Практическая работа № 2

Создание и отладка однопоточного приложения

В рамках данного практического занятия, будет рассмотрен процесс создания и отладки однопоточного приложения с использованием Visual Studio 2010.

1. Создадим Windows приложение с названием "WindowsDebugApplication":

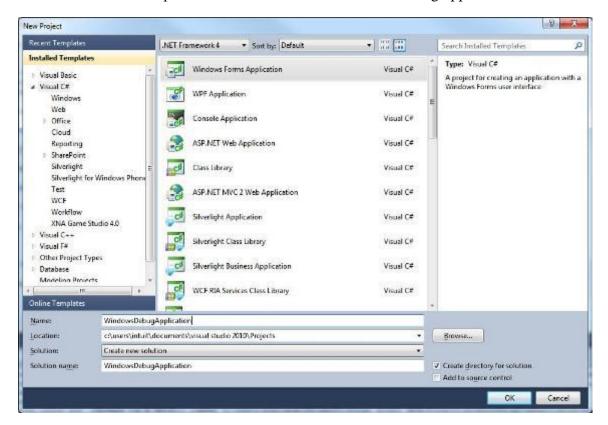


Рис. 1.

2. Создадим простейший калькулятор. Для этого разместим на форме 4 элемента (2 TextBox, 1 ComboBox, 1 Button):



Рис. 2.

3. Добавим в программу следующий код (Коллекция arraysymbol и цикл foreach добавлены для наглядности отладки приложения):

```
4. using System;
5. using System.Collections.Generic;
6. using System.ComponentModel;
7. using System.Data;
8. using System.Drawing;
9. using System.Linq;
10. using System. Text;
11. using System.Windows.Forms;
12. using System.Collections;
13. namespace WindowsDebugApplication
14. {
15.
     public partial class Form1: Form
16.
      {
17.
        public Form1()
18.
19.
          InitializeComponent();
20.
        }
21.
        private double c = 0;
22.
        private void Plus(double a,double b)
23.
24.
          c = a + b;
25.
26.
        private void Minus(double a, double b)
27.
        {
28.
          c = a - b;
29.
30.
        private void Multiply(double a, double b)
31.
        {
32.
          c = a * b;
33.
        }
```

private void Split(double a, double b)

34.

```
35.
36.
          c = a / b;
37.
38.
        private void ResultButton_Click(object sender, EventArgs e)
39.
40.
          string symbol =(string) SymbolcomboBox.SelectedItem;
41.
          ArrayList arraysymbol = new ArrayList();
42.
          foreach (var s in SymbolcomboBox.Items)
43.
          {
44.
            arraysymbol.Add(s);
45.
          }
46.
          double var1 =Convert.ToDouble(FirstVariable.Text);
47.
          double var2 = Convert.ToDouble(SecondVariable.Text);
48.
          switch (symbol)
49.
          {
50.
            case "+": Plus(var1, var1); break;
51.
            case "-": Minus(var1, var1); break;
52.
            case "*": Multiply(var1, var1); break;
53.
            case "/": Split(var1, var1); break;
54.
          }
55.
          MessageBox.Show("Результат: " + c);
56.
          MessageBox.Show("Количество символов в коллекции: "+ arraysymbol.Count);
57.
        }
58.
      }
   }
```

4. Теперь, расставим точки останова (breakpoints) в программе. В примере точки останова расставлены напротив методов математических операций и в событии кнопки:

```
Form1.cs* X Form1.cs [Design]*
Program.cs
WindowsDebuggingApplication.Form1
                                                            · Plus(double a, dou
            public Form1()
               InitializeComponent();
            private double c =0;
            private void Plus(double a, double b)
               c = a + b;
            private void Minus(double a, double b)
                c = a - b;
            }
            private void Multiply(double a, double b)
                c = a * b;
            private void Split(double a, double b)
                c = a / b;
            private void ResultButton_Click(object sender, EventArgs e)
               string symbol =(string) Symbolcombo8ox.5electedItem;
               ArrayList arraysymbol = new ArrayList();
               foreach (var s in SymbolcomboBox.Items)
                   arraysymbol.Add(s);
               double var1 =Convert.ToDouble(FirstVariable.Text);
               double var2 = Convert.ToDouble(SecondVariable.Text);
               switch (symbol)
```

Рис. 3.

Список доступных всех точек останова (Breakpoints), можно посмотреть в специальном окне, которое вызывается из меню Debug пункт меню "Breakpoints" или сочетанием клавиш CTRL+ALT+B:

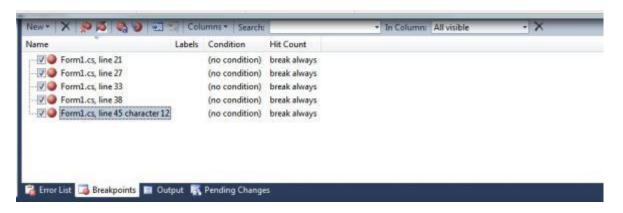


Рис. 4.

5. Запустите отладку приложения с помощью пункта "Start Debugging" - меню "Debbug" или с помощью клавиши "F5":

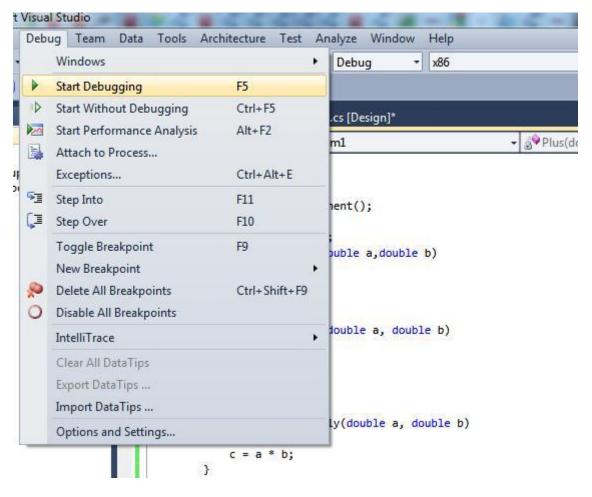


Рис. 5.

6. После ввода значений в поля программы и выбора соответствующей операции (сложение, вычитание и т.д), жмем кнопку "Рассчитать", тем самым вызовется метод ResulButton_Click():

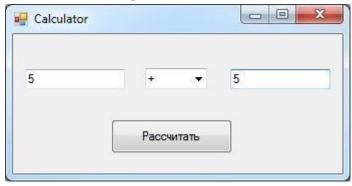


Рис. 6.

7. Запустится пошаговый процесс отладки приложения с точки останова (Breakpoint) в методе ResulButton_Click():

```
Forml.cs X Forml.cs [Design]
WindowsDebuggingApplication.Form1
                                                                                                               - Res
                c = a * b;
            private void Split(double a, double b)
                c = a / b;
            }
            private void ResultButton_Click(object sender, EventArgs e)
               string symbol =(string) SymbolcomboBox.SelectedItem;
ArroyList arraysymbol = new ArrayList();
               foreach (var s in SymbolcomboBox.Items)
                   arraysymbol.Add(s);
               double var1 =Convert.ToDouble(FirstVariable.Text);
               double var2 = Convert.ToDouble(SecondVariable.Text);
               switch (symbol)
                   case "+": Plus(var1, var1); break;
                   case "-": Minus(varl, varl); break;
                   case """: Multiply(var1, var1); break;
                   case "/": Split(var1, var1); break;
               MessageBox.Show("Результат: " + c);
               MessageBox.Show("Количество символов в коллекции: "+ arraysymbol.Count);
       }
  3
```

Рис. 7.

8. Добавим для просмотра значений переменных - переменные arraysymbol (коллекция), и переменную "с". Для этого щелкните на нужной переменной правой кнопкой мыши и выберите из списка "Add Watch":

```
private void ResultButton_Click(object sender, EventArgs e)
   string symbol =(string) SymbolcomboBox.SelectedItem;
  ArrayList arraysymbol = new ArrayList().
   foreach (var s in 🔳 View Designer
      arraysymbol.Ac Create Unit Tests...
                          Generate Sequence Diagram...
  double var1 =Conve
                                                         F12
                         Find All References
                                                          Shift+F12
  double var2 = Conv
  switch (symbol)
                    View Call Hierarchy
                                                         Ctrl+K, Ctrl+T
                          Breakpoint
      case "+": Plus 64 Add Watch
      case "-": Minu
      case "+": Mult 6d QuickWatch...
                                                          Shift+F9
      case "/": Spli
                         Pin To Source
```

Рис. 8.

Переменные также, можно вручную добавлять в список Watch, для этого, достаточно написать имя нужной переменной в колонке Name:

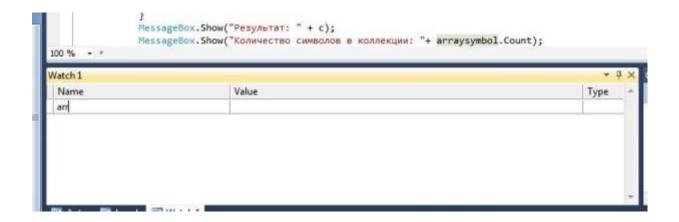


Рис. 9.

Если нужно просматривать состояние всех переменных во время отладки, используется окно Autos (переменные будут появляться в окне автоматически, в зависимости от шага отладчика):

Name	Value	Туре
1 g/ SecondVariable	(Text = '5')	System
symbol	***	Q, v string
this	(WindowsDebuggingApplication.Form1, Text: Calculator)	Windo
	5.0	double

Рис. 10.

9. Используйте кнопку "F10" для пошаговой отладки приложения. В процессе пошаговой отладки, курсор отладчика будет заходить в те методы, которые вызываются в методе ResulButton_Click(), в нашем случае это метод Plus() (т.к была выбрана операция сложения - "+"):

```
private double c =0;
private void Plus(double a,double b)

c = a + b;
}

private void Minus(double a, double b)

c = a - b;
}
```

Рис. 11.

10. В процессе отладки приложения, значения переменных, в списке Watch, будут изменяться в зависимости от шага:

```
private void ResultButton_Click(object sender, EventArgs e)
              string symbol =(string) SymbolcomboBox.SelectedItem;
              ArrayList arraysymbol = new ArrayList();
              foreach (var s in SymbolcomboBox.Items)
                  arraysymbol.Add(s);
              }
              double var1 =Convert.ToDouble(FirstVariable.Text);
              double var2 = Convert.ToDouble(SecondVariable.Text);
              switch (symbol)
                  case "+": Plus(var1, var1); break;
                  case "-": Minus(var1, var1); break;
                  case "*": Multiply(var1, var1); break;
                  case "/": Split(var1, var1); break;
              MessageBox.Show("Результат: " + с);
              MessageBox.Show("Количество символов в коллекции: "+ arraysymbol.Count);
       }
  3
00 % + 4
atch 1
Name
                               Value
                                                                                                Type
arraysymbol
                               Count = 4
                                                                                                System.(
    · [0]
                                                                                            Q - object {s
                               ..
    [1]
                                                                                            4 + object (s =
                                F#F
    · [2]
                                                                                            □ v object (s
    [3]

¬ object {s

 Raw View
                               10.0
                                                                                                double
90
```

Рис. 12.

Всего разработчику доступны четыре списка Watch, которые вызываются из меню Debug или с помощью горячих клавиш "Ctrl+Alt+W,1":

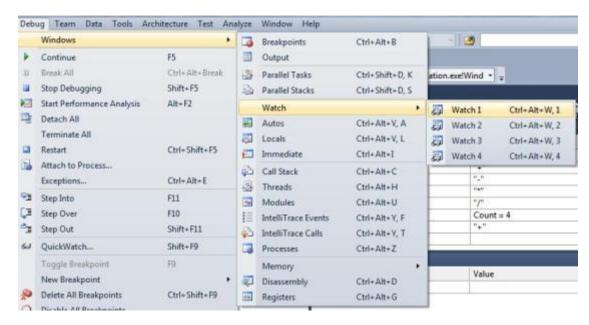


Рис. 13.

11. После завершения отладки (и если не возникло не каких ошибок) программа выдаст результаты:



Рис. 14.

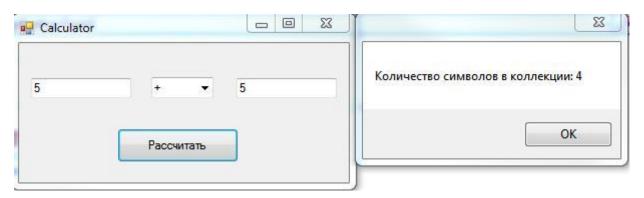


Рис. 15.

12. Повторно запустим отладку приложения и намеренно введем значения, вызывающие исключение. В нашем случае это "1a" и "2":

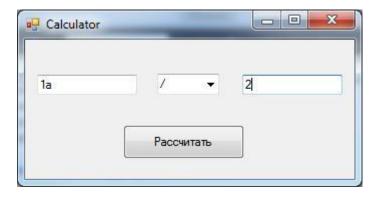


Рис. 16.

13. Если в программе не обрабатываются исключения (блок try, catch), отладчик выдаст ошибку, на строке, где возникает исключение. В нашем случае исключение, связанное с преобразованием формата типа string в формат double:

Рис. 17.

14. Для того что бы остановить отладку используйте пункт меню "Stop Debugging" меню Debug или используя сочетание клавиш "Shift+F5":

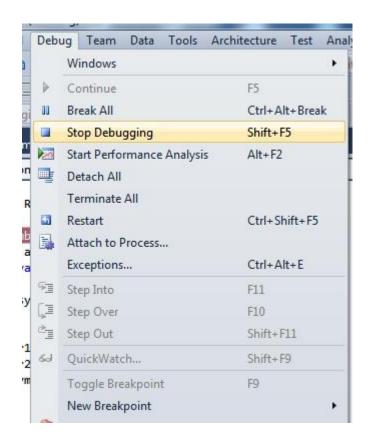


Рис. 18.