

13 События

Задание 1. При решении задач, предусмотреть обработку исключительных ситуаций. Создать 3 метода для подсчета значений трех разных функций. Создать делегат. В программе вызывать все три метода при помощи делегата. На основе своей программы создать метод, в котором делегат будет параметром.

Листинг программы:

```
namespace Space
{
    class Program
    {
        // методы для подсчета значений функций
        static double F1(double x)
        {
            return x * x;
        }
        static double F2(double x)
        {
            if (x == 0)
                throw new ArgumentException("x не может быть равен нулю");
            return 1 / x;
        }
        static double F3(double x)
        {
            if (x < 0)
                throw new ArgumentException("x не может быть отрицательным");
            return Math.Sqrt(x);
        }

        // создание делегата
        delegate double Function(double x);

        static void Main()
        {
```

					УП 2-40 01 01.37ТП.227.23.13							
Изм.	Лист	№ док	Подпись	Дата	События					Лит	Лист	Листов
Разраб.		Сорокина Е.А.										
Проверил.		Новик А.И.									61	
										Гродненский ГКТТид		
Н.контр.												
Утвердил.												

```

// создание экземпляра делегата
Function f = null;
// присвоение методов делегату
f += F1;
f += F2;
f += F3;
Console.Write("Введите x: ");
double x = Convert.ToDouble(Console.ReadLine());
// вызов всех методов через делегат
foreach (Function func in f.GetInvocationList())
{
    try
    {
        double result = func(x);
        Console.WriteLine("Функция {0}({1}) = {2}", func.Method.Name,
x, result);
    }
    catch (Exception ex)
    {
        Console.WriteLine("Функция {0}({1}) выдала исключение: {2}",
func.Method.Name, x, ex.Message);
    }
}

// метод с делегатом в качестве параметра
void CallFunctions(Function func)
{
    try
    {
        double result = func(x);
        Console.WriteLine("Функция {0}({1}) = {2}", func.Method.Name,
x, result);
    }
    catch (Exception ex)
    {
        Console.WriteLine("Функция {0}({1}) выдала исключение: {2}",
func.Method.Name, x, ex.Message);
    }
}

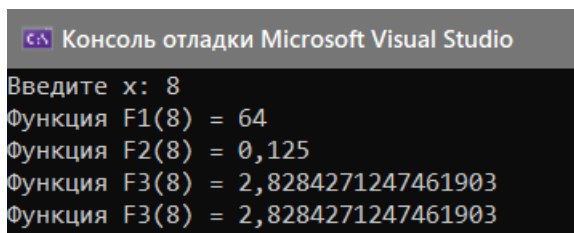
CallFunctions(f);
}
}
}

```

Таблица 13.1 – Входные и выходные данные

Входные данные	Выходные данные
8	64 0,125 2,8284271247461903 2,8284271247461903

Анализ результатов:



```

Консоль отладки Microsoft Visual Studio
Введите x: 8
Функция F1(8) = 64
Функция F2(8) = 0,125
Функция F3(8) = 2,8284271247461903
Функция F3(8) = 2,8284271247461903
    
```

Рисунок 13.1 – Результат работы программы

Задание 2. Для класса MyInfo содержащего поле name (Ваше имя) создать событие выводящее оповещение в случае изменения значения поля name.

Листинг программы:

```

namespace Space
{
    class MyInfo
    {
        private string name;

        // создание события для оповещения об изменении поля name
        public event EventHandler NameChanged;

        public string Name
        {
            get { return name; }
            set
            {
                if (value != name)
                {
                    name = value;
                    OnNameChanged();
                }
            }
        }

        // метод для вызова события
        protected virtual void OnNameChanged()
    }
}
    
```

```

    {
        NameChanged?.Invoke(this, EventArgs.Empty);
    }
}

class Program
{
    static void Main()
    {
        MyInfo info = new MyInfo();
        info.NameChanged += OnNameChanged;
        info.Name = "Екатерина";

        Console.Write("Введите новое имя: ");
        string NewName = Console.ReadLine();
        info.Name = NewName;
    }

    static void OnNameChanged(object sender, EventArgs e)
    {
        MyInfo info = (MyInfo)sender;
        Console.WriteLine("Имя установлено: " + info.Name);
    }
}
}

```

Таблица 13.2 – Входные и выходные данные

Входные данные	Выходные данные
Александра	Екатерина Александра

Анализ результатов:

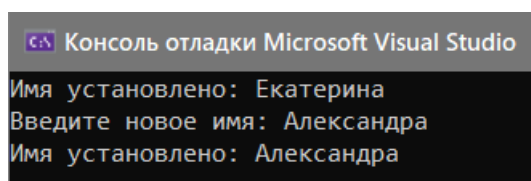


Рисунок 13.2 – Результат работы программы