

Лабораторная работа № 13.2

Тема: Разработка программ реализующих механизм позднего связывания функций при помощи делегатов.

Цель работы: Формирование умений и навыков использования делегатов в программе.

Время на выполнение работы: 2 часа

Этапы работы:

- I.** Ознакомиться с теоретическими сведениями
- II.** Выполнить задания.
- III.** Ответить на контрольные вопросы

I Теоретические сведения

Основы программирования на C#

Делегаты

Делегат — это объект, имеющий ссылку на метод. Делегат позволяет выбрать вызываемый метод во время выполнения программы. Фактически значение делегата – это адрес области памяти, где находится точка входа метода.

Важным свойством делегата является то, что он позволяет указать в коде программы вызов метода, но фактически вызываемый метод определяется во время работы программы, а не во время компилирования.

Делегат объявляется с помощью ключевого слова `delegate`, за которым указывается тип возвращаемого значения, имя делегата и список параметров вызываемых методов.

Синтаксис:

delegate *тип_возвращаемого_значения* *имя_делегата*
(*список_параметров*);

Характерной особенностью делегата является возможность его

использования для вызова любого метода, который соответствует подписи делегата. Это дает возможность определить во время выполнения программы, какой из методов должен быть вызван. Вызываемый метод может быть методом экземпляра, ассоциированным с объектом, либо статическим методом, ассоциированным с классом. Метод можно вызвать только тогда, когда его подпись соответствует подписи делегата.

Многоадресность делегатов

Многоадресность — это способность делегата хранить несколько ссылок на различные методы, что позволяет при вызове делегата инициировать эту цепочку методов.

Для создания цепочки методов необходимо создать экземпляр делегата, и пользуясь операторами `+` или `+=` добавлять методы к цепочке. Для удаления метода из цепочки используется оператор `-` или `-=`. Делегаты, хранящие несколько ссылок, должны иметь тип возвращаемого значения `void`.

Пример:

```
// Создадим делегат
delegate int IntOperation (int i, int j);

static class Calculator
{
    // Организуем ряд методов
    static int Sum(int x, int y)
    {
        return x + y;
    }

    static int Prz(int x, int y)
    {
        return x * y;
    }

    static int Del(int x, int y)
    {
        return x / y;
    }
}

class Program
{
    static void Main()
    {
        // Сконструируем делегат
        IntOperation op1 = new IntOperation(Sum);
    }
}
```

```

        int result = op1(5, 10);
        Console.WriteLine("Сумма: " + result);

        // Изменим ссылку на метод
        op1 = new IntOperation(Prz);
        result = op1(5, 10);
        Console.WriteLine("Произведение: " + result);

        Console.ReadLine();
    }
}

```

II Задание

Общее задание:

Создать меню(**в примере его нет**) для выбора операции. Операция должна при помощи делегатов подключаться динамически на этапе выполнения.

Варианты:

1. Создать класс телевизор с методами:

- Показать веселый смайлик
- Показать грустный смайлик
- Показать задумчивый смайлик.

Обязательно чтобы он был в телевизоре на консоле.

2. Создать класс агрегатор со следующими возможностями:

- Посчитать среднее арифметическое любого количества чисел
- Посчитать среднее геометрическое любого количества чисел

3. Создать класс матрица(ступенчатый массив) со следующими возможностями:

- Посчитать диагональную сумму чисел
- Посчитать седловую точку (максимум в столбце, минимум в строке)
- Посчитать среднее арифметическое всей матрицы чисел

4. Создать класс матрица (многомерный массив) со следующими возможностями:

- Посчитать диагональную сумму чисел
- Посчитать седловую точку (максимум в столбце, минимум в строке)

- Посчитать среднее арифметическое всей матрицы чисел
5. Создать класс агрегатор со следующими возможностями:
 - Посчитать сумму любого количества чисел
 - Посчитать произведение любого количества чисел
 - Найти максимальное среди любого количества чисел
 6. Создать класс агрегатор со следующими возможностями:
 - Посчитать среднее арифметическое любого количества чисел
 - Посчитать количества переданных в метод чисел
 - Найти минимальное среди любого количества чисел
 7. Создать класс архитектор со следующими возможностями:
 - Нарисовать домик в анфас на заданное количество точек
 - Нарисовать домик в профиль на заданное количество точек
 8. Создать класс обработчик матриц со следующими возможностями:
 - Отсортировать строки любой матрица типа `int[][]` так, чтобы вначале были те у которых сумма всех элементов строки меньше.
 - Отсортировать строки любой матрица типа `int[][]` так, чтобы вначале были те у которых произведение всех элементов строки меньше.

Обязательно матрица передается в методы в качестве параметра(и меняется в нем!)

9. Создать класс РисовательТаблиц со следующими возможностями:
 - Нарисовать таблицу синуса от а до б
 - Нарисовать таблицу Синус квадрат + косинус квадрат от а до б
 - Нарисовать таблицу $1/x$ от а до б(перехватить исключение)

Литература

1. Полный справочник по C#. Г. Шилдт. Издательский дом «Вильямс», 2004.
2. C# в подлиннике. Наиболее полное руководство. Х.Дейтел.
3. C# в задачах и примерах. Культин. Н.Б.
4. C# учебный курс. Г.Шилдт. СПб.: Питер, 2002.
5. C# программирование на языке высокого уровня Павловская Т.А. СПб.: БХВ-Петербург.