

Лабораторная работа №13

Тема: Разработка делегатов и использование их в программах.

Цель: Научиться разрабатывает делегаты и использовать их в программах.

Теоретические сведения:

Делегаты в C# подобны функциональным указателям в C или C++. Они объявляются в качестве объектов, содержащих адрес метода. Делегаты — референсный класс, инкапсулирующий метод со специфичной сигнатурой и типом возвращаемого значения.

Несмотря на то, что .NET использует концепцию функционального указателя посредством делегатов, есть несколько существенных отличий:

- делегаты нечувствительны к ошибкам ввода;
- объектно-ориентированы;
- безопасны.

Делегаты C# обладают следующими свойствами:

- позволяют обрабатывать методы в качестве аргумента;
- могут быть связаны вместе;
- несколько методов могут быть вызваны по одному событию;
- тип делегата определяется его именем;
- не зависят от класса объекта, на который ссылается;
- сигнатура метода должна совпадать с сигнатурой делегата.

Пример 1. Нижеприведённый код складывает между собой два числа:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace DelegateForNotes
{
    class Program
    {
        // Объявление делегата, ссылающегося на функцию
        // с двумя параметрами и целочисленным результатом
        public delegate int AddDelegate(int num1, int num2);
```

```

static void Main(string[] args)
{
    // Создание метода делегата и передача функции Add в качестве аргумента
    AddDelegate funct1 = new AddDelegate(Add);
    // Вызов делегата
    int k = funct1(7, 2);
    Console.WriteLine("Sumation = {0}", k);
    Console.Read();
}

// Статическая функция Add с той же сигнатурой, что и у делегата
public static int Add(int num1, int num2)
{
    Console.WriteLine("I am called by Delegate");
    int sumation;
    sumation = num1 + num2;
    return sumation;
}
}

```

Выводом будет следующий результат:

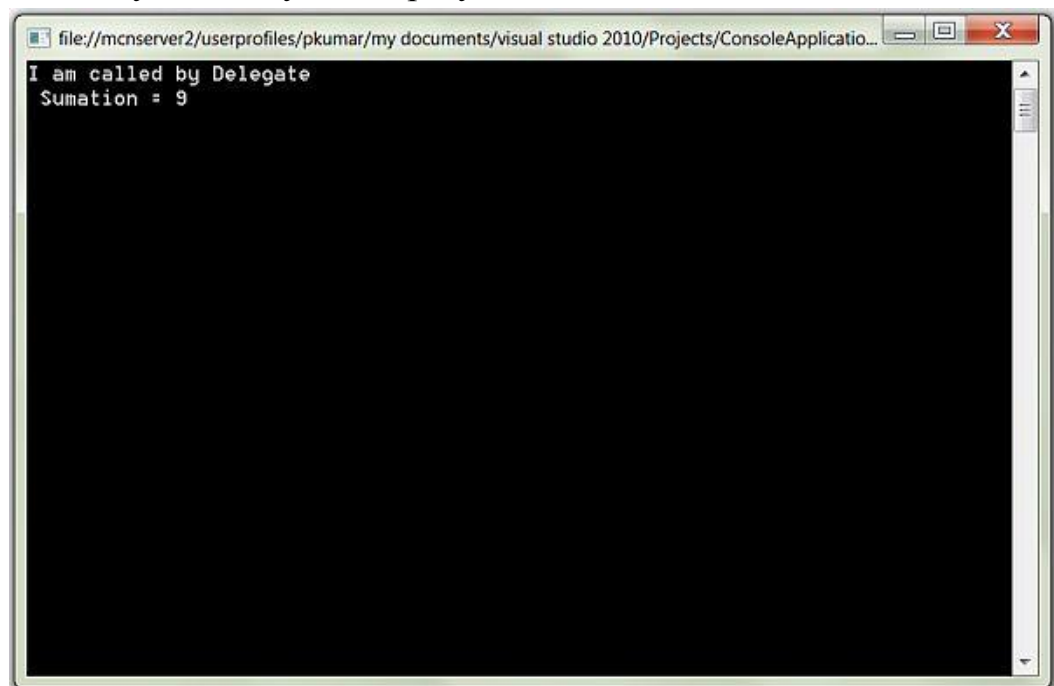


Рисунок 1 – Выполнение программы

Пример 2. Теперь рассмотрим реализацию сортировки «пузырьком» с помощью делегатов. Для этого мы создадим несколько классов:

- BubbleSortClass
- Student
- Program

BubbleSortClass.cs

Этот класс будет содержать статическую функцию Sort(), у которой нет возвращаемого значения. В качестве аргумента она принимает массив, который будет отсортирован путём сравнения пар элементов.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace DelgateForNotes
{
    public class BubbleSortClass
    {
        static public void Sort(object[] sortArray, CompareDelegate gtMethod)
        {
            for (int i = 0; i < sortArray.Length; i++)
            {
                for (int j = 0; j < sortArray.Length; j++)
                {
                    if (gtMethod(sortArray[j], sortArray[i]))
                    {
                        object temp = sortArray[i];
                        sortArray[i] = sortArray[j];
                        sortArray[j] = temp;
                    }
                }
            }
        }
    }
}
```

Student.cs

Объект этого класса и будет использован для сортировки.

```
using System;
```

```

using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace DelegateForNotes
{
    class Student
    {
        private string name;
        private int rollno;
        private int marks;
        // Инициализация объекта класса
        public Student(string name, int rollno, int marks)
        {
            this.name = name;
            this.rollno = rollno;
            this.marks = marks;
        }

        // Переопределение метода для вывода результата
        public override string ToString()
        {
            return string.Format("Name => {0}, RollNumber => {1}, Marks => {2} ", name,
rollno, marks);
        }

        // Пользовательская функция сравнение, возвращающая булево значение
        public static bool RhsIsGreater(object lhs, object rhs)
        {
            Student stdLhs = (Student)lhs;
            Student stdRhs = (Student)rhs;
            return stdRhs.marks > stdLhs.marks;
        }
    }
}

```

Program.cs

Код основной программы. Здесь создаётся делегат и выводится отсортированный массив.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace DelgateForNotes
{
    public class Program
    {
        // Объявление делегата, ссылающегося на функцию
        // с двумя параметрами и выводом булевого типа
        public delegate bool CompareDelegate(object lhs, object rhs);

        static void Main(string[] args)
        {
            // Создание массива объектов класса Student.cs
            Student[] students = {
                new Student("Mark", 1, 799),
                new Student("David", 2, 545),
                new Student("Lavish", 3, 999),
                new Student("Voora", 4, 228),
                new Student("Boll", 5, 768),
                new Student("Donna", 6, 367),
                new Student("Adam", 7, 799),
                new Student("Steve", 8, 867),
                new Student("Ricky", 9, 978),
                new Student("Brett", 10, 567)
            };

            // Создание делегата с передачей
            // статического метода класса Student в качестве аргумента
            CompareDelegate StudentCompareOp = new CompareDelegate(Student.RhsIsGreater);

            // Вызов статического метода класса BubbleSortClass,
            // передача массива объектов и делегата
            BubbleSortClass.Sort(students, StudentCompareOp);
            for(int i = 0; i < students.Length; i++)
            {
```

```

        Console.WriteLine(students[i].ToString());
    }
    Console.Read();
}
}
}

```

Мультикаст-делегаты

Делегаты, включающие в себя более одного метода, называются мультикаст-делегатами. При вызове они выполняют каждый метод в заданном порядке, позволяя таким образом связывать несколько методов в цепочку.

Для работы мультикаст-делегатов те не должны возвращать какой-либо результат. В противном случае обработается результат последнего метода цепочки.

Пример 3. Пример использования мультикаст-делегата для вывода приветствия.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace MulticastDelegate1
{
    class Program
    {
        // Объявление делегата, ссылающегося на метод
        // с результатом типа void и одним строковым аргументом
        public delegate void showDelegate(string s);
        static void Main(string[] args)
        {
            showDelegate s = Display;
            s += Show;
            s("Hello");
            s("Scott");
            Console.Read();
        }

        // Пользовательская функция для отображения результата
    }
}

```

```

public static void Display(string title)
{
    Console.WriteLine(title);
}

// Пользовательская функция для вывода
public static void Show(string title)
{
    Console.WriteLine(title);
}
}

```

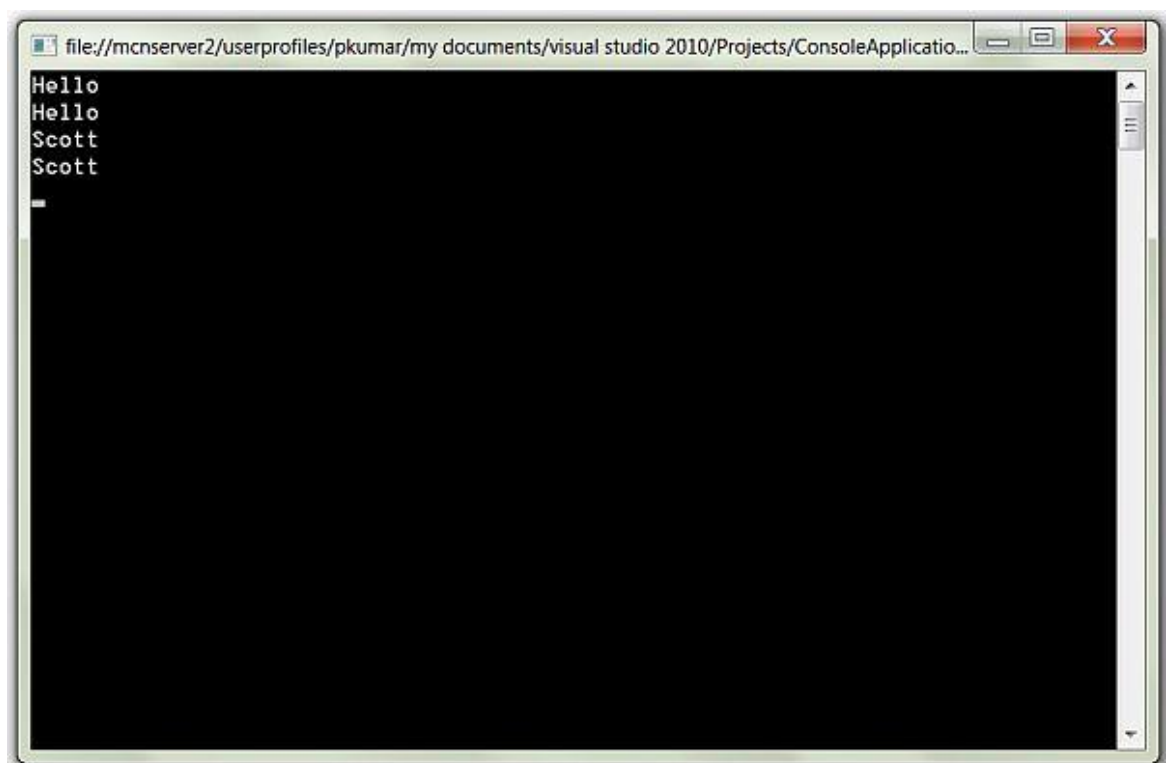


Рисунок 2 – Окно выполнения программы

Пример 4. Данный код умножает полученное число на два и возводит его же в квадрат.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace MulticastDelegate
{

```

```

// Пользовательский класс для математических операций,
// содержит два статических метода
class MathOperation
{
    // Метод умножения числа на 2
    public static void MultiplyByTwo(double d)
    {
        double res = d * 2;
        Console.WriteLine("Multiply: " + res.ToString());
    }

    // Метод возведения числа в квадрат
    public static void Squire(double e)
    {
        double res = e * e;
        Console.WriteLine("Square" + res.ToString());
    }
}

class NewProgram
{
    // Объявление делегата
    public delegate void del(double Qr);

    static void Main()
    {
        // Создание делегата
        del d = MathOperation.MultiplyByTwo;
        // Добавление метода к делегату
        d += MathOperation.Squire;
        // Вызов функции отображения,
        // в качестве аргументов – имя делегата и число типа double
        Display(d, 2.00);
        Display(d, 9.9);
        Console.ReadLine();
    }
}

// Пользовательская функция для вывода результата

```



```
static void Display(del action, double value)
{
    Console.WriteLine("Result = " + value);
    action(value);
}
}
```

Типы делегатов

System.Delegate

Смысл примерно тот же, что и у указателей на методы в C++. Однако, в C# делегаты не используют указатели, а сохраняют метаданные, определяющие целевой метод для последующего вызова. Также System.Delegate содержит экземпляр класса System.Reflection.MethodInfo — метаданные .NET для вызова метода через отражения.

Ещё одним важным аспектом System.Delegate является экземпляр объекта, для которого вызывается метод. При наличии неограниченного числа объектов с необходимой сигнатурой нам также нужно определить, какому именно объекту его назначить. Исключение — использование статического метода, определённого MethodInfo, ведь в таком случае ссылка на объект вернёт null.

System.MulticastDelegate

Этот тип делегатов позволяет использовать их для нескольких объектов. Это возможно благодаря тому, что делегаты типа System.MulticastDelegate содержат в себе экземпляр этого же класса, который создаётся при назначении объекта предыдущему. Новый экземпляр получает ссылку на следующий объект списка экземпляров делегатов. Таким образом, System.MulticastDelegate поддерживает связанный список объектов для делегатов.

Цепочка делегатов

При вызове мультикаст-делегатов каждый экземпляр в цепочке вызывается последовательно. Однако это может обернуться проблемой, если в одном из экземпляров сработало исключение или он вернул результат не void-типа.

Выполнение работы:

Вариант индивидуального задания получите у преподавателя!!!

Разработать программу с использованием делегата.

1. Вычислить значение функции $f(x)$, для введенного пользователем x :

$$f(x) = \begin{cases} (x+1)^2, \wedge x < 0 \\ \sin 3x, \wedge x \geq 0 \end{cases}$$

Модифицировать программу для вычисления следующей функции:

$$f(x) = \begin{cases} \tan 2x, \wedge x < 1 \\ (2x+10)^3, \wedge x \geq 1 \end{cases}$$

2. Найти строку в двумерном числовом массиве с максимальной суммой элементов. Модифицировать программу для нахождения строки с минимальной суммой элементов.

3. Вычислить значение функции $f(x)$, для введенного пользователем x .
 $\sum_i a_i$ – это сумма всех целых чисел до x .

$$f(x) = x^2 \times \sum_i a_i$$

Модифицировать программу для вычисления следующей функции,
где $\prod_i a_i$ - это произведение всех четных чисел до x .

$$f(x) = x^2 \times \prod_i a_i$$

4. Вычислить значение функции $f(x)$, для введенного пользователем x :

$$f(x) = \begin{cases} (x^8 - x^6 + 7)^9, \wedge x < 0 \\ \cos 4x, \wedge x \geq 0 \end{cases}$$

Модифицировать программу для вычисления следующей функции:

$$f(x) = \begin{cases} \sin(7x+2), \wedge x < 4 \\ \sqrt{x^2-16}, \wedge x \geq 4 \end{cases}$$

5. Вычислить значение функции $f(x)$, для введенного пользователем x .
 $\sum_i a_i$ – это сумма всех натуральных чисел до x .

$$f(x) = \frac{1}{x} \times \sum_i a_i$$

Модифицировать программу для вычисления следующей функции,
где $\prod_i a_i$ - это произведение всех нечетных чисел до x .

$$f(x) = \frac{1}{x} \times \prod_i a_i$$

6. Найти столбец в двумерном числовом массиве с максимальным произведением нечетных элементов. Модифицировать программу для нахождения столбца с минимальным произведением нечетных элементов.

7. Вычислить значение функции $f(x)$, для введенного пользователем x .
 $\sum_i a_i$ - это сумма всех четных чисел до x .

$$f(x) = \sqrt{x} \times \sum_i a_i$$

Модифицировать программу для вычисления следующей функции,
где $\prod_i a_i$ - это произведение всех натуральных чисел до x .

$$f(x) = \sqrt{x} \times \prod_i a_i$$

8. Вычислить значение функции $f(x)$, для введенного пользователем x :

$$f(x) = \begin{cases} \cos 20x, & x < -4 \\ \sqrt{x^8 + 5x}, & -4 \leq x < 4 \\ \frac{x}{x-4}, & x \geq 4 \end{cases}$$

Модифицировать программу для вычисления следующей функции:

$$f(x) = \begin{cases} \frac{1}{x^2 - 4}, & x < -3 \\ \sqrt{x^2 - 2}, & -3 \leq x < 5 \\ 3 \cos(5x + 6), & x \geq 5 \end{cases}$$

9. Вычислить значение функции $f(x)$, для введенного пользователем x .
 $\sum_i a_i$ - это сумма всех нечетных чисел до x .

$$f(x) = 2x \times \sum_i a_i$$

Модифицировать программу для вычисления следующей функции,
 где $\prod_i a_i$ - это произведение всех целых чисел до x .

$$f(x) = 2x \times \prod_i a_i$$

10. Составить делегат определения минимума функции $y = f(x)$, заданной в дискретных точках отрезка $[a, b]$ с постоянным шагом h .

$$\sin \frac{x}{i}$$

Используя делегат, найти минимумы функций $y = \frac{x}{i}$.

11. Составить делегат упорядочения элементов одномерного массива по убыванию их значений методом выбора максимального элемента, используя метод нахождения номера максимального элемента в последовательности чисел. Упорядочить по убыванию введенный пользователем массив.

12. Вычислить значение функции $f(x)$, для введенного пользователем x .
 $\sum_i a_i$ – это сумма всех четных чисел до x .

$$f(x) = \frac{x^4}{2} \times \sum_i a_i$$

Модифицировать программу для вычисления следующей функции,
 где $\prod_i a_i$ - это произведение всех нечетных чисел до x .

$$f(x) = \frac{x^4}{2} \times \prod_i a_i$$

13. Вычислить значение функции $f(x)$, для введенного пользователем x :

$$f(x) = \begin{cases} -3 \sin(2x), & x < -7 \\ \sqrt{x^3 - 5}, & -7 \leq x < 0 \\ \frac{1}{x}, & x \geq 0 \end{cases}$$

Модифицировать программу для вычисления следующей функции:

$$f(x) = \begin{cases} \frac{1}{x^2}, & x < 1 \\ \frac{2x}{x^2 - 2}, & 1 \leq x < 5 \\ \cos(5x), & x \geq 5 \end{cases}$$

14. Найти строку в двумерном числовом массиве с максимальной суммой четных элементов. Модифицировать программу для нахождения строки с минимальной суммой четных элементов.

15. Составить делегат для вычисления дуги по формуле:

$$L = \sum_{i=0}^{n-1} \sqrt{(f(x_{i+1}) - f(x_i))^2 + (x_{i+1} - x_i)^2}$$

Образованной функцией $f(x)$ в интервале (a, b) . Используя созданный делегат определить самую длинную из дуг, образованных функциями $f(x) = x^2 + 2\ln(1+x^2)$, $f(x) = x \ln|x^2 - 2x|$, $f(x) = (x^2 + 2x - 3)e^{-x}$ на интервале (a, b) .

16. Составить делегат определения максимума функции $y = f(x)$, заданной в дискретных точках отрезка $[a, b]$ с постоянным шагом h .

$$\begin{matrix} x \\ \cos \\ \int \end{matrix}$$

Используя делегат, найти максимумы функций $y = e^{-x}$.

17. Найти столбец в двумерном числовом массиве с максимальным произведением элементов. Модифицировать программу для нахождения столбца с минимальным произведением элементов.

18. Составить делегат вычисления определенного интеграла по формуле прямоугольников:

$$\int_a^b f(x) dx \approx \frac{b-a}{n} \sum_{i=0}^{n-1} f\left(a + i \frac{b-a}{n}\right)$$

Используя делегат вычислить:

$$\begin{aligned} & (3x - \sin 2x) dx - \int_0^{\pi} (e^{-2x} - 2x + 1) dx \\ & \int_0^{2\pi} \end{aligned}$$

19. Составить делегат вычисления коэффициента корреляции двух случайных величин x и y на основании выборок $x=(x_1, x_2, \dots, x_n)$, $y=(y_1, y_2, \dots, y_n)$ по формуле

$$R = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{j=1}^n (y_j - \bar{y})^2}}$$

Для вычисления $\bar{x} = \sum_{i=1}^n x_i$, $\bar{y} = \sum_{i=1}^n y_i$ использовать отдельный метод. Найти R для введенных пользователем выборок двух случайных величин x , y .

20. Для функции $y = \frac{\tan x}{2} e^{-x}$ вывести на экран значения от a до b с шагом h ; a, b, h – вводятся пользователем. Для вывода на экран применить следующее форматирование:

|x|y|

| 1 | 1 |

| 2 | 2 |

Модифицировать программу так, чтобы результаты выводились следующим образом:

x= 1, y = 1

x = 2, y = 2

Содержание отчета:

1. Номер и тема лабораторной работы.
2. Цель лабораторной работы.
3. Техническое оснащение.
4. Скриншоты выполнения примеров.
5. При выполнении индивидуальных заданий в отчет внести изображение кода программы и окно выполнения программы.
6. Вывод по лабораторной работе.