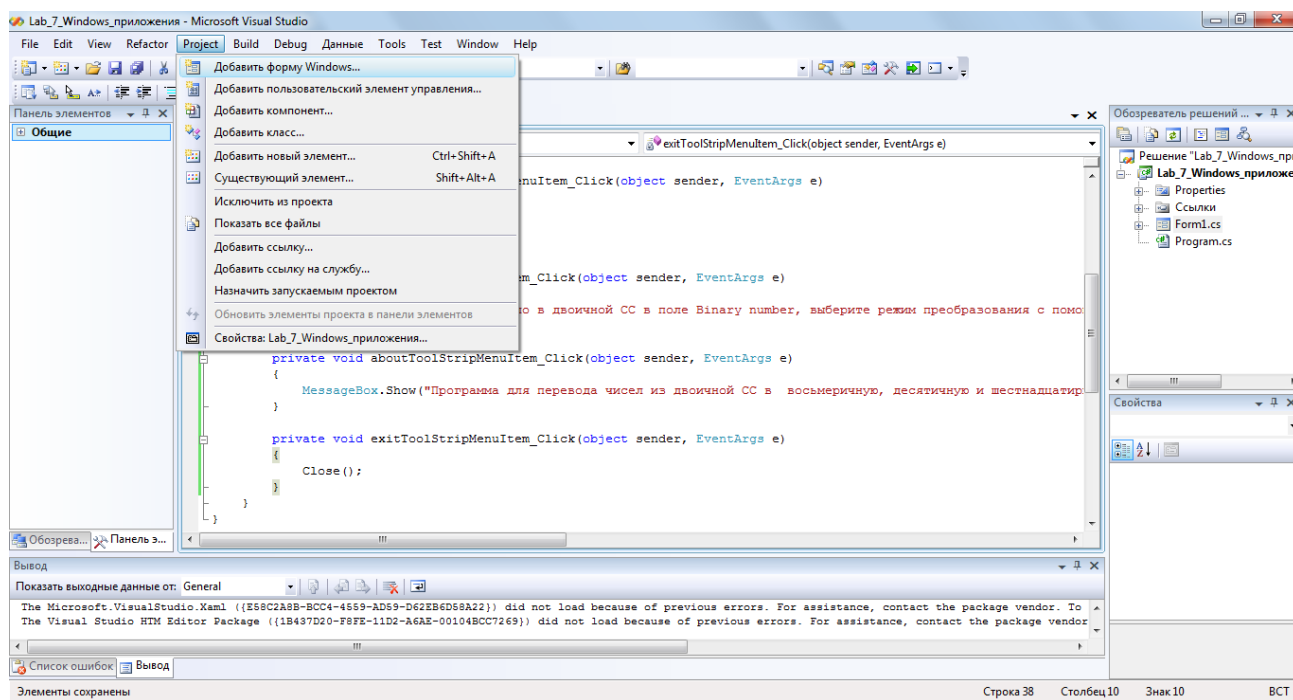


Рис. 1.5. Меню добавления новых элементов

Теперь во вновь возникшем окне «Добавление нового элемента» не изменяя названия нашей формы «**Form2**» нажимаем кнопку «Добавить».

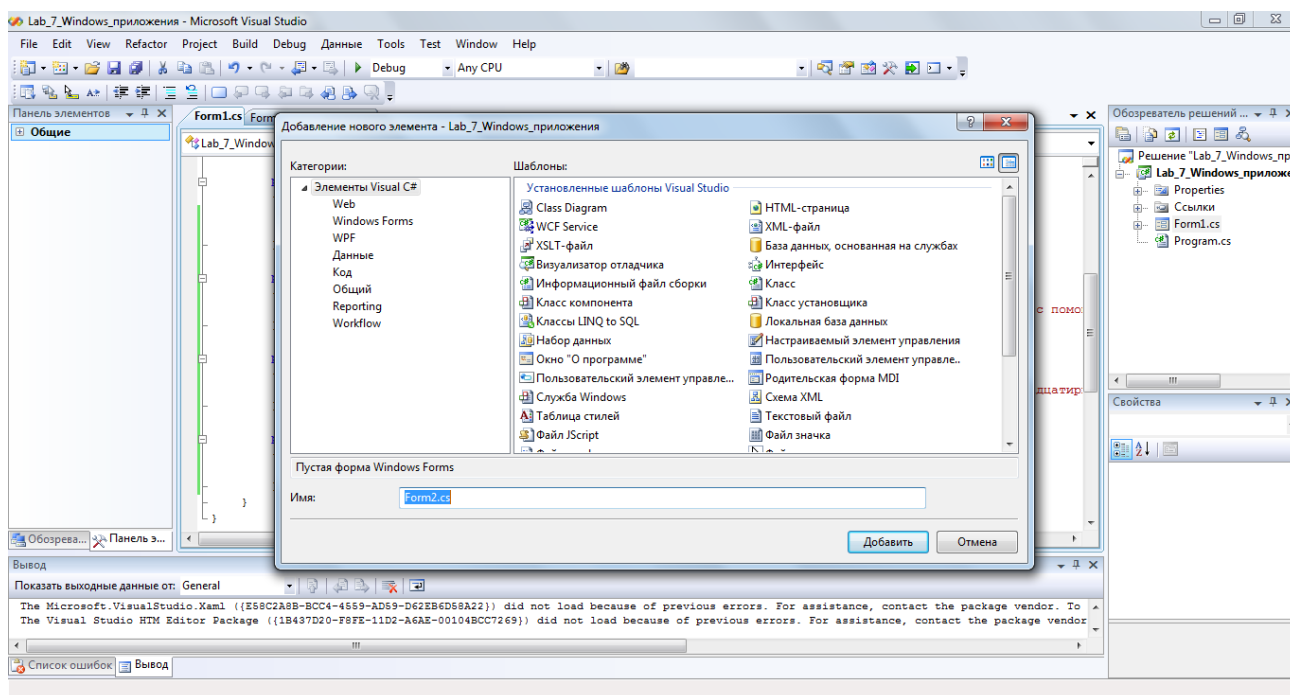


Рис. 1.6. Окно добавления нового элемента

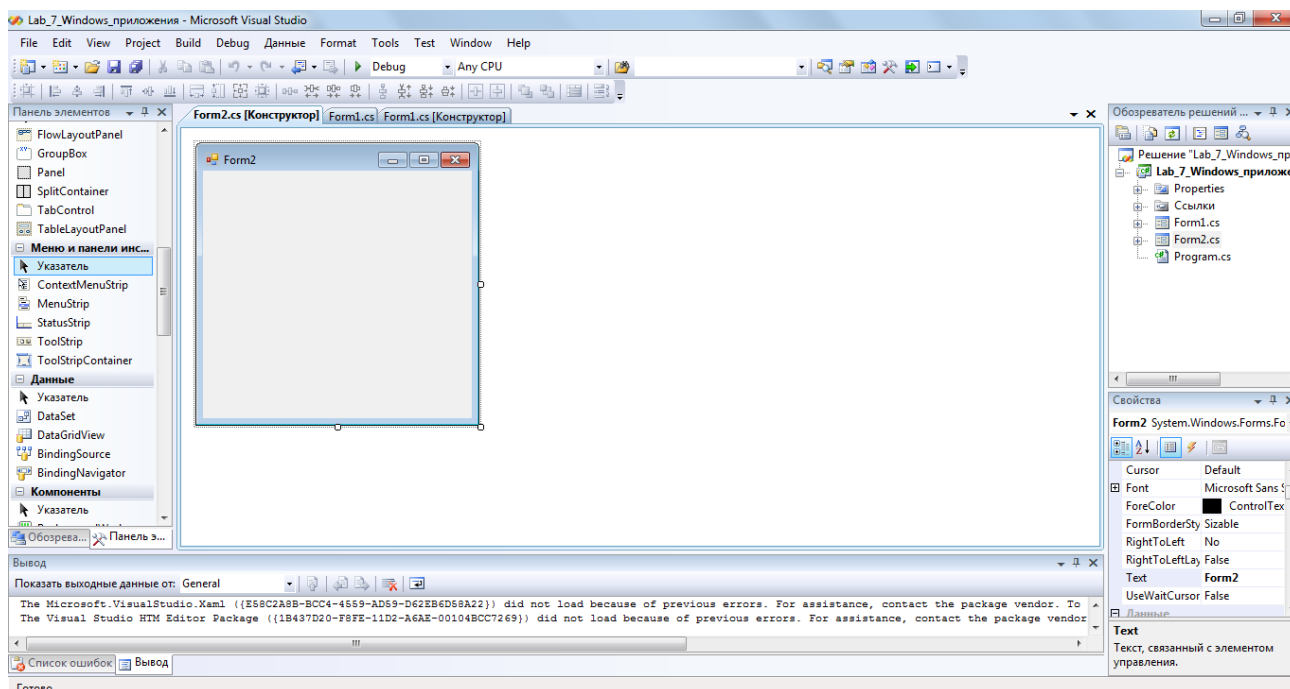


Рис.1.7. Основное окно Visial Studio(Form 2)

Разместим на форме:

- два поля (**TextBox**). Одно для ввода числа в двоичной системе счисления (Binary number), а другое для вывода результата работы программы.

- три поля (**TextBox**) для вывода результата (значения площади окружности, радиуса окружности и попадания заданной точки внутрь окружности)
- три флаговых кнопки (**RadioButton**) для выбора, в какую систему счисления выполнять перевод.
- кнопку (**Button**) для выполнения выбранного действия.
- три поля(**Label**), в которых подпишем назначения полей textbox и флаговых кнопок.

Изменим в инспекторе свойств объектов, значения названия кнопок. Для этого в списке **Appearance** в параметре **Text** введем название нашей кнопки, а именно «**DO**».

Форма примет вид:

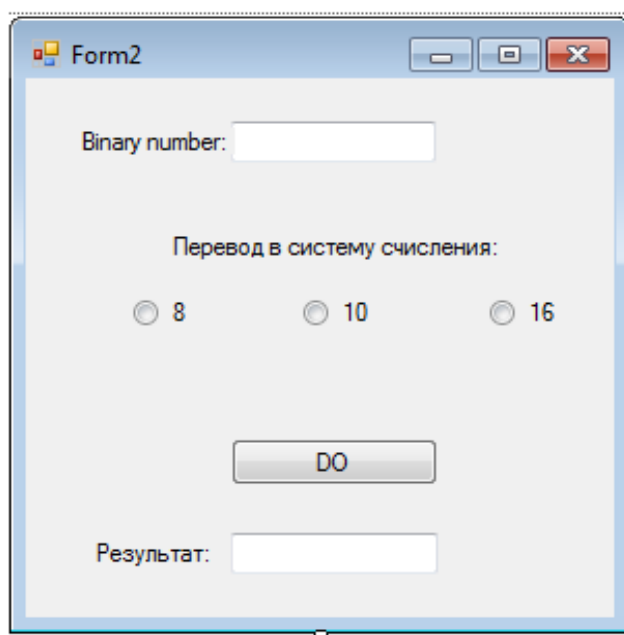


Рис. 1.8. Форма с размещенными компонентами

Теперь для того что бы создать события нажатия на кнопку «**DO**», необходимо два раза кликнуть по ней. Автоматически сгенерируется код:

```
private void button1_Click(object sender, EventArgs e)
{
}
}
```


Теперь между скобок напишем код, который будет выполняться при нажатии на команду на форме «**DO**».

`char[] ch = textBox1.Text.ToCharArray();` // объявление массива символов, который вводится через `textBox1`.

Метод `ToCharArray` преобразует строку, введенную через `textBox1` в массив символов.

`int parent = 0;` //инициализация переменной `parent`

`int count = 0;` //инициализация переменной `count`

`for (int i = ch.Length - 1; i >= 0; --i)` // `for` - цикл с параметром, начинаеся со значения `i` равного количеству символов, введенных в `textBox1`. Цикл будет продолжаться пока `i >= 0`, с шагом раыным `i=i-1`.

{

`int j = int.Parse(ch[i].ToString());` // переменной `j` присвоено значение преобразованного элемента строки в переменую типа `integer`.

`if (count != 0 || j != 0)` // оператор `if`. Если переменная `count` не равно 0 или переменная `j` не равно 0, то выполняется следующая математическая формула:

`parent += (int)Math.Pow(j * 2.0, count);` // т.е `parent= parent + Math.Pow(j * 2.0, count)`

`count++;` // т.е `count= count + 1`

}

`if (radioButton1.Checked)` // если пользователем отмечена флаговая кнопка `radioButton1`, то в поле `textBox2` будет выводиться результат перевода в восьмеричную систему

`textBox2.Text = Convert.ToString(parent, 8);`

`else if (radioButton2.Checked)` // иначе, если пользователем отмечена флаговая кнопка `radioButton2`, то в поле `textBox2` будет выводиться результат перевода в десятичную систему

```
textBox2.Text = parent.ToString();
```

`else if (radioButton3.Checked)` // иначе, если пользователем отмечена флаговая кнопка `radioButton3`, то в поле `textBox2` будет выводиться результат перевода в шестнадцатеричную систему

```
textBox2.Text = Convert.ToString(parent, 16);
```

Таким образом, код примет вид:

```
private void button1_Click(object sender, EventArgs e)
{
    char[] ch = textBox1.Text.ToCharArray();
    int parent = 0;
    int count = 0;
    for (int i = ch.Length - 1; i >= 0; --i)
    {
        int j = int.Parse(ch[i].ToString());
        if (count != 0 || j != 0)
            parent += (int)Math.Pow(j * 2.0, count);
        count++;
    }
    if (radioButton1.Checked)
        textBox2.Text = Convert.ToString(parent, 8);
    else if (radioButton2.Checked)
        textBox2.Text = parent.ToString();
    else if (radioButton3.Checked)
        textBox2.Text = Convert.ToString(parent, 16);
}
```

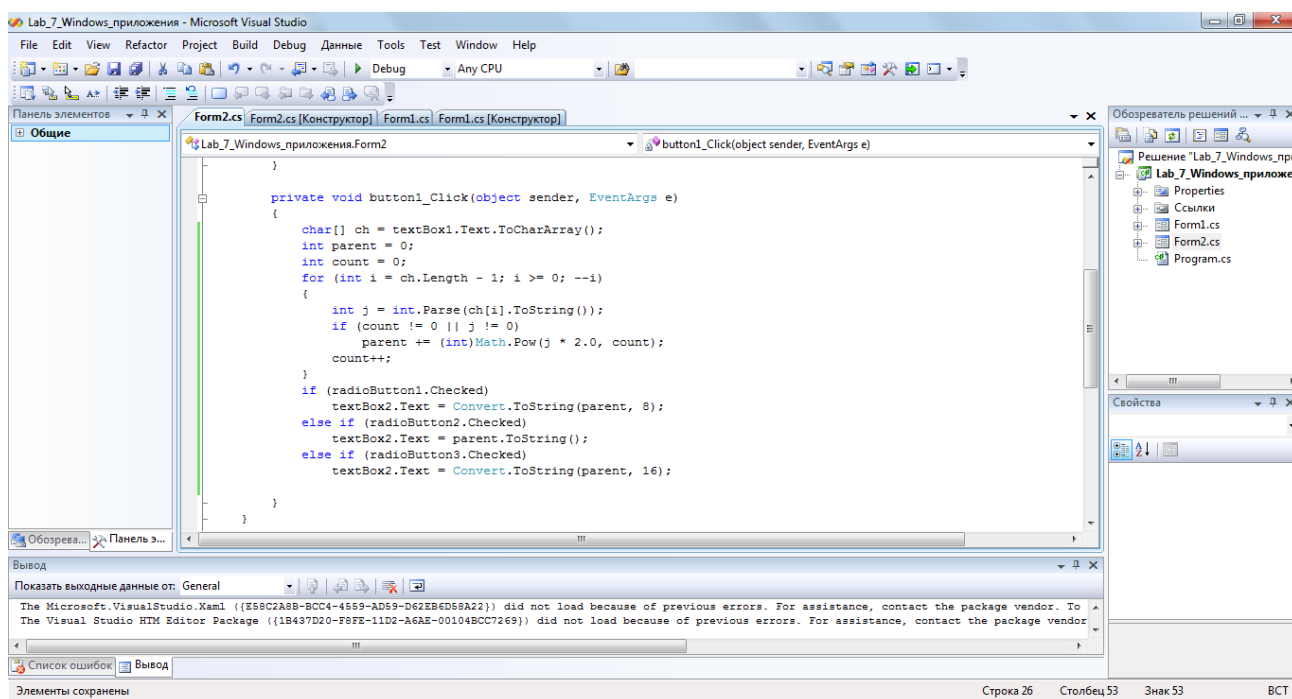


Рис. 1.9. Код программы, выполняющийся при нажатии кнопки «DO»

Следующим шагом мы ограничим ввод символов в поле **textBox1**. Это необходимо нам, для того, чтоб пользователь смог вводить только числа в двоичной системе счисления. А так как все числа в двоичной системе счисления состоят только из 1 и 0, то мы разрешим только ввод 1 и 0, а всё остальные символы вводиться не будут.

Это можно сделать с помощью события **Control.KeyPress**.

В окне **Form2** единожды щелкаем левой кнопкой мыши по полю **textBox1**, в правом нижнем углу, в окне **Свойства** щелкаем по иконке с молнией, откроется вкладка **События**. Нам необходимо найти «**KeyPress**» и дважды кликнуть по нему левой кнопкой мыши. Автоматически сгенерируется код:

```

private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
}

```

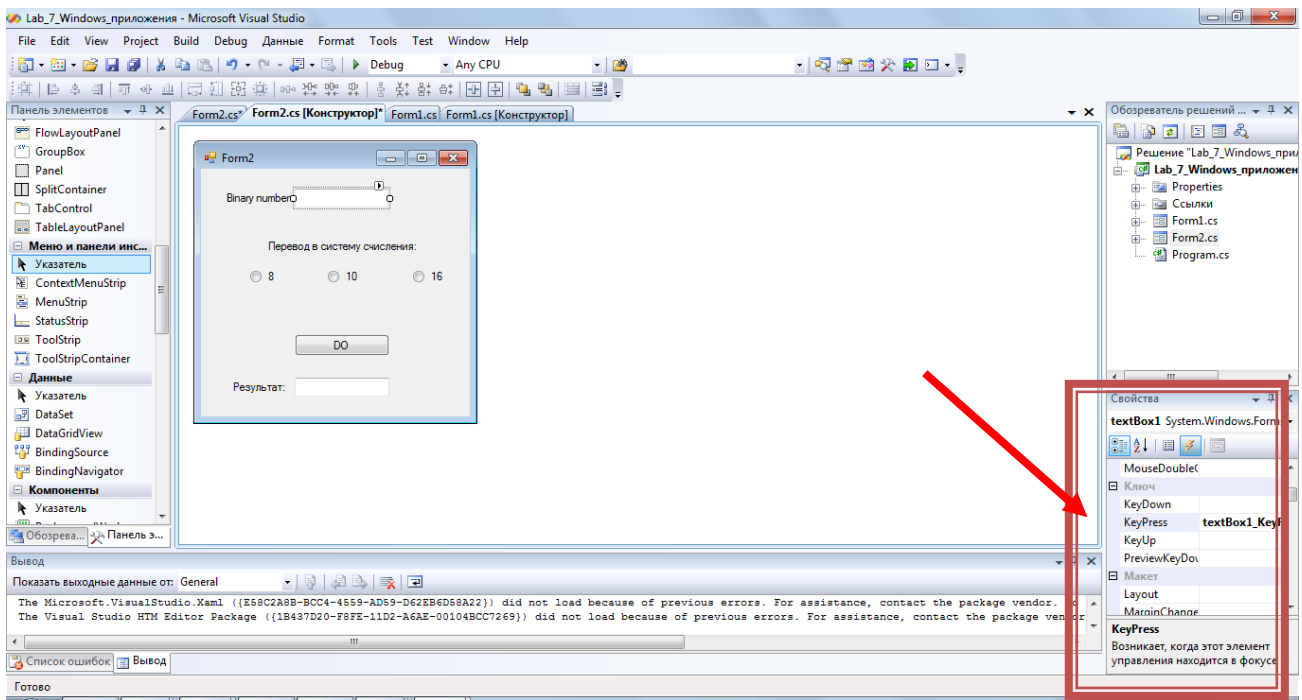


Рис.2. Создание события Control.KeyPress.

Теперь между скобок напишем код, который будет позволять контролировать ввод символов в поле textBox1:

```
if (e.KeyChar != '1' && e.KeyChar != '0')
```

```
e.Handled = true;
```

// если вводимый символ не 1 и не 0, то событие обрабатывается. Т.е. ввести в textBox1 можно только 1 и 0.

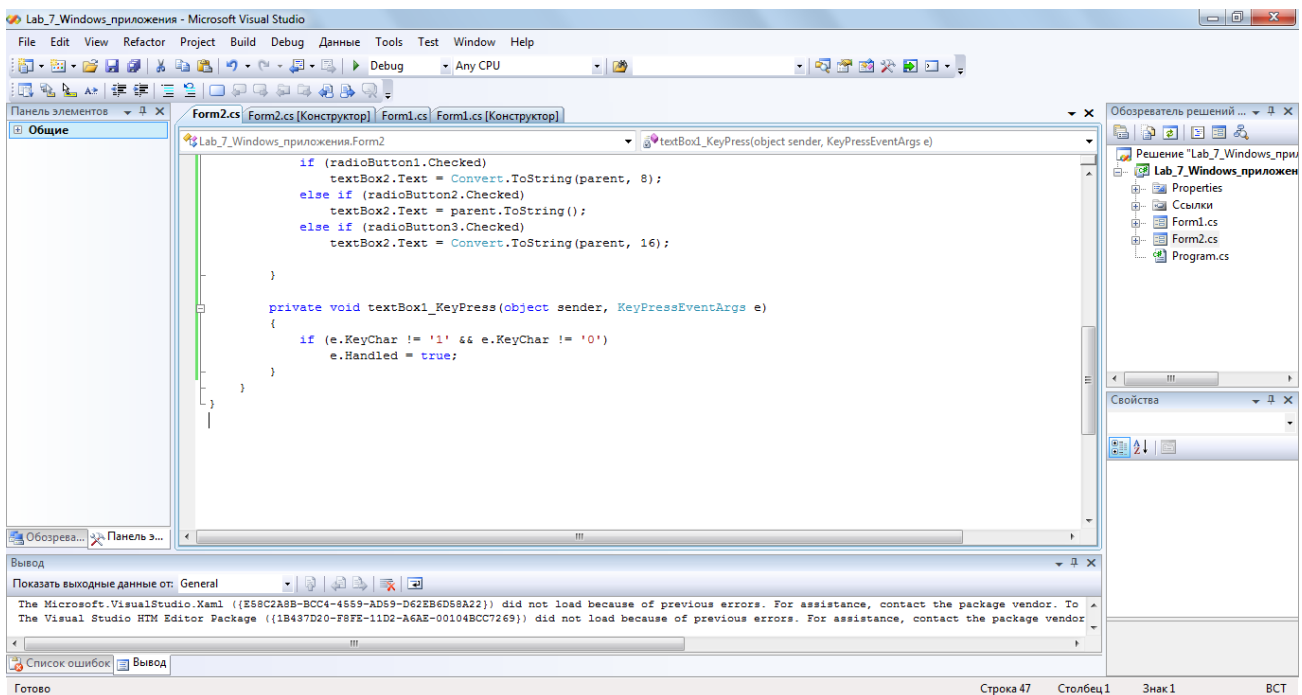


Рис.2.1. Ввод символов в поле textBox1

Код примет вид:

```
private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar != '1' && e.KeyChar != '0')
        e.Handled = true;
}
```

Теперь изменим различные свойства основной формы.

Сделаем так что не было возможно изменять рамеры окна. Для этого выберем в свойствах формы список **Appearance**, в нем параметр **BorderStyle** меняем на **FixedDialog**.

Причем сделаем это и для Form1, и для Form2.

Здесь же изменим заголовок приложения (то что выводится в заголовке окна). Меняем значение поля «**Name**» на «**Лабораторная работа 7**». Причем так же сделаем это и для Form1, и для Form2.

Для запуска проекта выбираем **Dedug > Start Debugging**, либо нажимаем на стрелку для запуска, либо жмем **F5** на клавиатуре.

Ниже приводится ИСХОДНЫЙ КОД данного ПРИЛОЖЕНИЯ.

1) Код Form1:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Lab_7_Windows_приложения
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void translateToolStripMenuItem_Click(object sender, EventArgs e)
        {

```

```

    Form2 dial = new Form2();
    dial.ShowDialog();
}

```

```

private void helpToolStripMenuItem_Click(object sender, EventArgs e)
{

```

```

    MessageBox.Show("Введите число в двоичной СС в поле Binary number, выберите режим преобразования с помощью переключателей: 8(перевод в восьмеричную), 10(в десятичную) или 16(в шестнадцатеричную)СС. И нажмите кнопку DO, для вывода результата.");
}

```

```

private void aboutToolStripMenuItem_Click(object sender, EventArgs e)
{

```

```

    MessageBox.Show("Программа для перевода чисел из двоичной СС в восьмеричную, десятичную и шестнадцатеричную. Создатель: указать ФИО.");
}

```

```

private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{

```

```

    Close();
}

```

```

}
}

```

2) Код Form2:

```

using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

```

```

namespace Lab_7_Windows_приложения
{

```

```

    public partial class Form2 : Form
    {

```

```

        public Form2()
        {
            InitializeComponent();
        }

```

```

        private void button1_Click(object sender, EventArgs e)
        {

```

```

            char[] ch = textBox1.Text.ToCharArray();
            int parent = 0;
            int count = 0;

```

```

for (int i = ch.Length - 1; i >= 0; --i)
{
    int j = int.Parse(ch[i].ToString());
    if (count != 0 || j != 0)
        parent += (int)Math.Pow(j * 2.0, count);
    count++;
}
if (radioButton1.Checked)
    textBox2.Text = Convert.ToString(parent, 8);
else if (radioButton2.Checked)
    textBox2.Text = parent.ToString();
else if (radioButton3.Checked)
    textBox2.Text = Convert.ToString(parent, 16);
}

private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar != '1' && e.KeyChar != '0')
        e.Handled = true;
}
}
}

```

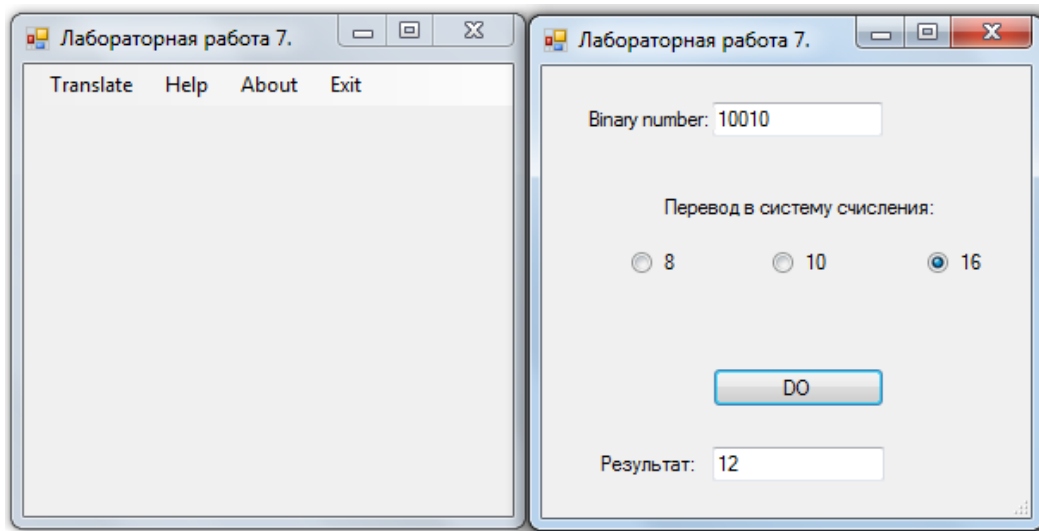


Рис. 2.2. Результат выполнения программы

Индивидуальные задания для самостоятельной работы

Вариант 1

Создать меню с командами Input, Calc и Exit. Команда Calc недоступна. При выборе команды Input открывается диалоговое окно, содержащее:

- три поля типа TextBox для ввода длин трех сторон треугольника;
- группу из двух флажков (Периметр и Площадь) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность:

- ввода длин трех сторон треугольника;
- выбора режима с помощью флажков: подсчет периметра и/или площади треугольника.

При выборе команды Calc открывается диалоговое окно с результатами. При выборе команды Exit приложение завершается.

Вариант 2

Создать меню с командами Size, Calc, Quit. Команда Calc недоступна. При выборе команды Quit приложение завершается. При выборе команды Size открывается диалоговое окно, содержащее:

- два поля типа TextBox для ввода длин сторон прямоугольника;
- группу из двух флажков (Периметр и Площадь) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность:

- ввода длин сторон прямоугольника;
- выбора режима с помощью флажков: подсчет периметра и/или площади прямоугольника.

При выборе команды Calc открывается диалоговое окно с результатами.

Вариант 3

Создать меню с командами Input, Work, Exit. Команда Work недоступна. При выборе команды Exit приложение завершает работу. При выборе команды Input открывается диалоговое окно, содержащее:

- три поля ввода типа TextBox с метками Radius, Height, Density;
- группу из двух флажков (Volume, Mass) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность:

- ввода радиуса, высоты и плотности конуса;
- выбора режима с помощью флажков: подсчет объема и/или массы конуса.

При выборе команды Work открывается окно сообщений с результатами.

Вариант 4

Создать меню с командами Input, Calc, Exit. Команда Calc недоступна. При выборе команды Exit приложение завершает работу. При выборе команды Input открывается диалоговое окно, содержащее:

- поле ввода типа TextBox с меткой Radius;
- группу из двух флажков (Square, Length) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность:

- ввода радиуса окружности;
- выбора режима с помощью флажков: подсчет площади круга (Square) и/или длины окружности (Length).

При выборе команды Calc открывается окно сообщений с результатами.

Вариант 5

Создать меню с командами input, Calc, About. При выборе команды About открывается окно с информацией о разработчике. При выборе команды Input открывается диалоговое окно, содержащее:

- три поля ввода типа TextBox с метками Number 1, Number 2, Number 3;
- группу из двух флажков (Summ, Least multiple) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность ввода трех чисел и выбора режима вычислений с помощью флажков: подсчет суммы трех чисел (Summ) и/или наименьшего общего кратного двух первых чисел (Least multiple). При выборе команды Calc открывается диалоговое окно с результатами.

Вариант 6

Создать меню с командами Input, Calc, Quit. Команда Calc недоступна. При выборе команды Quit приложение завершается. При выборе команды Input открывается диалоговое окно, содержащее:

- два поля ввода типа TextBox с метками Number 1, Number 2;
- группу из трех флажков (Summa, Max divisor, Multiply) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность:

- ввода двух чисел;
- выбора режима вычислений с помощью флажков (можно вычислять в любой комбинации такие величины, как сумма, наибольший общий делитель и произведение двух чисел).

При выборе команды Calc открывается окно сообщений с результатами.

Вариант 7

Создать меню с командами Begin, Help, About. При выборе команды About открывается окно с информацией о разработчике. При выборе команды Help открывается окно с информацией о программе. При выборе команды Begin открывается диалоговое окно, содержащее:

- поле ввода типа TextBox с меткой input;
- метку типа Label для вывода результата;
- группу из трех переключателей (2, 8, 16) типа RadioButton;
- две кнопки типа Button – Do и OK.

Обеспечить возможность:

- ввода числа в десятичной системе в поле input;
- выбора режима преобразования с помощью переключателей: перевод в двоичную, восьмеричную или шестнадцатеричную систему счисления. При щелчке на кнопке Do должен появляться результат перевода.

Вариант 8

Создать меню с командами Begin, Work, About. Команда Work недоступна. При выборе команды About открывается окно с информацией о разработчике. При выборе команды Begin открывается диалоговое окно, содержащее:

- поле ввода типа TextBox с меткой Input word;
- группу из двух переключателей (Upper case, Lower case) типа RadioButton;
- кнопку типа Button.

Обеспечить возможность ввода слова и выбора режима перевода в верхний или нижний регистр в зависимости от положения переключателей. При выборе команды Work открывается диалоговое окно с результатом перевода.

Вариант 9

Создать меню с командами Translate, About, Exit. При выборе команды Exit приложение завершает работу. При выборе команды About открывается окно с информацией о разработчике. При выборе команды Translate открывается диалоговое окно, содержащее:

- поле ввода типа TextBox с меткой Binary number;
- поле ввода типа TextBox для вывода результата (read-only);
- группу из трех переключателей (8, 10, 16) типа RadioButton;
- кнопку Do типа Button.

Обеспечить возможность:

- ввода числа в двоичной системе в поле Binary number;
- выбора режима преобразования с помощью переключателей: перевод в восьмеричную, десятичную или шестнадцатеричную систему счисления.

При щелчке на кнопке Do должен появляться результат перевода.

Вариант 10

Создать меню с командами Reverse, About, Exit. При выборе команды About открывается окно с информацией о разработчике. При выборе команды Reverse открывается диалоговое окно, содержащее:

- поле ввода типа TextBox с меткой Input;
- группу из двух переключателей (Upper case, Reverse) типа CheckBox;
- кнопку ОК типа Button.

Обеспечить возможность ввода фразы и выбора режима: перевод в верхний регистр и/или изменение порядка следования символов на обратный в зависимости от состояния переключателей. Результат преобразования выводится в исходное поле ввода.

При выборе команды Exit приложение завершает работу.

Вариант 11

Создать меню с командами Input, Show и Exit. Команда Show недоступна. При выборе команды Exit приложение завершает работу. При выборе команды Input открывается диалоговое окно, содержащее:

- поля ввода типа TextBox с метками Input;
- группу из двух флажков (length и koef: подсчет длины отрезка, соединяющего эти точки, и/или углового коэффициента) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность:

- ввода двух чисел;
- выбора режима вычислений с помощью флажков.

При выборе команды Show открывается окно сообщений с результатами подсчета.

Вариант 12

Создать меню с командами Begin, Work, About. Команда Work недоступна. При выборе команды About открывается окно с информацией о разработчике. При выборе команды Begin открывается диалоговое окно, содержащее:

- поле ввода типа TextBox с меткой Input;
- группу из двух переключателей (First letter, All letters) типа RadioButton;
- кнопку типа Button.

Обеспечить возможность ввода предложения и выбора режима его преобразования: либо начинать с прописной буквы каждое слово (First letter), либо перевести все буквы в верхний регистр (All letters). При выборе команды Work открывается диалоговое окно с результатом преобразования.

Вариант 13

Создать меню с командами Input, Calc, About. Команда Calc недоступна. При выборе команды About открывается окно с информацией о разработчике. При выборе команды Input открывается диалоговое окно, содержащее:

- три поля ввода типа TextBox с метками Number 1, Number 2, Number 3;
- группу из двух флажков (Summ, Least multiple) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность ввода трех чисел и выбора режима вычислений с помощью флажков: подсчет суммы трех чисел (Summ) и/или наименьшего общего кратного двух первых чисел (Least multiple). При выборе команды Calc открывается диалоговое окно с результатами.

Вариант 14

Создать меню с командами Input, Calc, Quit. Команда Calc недоступна. При выборе команды Quit приложение завершается. При выборе команды Input открывается диалоговое окно, содержащее:

- два поля ввода типа TextBox с метками Number 1, Number 2;
- группу из трех флажков (Summa, Max divisor, Multiply) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность:

- ввода двух чисел;
- выбора режима вычислений с помощью флажков (можно вычислять в любой комбинации такие величины, как сумма, наибольший общий делитель и произведение двух чисел).

При выборе команды Calc открывается окно сообщений с результатами.

Вариант 15

Создать меню с командами Input, Calc, Exit. Команда Calc недоступна. При выборе команды Exit приложение завершает работу. При выборе команды Input открывается диалоговое окно, содержащее:

- поле ввода типа TextBox с меткой Radius;
- группу из двух флажков (Square, Length) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность:

- ввода радиуса окружности;
- выбора режима с помощью флажков: подсчет площади круга (Square) и/или длины окружности (Length).

При выборе команды Calc открывается окно сообщений с результатами.

Вариант 16

Создать меню с командами Input, Calc, Quit. Команда Calc недоступна. При выборе команды Quit приложение завершается. При выборе команды Input открывается диалоговое окно, содержащее:

- два поля ввода типа TextBox с метками Number 1, Number 2;
- группу из трех флажков (Summa, Max divisor, Multiply) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность:

- ввода двух чисел;
- выбора режима вычислений с помощью флажков (можно вычислять в любой комбинации такие величины, как сумма, наибольший общий делитель и произведение двух чисел).

При выборе команды Calc открывается окно сообщений с результатами.

Вариант 17

Создать меню с командами Translate, About, Exit. При выборе команды Exit приложение завершает работу. При выборе команды About открывается окно с информацией о разработчике. При выборе команды Translate открывается диалоговое окно, содержащее:

- поле ввода типа TextBox с меткой Digit number;
- поле ввода типа TextBox для вывода результата (read-only);
- группу из трех переключателей (2, 3, 16) типа RadioButton;
- кнопку Do типа Button.

Обеспечить возможность:

- ввода числа в десятичной системе в поле Digit number;
- выбора режима преобразования с помощью переключателей: перевод в двоичную, троичную или шестнадцатеричную систему счисления.

При щелчке на кнопке Do должен появляться результат перевода.

Вариант 18

Создать меню с командами Input, Calc, About. Команда Calc недоступна. При выборе команды About открывается окно с информацией о разработчике. При выборе команды Input открывается диалоговое окно, содержащее:

- три поля ввода типа TextBox с метками Number 1, Number 2, Number 3;
- группу из двух флажков (Summ, Least multiple) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность ввода трех чисел и выбора режима вычислений с помощью флажков: подсчет суммы трех чисел (Summ) и/или наименьшего общего кратного двух первых чисел (Least multiple). При выборе команды Calc открывается диалоговое окно с результатами.

Вариант 19

Создать меню с командами Reverse, About, Exit. При выборе команды About открывается окно с информацией о разработчике. При выборе команды Reverse открывается диалоговое окно, содержащее:

- поле ввода типа TextBox с меткой Input;
- группу из двух переключателей (Upper case, Reverse) типа CheckBox;
- кнопку ОК типа Button.

Обеспечить возможность ввода фразы и выбора режима: перевод в верхний регистр и/или изменение порядка следования символов на обратный в зависимости от состояния переключателей. Результат преобразования выводится в исходное поле ввода.

При выборе команды Exit приложение завершает работу.

Вариант 20

Создать меню с командами Size, Calc, Quit. Команда Calc недоступна. При выборе команды Quit приложение завершается. При выборе команды Size открывается диалоговое окно, содержащее:

- два поля типа TextBox для ввода длин сторон прямоугольника;
- группу из двух флажков (Периметр и Площадь) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность:

- ввода длин сторон прямоугольника;
- выбора режима с помощью флажков: подсчет периметра и/или площади прямоугольника.

При выборе команды Calc открывается диалоговое окно с результатами.