**Лабораторная работа №10**

**Тема:** Разработка программ, реализующих механизм наследования классов.

**Цель:** Научиться разработке программ, реализующих механизм наследования классов.

**Технологическое оснащение:** Ноутбук Lenovo Ideapad 320-15IKB, Windows 10, MS Word 2016, MS Visual Studio 2019, Visio 2016, методическое указание к работе.

**Выполнение работы**

**Вариант 29**

**Пример 1.** Разработаем класс для работы с одномерным массивом. Создадим в нём индексатор, позволяющий обращаться к элементу массива по индексу.

Выполнение примера 1 изображена на рисунке 1:

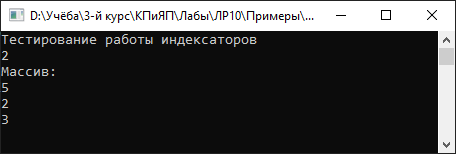


Рисунок 1 – Выполнение работы примера 1

**Пример 2.** Разработаем класс для работы с двумерным массивом (матрицей). Создадим в нём индексатор, позволяющий обращаться к элементу матрицы по индексу.

Выполнение примера 2 изображена на рисунке 2:

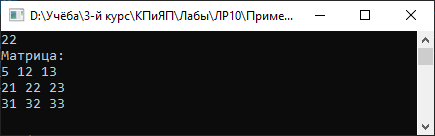


Рисунок 2 – Выполнение работы примера 2

**Пример 3.** Использование абстрактного класса.

Выполнение примера 3 изображена на рисунке 3:

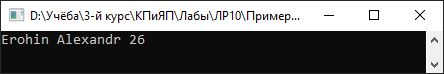


Рисунок 3 – Выполнение работы примера 3

**Пример 4.** Использование наследования.

Выполнение примера 4 изображена на рисунке 4:

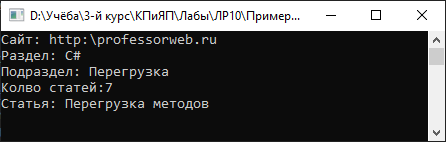


Рисунок 4 – Выполнение работы примера 4

**Пример 5.** Использование конструктора при наследовании.

Выполнение примера 5 изображена на рисунке 5:

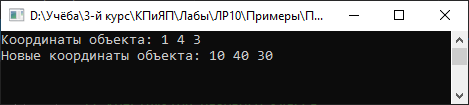


Рисунок 5 – Выполнение работы примера 5

**Пример 6.** Использование базового класса.

Выполнение примера 6 изображена на рисунке 6:

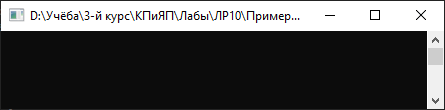


Рисунок 6 – Выполнение работы примера 6

**Пример 7.** Использование виртуальных методов, свойств и индексаторов.

Выполнение примера 7 изображена на рисунке 7:

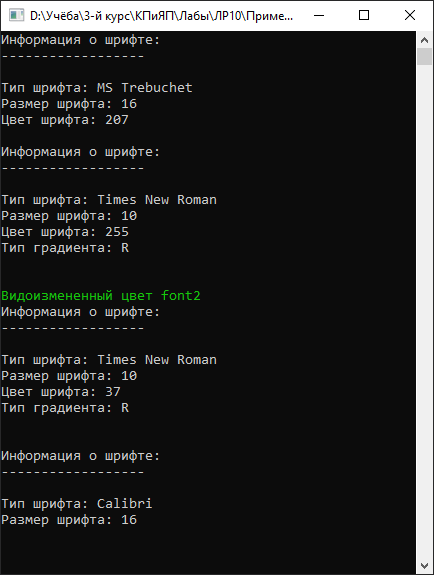


Рисунок 7 – Выполнение работы примера 7

**Задание.** Описать класс «множество», позволяющий выполнять основные операции: добавление и удаление элемента, пересечение, объединение и разность множеств.

Написать программу, демонстрирующую все разработанные элементы класса. Реализовать механизм наследования.

Блок-схема изображена на рисунке 8:

Рисунок 8 ­ Блок-схема

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace kipijp\_lr10\_v29

{

class MySet

{

public int[] elems;

public int size;

public int lastindex;

public MySet(int size)

{

this.lastindex = -1;

this.size = size;

this.elems = new int[size];

}

public bool isInSet(int elem)

{

bool isFind = false;

foreach (var i in elems)

{

if (elem == i) isFind = true;

}

return isFind;

}

public bool AddElem(int elem)

{

if(lastindex == size-1) return false;

if (isInSet(elem)) return false;

lastindex++;

elems[lastindex] = elem;

return true;

}

public bool RemElem(int elem)

{

if (!isInSet(elem)) return false;

int index = -1;

int[] arr = new int[size];

foreach (var i in elems)

{

if (i == elem) continue;

index++;

arr[index] = i;

}

elems = arr;

lastindex--;

return true;

}

public void PrintSet()

{

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

{

Console.Write("{0}\t", elems[i]);

}

Console.WriteLine();

}

}

class MySetAdvance : MySet

{

public MySetAdvance(int size, params int[] elms) : base(size)

{

foreach (var i in elms)

{

AddElem(i);

}

}

public MySet Cross(MySet set2)

{

MySet set3 = new MySet(size);

foreach (var i in elems)

{

if (set2.isInSet(i)) set3.AddElem(i);

}

return set3;

}

public MySet Diff(MySet set2)

{

MySet set3 = new MySet(size);

foreach (var i in elems)

{

if (!set2.isInSet(i)) set3.AddElem(i);

}

return set3;

}

public MySet Union(MySet set2)

{

MySet set3 = new MySet(size);

foreach (var i in elems)

{

set3.AddElem(i);

}

foreach (var i in set2.elems)

{

set3.AddElem(i);

}

return set3;

}

}

class Program

{

static void Main(string[] args)

{

int size = 10;

MySetAdvance set1 = new MySetAdvance(size, 1, 2, 3, 4, 5);

MySetAdvance set2 = new MySetAdvance(size, 3, 4, 5, 6, 7);

Console.WriteLine("Множестово 1:");

set1.PrintSet();

Console.WriteLine("Множестово 2:");

set2.PrintSet();

MySet crossSet = set1.Cross(set2);

MySet diffSet = set1.Diff(set2);

MySet diffSet2 = set2.Diff(set1);

MySet unionSet = set1.Union(set2);

Console.WriteLine("Пересечения множеств:");

crossSet.PrintSet();

Console.WriteLine("Разность множеств:");

diffSet.PrintSet();

diffSet2.PrintSet();

Console.WriteLine("Объеденение множеств:");

unionSet.PrintSet();

Console.WriteLine("Добавление элемента в множество:");

set1.AddElem(-1);

set1.PrintSet();

Console.WriteLine("Удаление элемента в множестве:");

set1.RemElem(-1);

set1.PrintSet();

Console.ReadKey(true);

}

}

}

Результат выполнения программы приведен на рисунке 9:

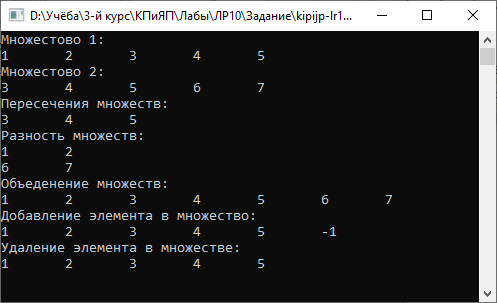


Рисунок 9 – Проверка работы программы

**Вывод:** в результате выполнения лабораторной работы были получены практические навыки по использованию механизма наследования в языке программирования C#.