Выполнили:

студенты группы 19ВВ2

Гусев Д.О.

Кубасов И.М.

Приняли:

Синев М.П.

Дорошенко И.Н.

Пенза 2021

ОТЧЕТ

по лабораторной работе №5

по курсу «Технологии программирования»

на тему «Стандартные коллекции и файловые потоки»

Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

### Цель работы

Изучение стандартных коллекций и классов поддержки файловых потоков, входящих в состав Framework Class Library.

### Лабораторное задание

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

В работе требуется реализовать коллекцию элементов в соответствии с вариантом задания. В качестве элементов коллекции должен использоваться класс из лабораторной работы №1.

Необходимо сформировать коллекцию элементов, сохранить ее в файл, а затем восстановить коллекцию из файла.

### Листинг

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab5oop

{

public delegate void DelegateOutput();

public delegate void DelegateInput();

public class Phones

{

private const string file = "Serialized.txt";

private int key;

private int BooksListLen;

public SortedList<int, Derived1> BooksList { get; set; } = new SortedList<int, Derived1>();

public Phones()

{

key = 0;

BooksListLen = 3;

}

public Phones(int Len)

{

key = 0;

BooksListLen = Len;

}

public void AddToList(Derived1 obj)

{

BooksList.Add(key, obj);

key++;

}

public void PhonesListSerialization()

{

StreamWriter streamW = new StreamWriter(file, true);

foreach (KeyValuePair<int, Derived1> pair in BooksList)

{

pair.Value.Derived1Serialization(streamW);

}

streamW.Close();

}

public void PhonesListDeserialization()

{

StreamReader streamR = new StreamReader(file);

for (int i = 0; i < BooksListLen; ++i)

{

Derived1 obj = new Derived1();

obj.Derived1Deserialization(streamR);

AddToList(obj);

}

streamR.Close();

}

public void PhonesListOutput()

{

foreach (KeyValuePair<int, Derived1> pair in BooksList)

{

pair.Value.Output();

}

}

~Phones()

{

key = 0;

}

};

public 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 string m\_size;

protected string Size

{

get

{

return this.m\_size;

}

set

{

this.m\_size = value;

}

}

private string m\_price;

public string Price

{

get

{

return this.m\_price;

}

set

{

this.m\_price = value;

}

}

public Phone()

{

Producer = "";

Model = "";

Size = "";

Price = "";

}

public Phone(string producer = "Samsung", string model = "A50", string size = "22.2", string 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 = Console.ReadLine();

Console.WriteLine("Введите цену: ");

Price = Console.ReadLine();

}

public void Output()

{

if (Size == "")

{

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

}

else

{

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

}

}

~Phone()

{

Producer = "";

Model = "";

Size = "";

Price = "";

}

};

public 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", string size = "22.2", string 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 != "")

{

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

}

}

public void Derived1Serialization(StreamWriter stream)

{

stream.WriteLine(Producer);

stream.WriteLine(Model);

stream.WriteLine(Size);

stream.WriteLine(Price);

stream.WriteLine(Country);

}

public void Derived1Deserialization(StreamReader stream)

{

Producer = stream.ReadLine();

Model = stream.ReadLine();

Size = stream.ReadLine();

Price = stream.ReadLine();

Country = stream.ReadLine();

}

~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", string size = "22.2", string 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 != "")

{

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", string size = "22.2", string 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 != "")

{

Console.WriteLine("Город: " + City);

}

}

~Derived3()

{

City = "";

}

};

internal class Program

{

static int Main(string[] args)

{

Phones List = new Phones();

Derived1 obj1 = new Derived1();

obj1.Input();

Derived1 obj2 = new Derived1();

obj2.Input();

List.AddToList(obj1);

List.AddToList(obj2);

Console.WriteLine("Коллекция List перед сереализацией\n");

List.PhonesListOutput();

List.PhonesListSerialization();

Phones List2 = new Phones(2);

List2.PhonesListDeserialization();

Console.WriteLine("Коллекция List2 после десереализации\n");

List2.PhonesListOutput();

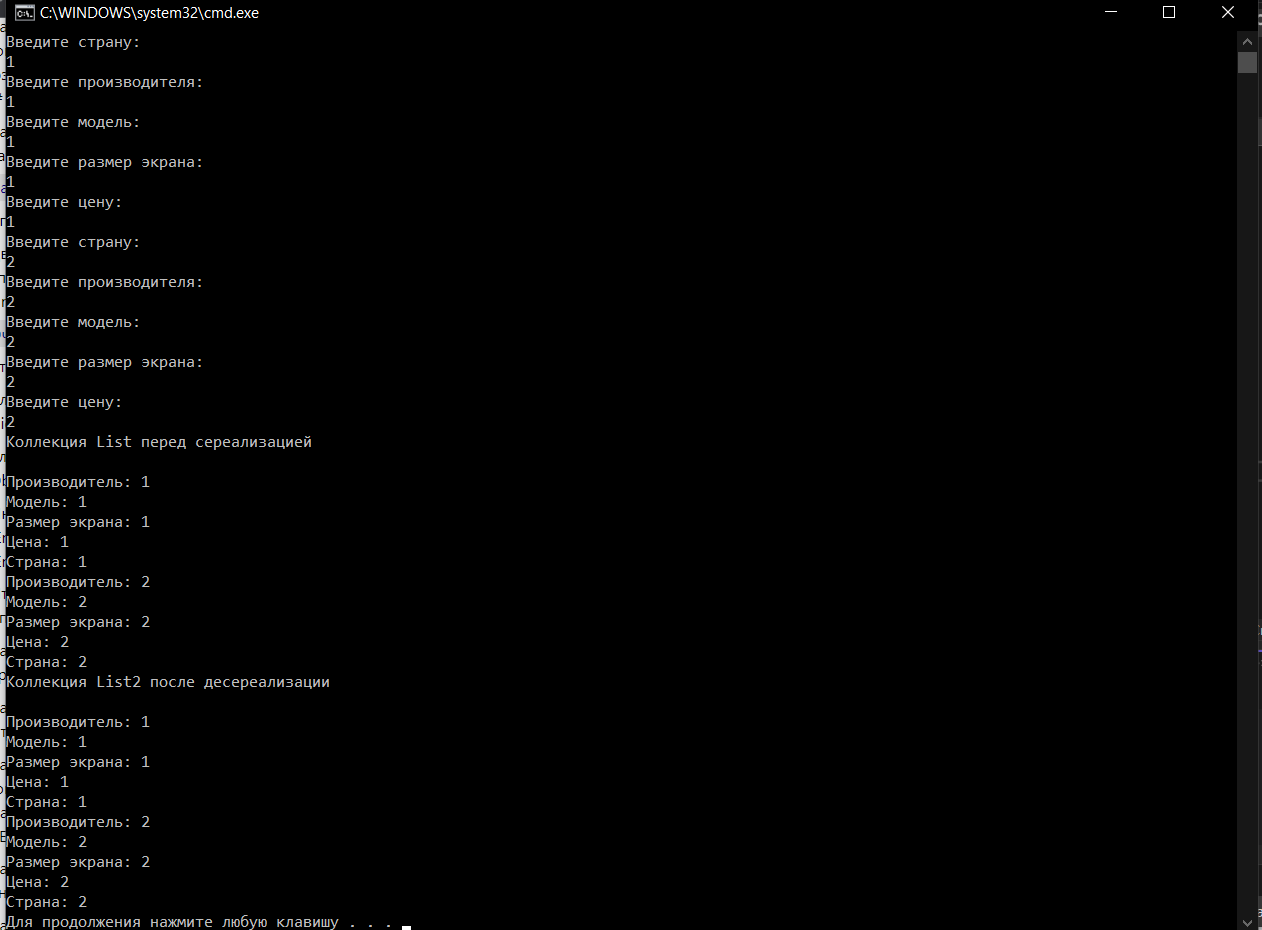
return 0;

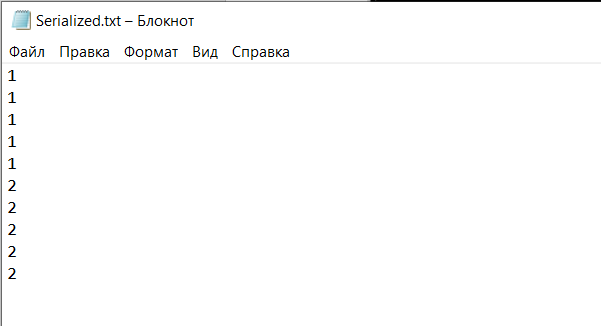
}

}

}

### Результаты работы программы





### Вывод:

Мы изучили стандартные коллекции и классов поддержку файловых потоков, входящих в состав Framework Class Library.