Выполнили:

студенты группы 19ВВ2

Гусев Д.О.

Кубасов И.М.

Приняли:

Синев М.П.

Дорошенко И.Н.

Пенза 2022

ОТЧЕТ

по лабораторной работе №2

по курсу «Технологии программирования»

на тему «Интерфейсы и Виртуальные функции»

Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

### Цель работы

Изучение правил описания интерфейсов и механизма перегрузки операторов.

### Лабораторное задание

Лабораторная работа выполняется на базе работы №1.

Необходимо реализовать один интерфейс Interface1 и один абстрактный класс Abstruct1. Interface1 должен включить описание функции Input. Abstruct1 должен включить описание функции Output.

Расширить наследование класса Base, добавив к нему Interface1 и Abstruct1. В каждом классе Derived1, Derived2 и Derived3, реализовать функции Input и Output.

### Листинг

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab2oop

{

interface Interface

{

void Input();

}

abstract class Abstruct

{

public abstract void Output();

};

class Phone : Abstruct, Interface

{

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 override 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 override 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 override 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 override void Output()

{

base.Output();

if (Size != -1)

{

Console.WriteLine("Город: " + City);

}

}

~Derived3()

{

City = "";

}

};

internal class Program

{

static int Main(string[] args)

{

Phone phone = new Phone();

phone.Output();

Console.WriteLine("\n");

Derived3 a = new Derived3("Варшава");

a.Output();

Console.WriteLine("\n");

Derived2 c = new Derived2(2020);

c.Output();

Console.WriteLine("\n");

a.Input();

Console.WriteLine("\n");

a.Output();

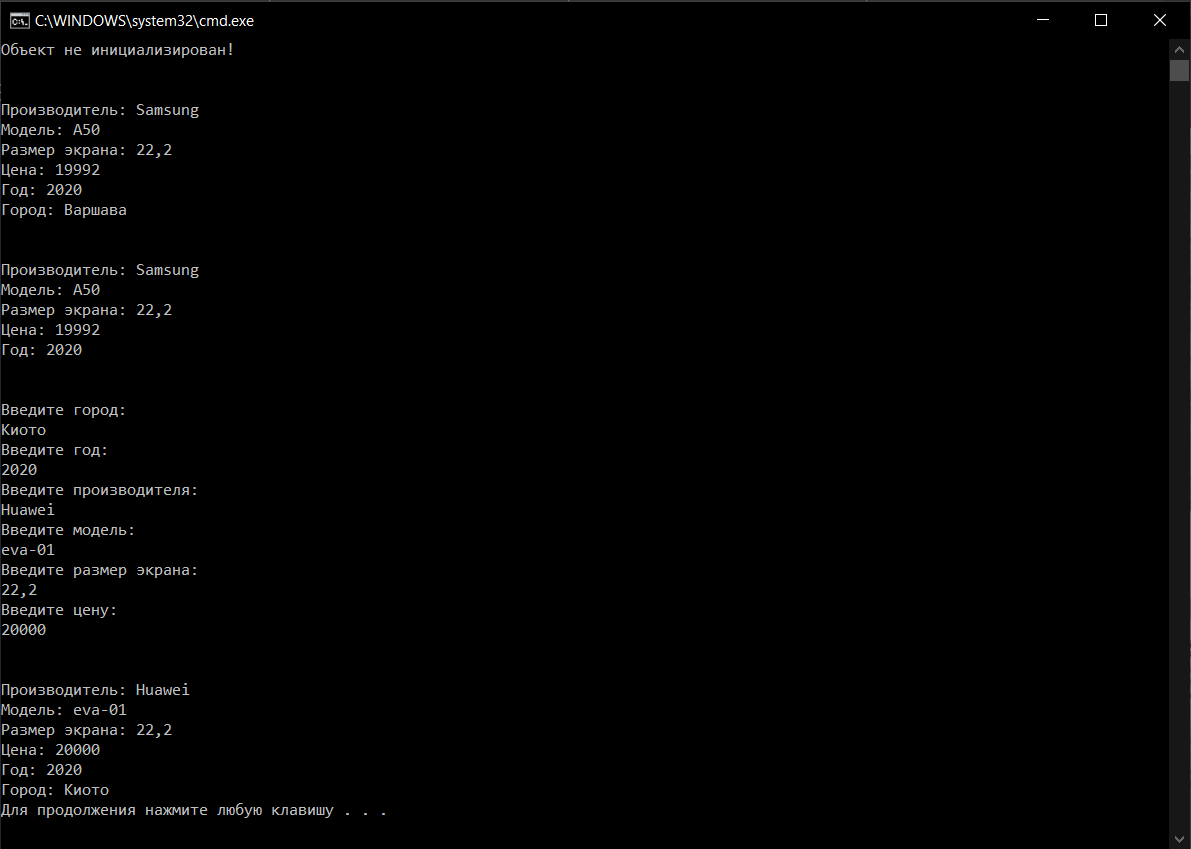
return 0;

}

}

}

### Результаты работы программы



### Вывод:

Мы изучили правила описания интерфейсов и механизма перегрузки операторов.