Выполнили:

студенты группы 19ВВ2

Гусев Д.О.

Кубасов И.М.

Приняли:

Синев М.П.

Дорошенко И.Н.

Пенза 2021

ОТЧЕТ

по лабораторной работе №1

по курсу «Технологии программирования»

на тему «Класс и Наследование»

Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

### Цель работы

Изучение основных особенностей синтаксиса и семантики языка C#.

### Лабораторное задание

Разработать консольное приложение, в котором должен быть реализован набор классов в соответствии с вариантом задания.

Должны быть разработаны базовый класс (Base) не менее 3 производных классов (Derived1, Derived2 и Derived3). Каждый производный класс должен расширять базовый. В каждый производный класс необходимо добавить переменные и функции, уточняющие базовый класс.

Структура классов должна быть следующей:

* **Base** : **Derived1**
* **Base** : **Derived2**
* **Derived2** : **Derived3**

Производные классы **Derived1** и **Derived2** должны быть пронаследованы от базового класса (**Base**). Класс **Derived3** должен быть пронаследован от производного класса **Derived2**.

Требования к классам:

* Класс должен содержать конструктор без параметров. В конструкторе должны инициализироваться переменные класса значением по умолчанию.
* Класс должен содержать два или более конструкторов с параметрами. В конструкторах должны инициализироваться переменные класса значениями, которые переданы в параметрах конструктора. Если для какой-либо переменной класса в параметрах конструктора не предусмотрен параметр, то переменная должна инициализироваться значением по умолчанию.
* Класс должен содержать деструктор. В деструкторе все переменные класса должны быть затерты (присвоены недействительные значения).
* Класс должен содержать три или более методов класса (функций). Методы должны быть реализованы в трех секциях: public, protected и private. В каждом методе должно быть обращение к одной или более переменной класса.
* Класс должен содержать три или более (в соответствии с вариантом задания) переменных класса. Переменные класса должны быть реализованы в трех секциях: public, protected и private.
* Класс должен содержать одну или более переменную со строковыми данными.
* В классе должны быть реализованы два метода: Input и Output. Метод Input должен выполнять ввод данных класса с клавиатуры. Метод Output должен печатать на экране данные класса. Методы должны располагаться в секции public. В производных классах (**Derived1**, **Derived2** и **Derived3**) методы должны использовать вызовы из родительского класса. Т.е. в производном классе в функциях первоначально должны быть вызваны функции базового класса (для **Derived1** и **Derived2** функции из класса **Base**, для **Derived3** функции из класса **Derived2**), а затем выполнен ввод и вывод данных текущего класса.

Требования к программе:

* В основной функции программы создать объект класса с вызовом конструктора без параметров.
* В основной функции программы создать несколько объектов класса с вызовом конструкторов с параметрами.
* Для любого созданного объекта класса вызвать функции Input, Output и все остальные методы, реализованные в классе. Вызов можно выполнять как из основной функции программы, так и из любой функции класса (это касается функций, объявленных в секциях protected и private).

### Варианты лабораторных заданий

3. Класс «Телефон»

Листинг

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab1oop

{

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;

}

private void Input()

{

Console.WriteLine("Введите город: ");

City = Console.ReadLine();

base.Input();

}

public void Inputt()

{

Input();

}

protected void Output()

{

base.Output();

if (Size != -1)

{

Console.WriteLine("Город: " + City);

}

}

public void Outputt()

{

Output();

}

~Derived3()

{

City = "";

}

};

internal class Program

{

static int Main(string[] args)

{

Phone phone = new Phone();

phone.Output(); //Вывод объекта созданного с пом. конструктора без пораметров

Console.WriteLine("\n");

Derived3 a = new Derived3("Варшава"); //– В основной функции программы создать несколько объектов класса с вызовом конструкторов с параметрами.

a.Outputt();

Console.WriteLine("\n");

Derived2 c = new Derived2(2019);

c.Output();

Console.WriteLine("\n");

a.Inputt();

Console.WriteLine("\n");

a.Outputt();

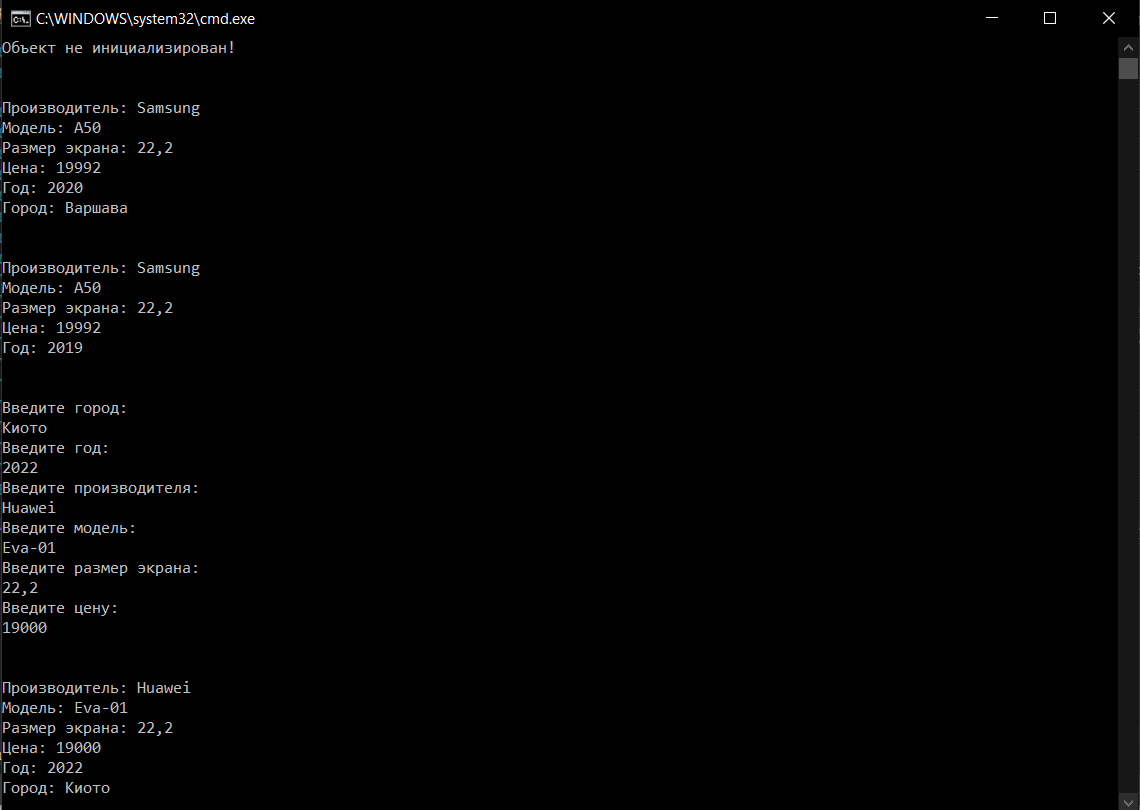
return 0;

}

}

}

### Результаты работы программы



### Вывод:

Мы изучили основные особенности синтаксиса и семантики языка C#.