Выполнили:

студенты группы 19ВВ2

Гусев Д.О.

Кубасов И.М.

Приняли:

Синев М.П.

Дорошенко И.Н.

Пенза 2021

ОТЧЕТ

по лабораторной работе №4

по курсу «Технологии программирования»

на тему «Делегаты»

Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

### Цель работы

Изучение способов работы и использования делегатов в языке C#.

### Лабораторное задание

Лабораторное задание выполняется на базе лабораторной работы №1. Из лабораторной работы используется класс Derived1.

Требуется объявить 2 делегата для функций Input и Output.

Требования к программе:

* В основной функции программы необходимо создать 3-5 объектов класса **Derived1**.
* Создать 2 объекта делегата для функций Input и Output. Присвоить им адреса функций 1-го созданного объекта класса **Derived1**. Вызвать функции с использованием делегатов.
* Выполнить группирование делегатов. Для ранее созданных объектов делегатов добавить адреса функций оставшихся объектов класса **Derived1** (со 2 по 3-5). Вызвать функции с использованием делегатов.
* Выполнить разгруппирование делегатов. Удалить из объектов делегатов все адреса функций. Вызвать функции с использованием делегатов (убедиться, что ни одна функция не вызвана).

### Листинг

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab4oop

{

public delegate void DelegateOutput();

public delegate void DelegateInput();

class Phone

{

private string m\_producer;

public string Producer

{

get

{

return this.m\_producer;

}

set

{

this.m\_producer = value;

}

}

private string m\_model;

public string Model

{

get

{

return this.m\_model;

}

set

{

this.m\_model = value;

}

}

private double m\_size;

protected double Size

{

get

{

return this.m\_size;

}

set

{

this.m\_size = value;

}

}

private int m\_price;

public int Price

{

get

{

return this.m\_price;

}

set

{

this.m\_price = value;

}

}

public Phone()

{

Producer = "";

Model = "";

Size = -1;

Price = -1;

}

public Phone(string producer = "Samsung", string model = "A50", double size = 22.2, int price = 19992)

{

Producer = producer;

Model = model;

Size = size;

Price = price;

}

public void Input()

{

Console.WriteLine("Введите производителя: ");

Producer = Console.ReadLine();

Console.WriteLine("Введите модель: ");

Model = Console.ReadLine();

Console.WriteLine("Введите размер экрана: ");

Size = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Введите цену: ");

Price = Convert.ToInt32(Console.ReadLine());

}

public void Output()

{

if (Size == -1)

{

Console.WriteLine("Объект не инициализирован!");

}

else

{

Console.WriteLine("Производитель: " + Producer + "\n" + "Модель: " + Model + "\n" + "Размер экрана: " + Size + "\n" + "Цена: " + Price);

}

}

~Phone()

{

Producer = "";

Model = "";

Size = -1;

Price = -1;

}

};

class Derived1 : Phone

{

private string m\_country;

public string Country

{

get

{

return this.m\_country;

}

set

{

m\_country = value;

}

}

public Derived1()

{

Country = "";

}

public Derived1(string country = "Япония", string producer = "Samsung", string model = "A50", double size = 22.2, int price = 19992)

{

Producer = producer;

Model = model;

Size = size;

Price = price;

Country = country;

}

public void Input()

{

Console.WriteLine("Введите страну: ");

Country = Console.ReadLine();

base.Input();

}

public void Output()

{

base.Output();

if (Size != -1)

{

Console.WriteLine("Страна: " + Country);

}

}

~Derived1()

{

Country = "";

}

};

class Derived2 : Phone

{

private int m\_year;

public int Year

{

get

{

return this.m\_year;

}

set

{

m\_year = value;

}

}

public Derived2()

{

Year = -1;

}

public Derived2(int year = 2018, string producer = "Samsung", string model = "A50", double size = 22.2, int price = 19992)

{

Producer = producer;

Model = model;

Size = size;

Price = price;

Year = year;

}

public void Input()

{

Console.WriteLine("Введите год: ");

Year = Convert.ToInt32(Console.ReadLine());

base.Input();

}

public void Output()

{

base.Output();

if (Size != -1)

{

Console.WriteLine("Год: " + Year);

}

}

~Derived2()

{

Year = -1;

}

};

class Derived3 : Derived2

{

private string m\_city;

public string City

{

get

{

return this.m\_city;

}

set

{

m\_city = value;

}

}

public Derived3()

{

City = "";

}

public Derived3(string city = "Токио", int year = 2020, string producer = "Samsung", string model = "A50", double size = 22.2, int price = 19992)

{

Producer = producer;

Model = model;

Size = size;

Price = price;

Year = year;

City = city;

}

public void Input()

{

Console.WriteLine("Введите город: ");

City = Console.ReadLine();

base.Input();

}

public void Output()

{

base.Output();

if (Size != -1)

{

Console.WriteLine("Город: " + City);

}

}

public static Derived3 operator &(Derived3 obj, int a)

{

string temp\_a = Convert.ToString(a);

int genre\_len = obj.City.Length - 5;

if (genre\_len > 0)

{

obj.City = obj.City.Remove(genre\_len);

obj.City += temp\_a;

}

else

{

obj.City = temp\_a;

}

return obj;

}

public static Derived3 operator &(Derived3 obj, string str)

{

string temp\_str = str;

int City\_len = obj.City.Length - 5;

if (City\_len > 0)

{

obj.City = obj.City.Remove(City\_len);

obj.City += temp\_str;

}

else

{

obj.City = temp\_str;

}

return obj;

}

~Derived3()

{

City = "";

}

};

internal class Program

{

static int Main(string[] args)

{

DelegateInput delegateInput;

DelegateOutput delegateOutput;

Derived1 obj1 = new Derived1();

delegateInput = obj1.Input;

delegateOutput = obj1.Output;

Console.WriteLine("Использование методов Input и Output с использованием делегатов \n");

delegateInput();

delegateOutput();

Console.WriteLine("\nГруппируем делегаты для еще нескольких объектов \n");

Derived1 obj2 = new Derived1();

Derived1 obj3 = new Derived1();

delegateInput = delegateInput + obj2.Input + obj3.Input;

delegateOutput = delegateOutput + obj2.Output + obj3.Output;

delegateInput();

delegateOutput();

delegateInput = delegateInput - obj3.Input - obj2.Input - obj1.Input;

delegateOutput = delegateOutput - obj3.Output - obj2.Output - obj1.Output;

Console.WriteLine("Разгруппировали делегаты\n");

if (delegateInput == null)

{

Console.WriteLine("Делегат delegateInput пуст");

}

else

{

delegateInput();

}

if (delegateOutput == null)

{

Console.WriteLine("Делегат delegateOutput пуст");

}

else

{

delegateOutput();

}

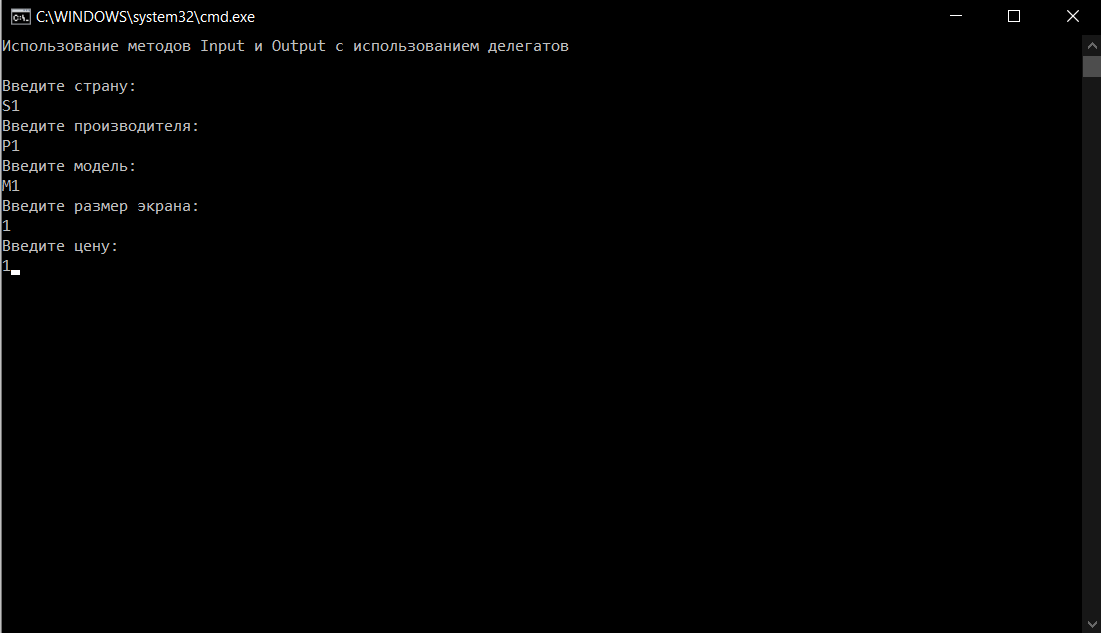
return 0;

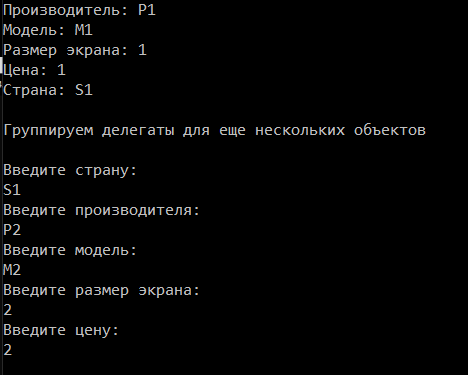
}

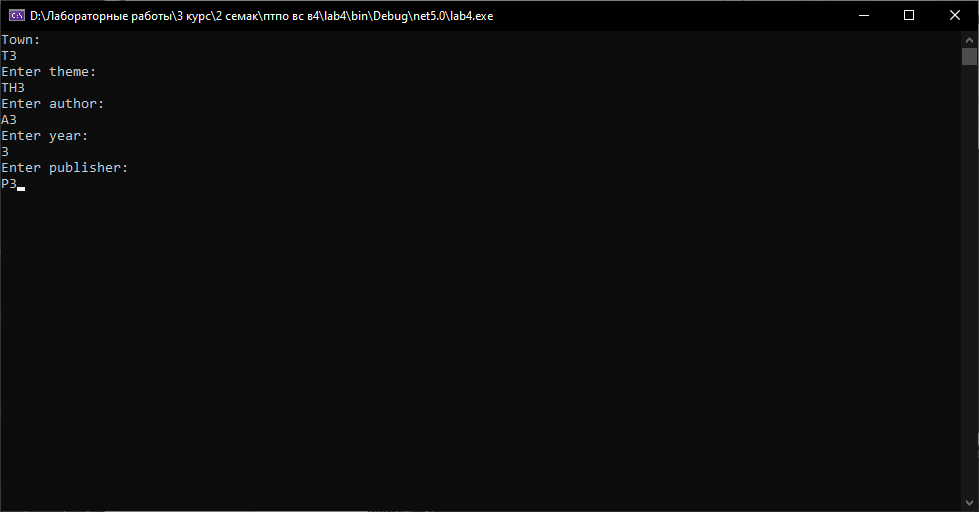
}

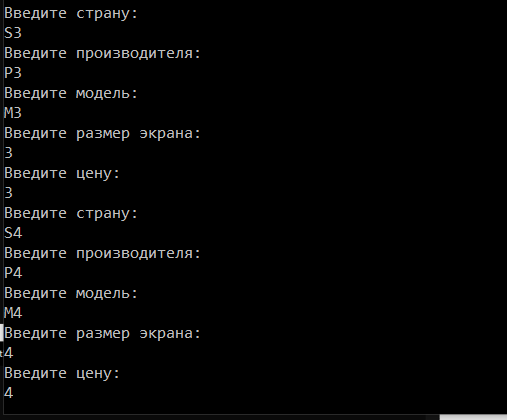
}

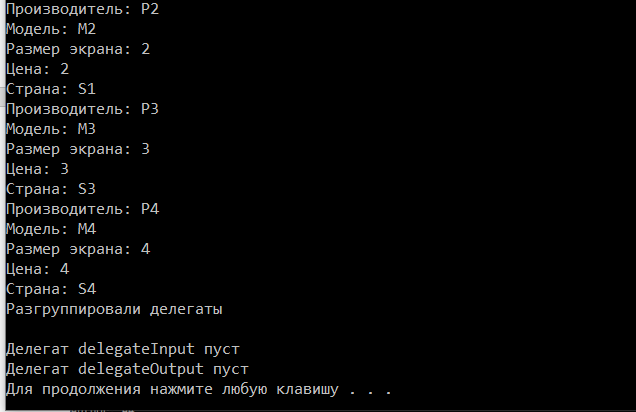
**Результаты работы программы**











### Вывод:

Мы изучили способы работы и использования делегатов в языке C#.