Пенза 2020

Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

Выполнил:

студенты группы 17ВВ1

Кокин Денис

Беленков Никита

Принял:

Дорошенко И.Н.

**ОТЧЁТ**

по лабораторной работе №2

по курсу «Технологии программирования»

на тему «Интерфейсы и Виртуальные функции»

**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 программы**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Confrontation

{

class Township : Locality

{

public int SchoolCount;

public override void output()

{

base.output();

Console.WriteLine("A count of schools: " + this.SchoolCount);

}

public void input()

{

base.input();

Console.Write("Enter a count of schools: ");

SchoolCount = Int32.Parse(Console.ReadLine());

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Confrontation

{

abstract class OutputAbstract

{

public abstract void output();

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Confrontation

{

interface InputInterface

{

void input();

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Confrontation

{

class City : Locality

{

public bool isRegionCenter;

public override void output()

{

base.output();

Console.WriteLine("This is a region center: " + this.isRegionCenter);

}

public void input()

{

base.input();

Console.Write("Is this a region center?: ");

isRegionCenter = System.Convert.ToBoolean(Console.ReadLine());

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Confrontation

{

class Locality : OutputAbstract, InputInterface

{

public String Name;

public String Region;

public Locality()

{

Population = 0;

Budget = 0;

Code = 0;

}

public Locality(int Population, int Budget, int Code, String Name = "Unnamed", String Region = "Unnamed")

{

this.Population = Population;

this.Budget = Budget;

this.Code = Code;

this.Name = Name;

this.Region = Region;

}

public Locality(String Name, String Region, int Population = 0, int Budget = 0, int Code = 0)

{

this.Population = Population;

this.Budget = Budget;

this.Code = Code;

this.Name = Name;

this.Region = Region;

}

public void ClassInfo()

{

Console.WriteLine("Full region name: " + this.Name + " in " + this.Region);

this.AbleBodied();

this.BudgetToUSD();

}

~Locality()

{

Name = null;

Region = null;

Population = 0;

Budget = 0;

Code = 0;

}

public void input()

{

Console.Write("Input a name of the region: ");

this.Name = Console.ReadLine();

Console.Write("Input a regional center of the region: ");

this.Region = Console.ReadLine();

Console.Write("Input a population of the region: ");

this.Population = Int32.Parse(Console.ReadLine());

Console.Write("Input a budget of the region: ");

this.Budget = Int32.Parse(Console.ReadLine());

Console.Write("Input a code of the region: ");

this.Code = Int32.Parse(Console.ReadLine());

}

public override void output()

{

Console.WriteLine("A name of the region: " + this.Name);

Console.WriteLine("A regional center of the region: " + this.Region);

Console.WriteLine("A population of the region: " + this.Population);

Console.WriteLine("A budget of the region: " + this.Budget);

Console.WriteLine("A code of the region: " + this.Code);

}

protected int Population;

protected int Code;

protected int Budget;

protected void AbleBodied()

{

Console.WriteLine("Able-bodied population: " + (this.Population / 2));

}

private void BudgetToUSD()

{

Console.WriteLine("Budget of locality: " + (this.Budget / 70) + " USD");

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Confrontation

{

class Village : Township

{

public int ShopCount;

public override void output()

{

base.output();

Console.WriteLine("A count of shops: " + this.ShopCount);

}

public void input()

{

base.input();

Console.Write("Enter a count of shops: ");

ShopCount = Int32.Parse(Console.ReadLine());

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Confrontation {

class Program{

static void Main(string[] args){

Locality SampleLocality = new Locality();

SampleLocality.input();

SampleLocality.output();

Township SampleTownship = new Township();

SampleTownship.input();

SampleTownship.output();

Village SampleVillage = new Village();

SampleVillage.input();

SampleVillage.output();

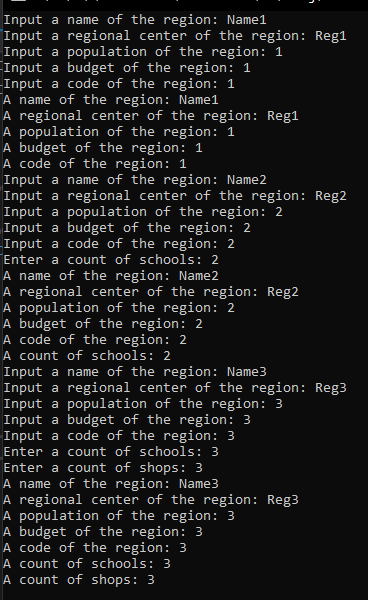
Console.Read();

}

}

}

**Результаты работы программы**



**Вывод:** студенты изучили механизмы сохранения и восстановления объектов данных. Получили навыки разработки самовосстанавливающихся структур данных.