Міністерство освіти та науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

ЗВІТ

до лабораторної роботи №1

з дисципліни «Основи Web-програмування»

Виконав: студент 2 курсу групи ІП-64 , ФІОТ

Хмельницький Арсеній

Номер залікової: 6428 Номер в списку: 27

Київ – 2017

**Код програми:**

using System;

using System.Collections.Generic;

using System.Text;

using System.Linq;

using System.Reflection;

using System.Runtime.Serialization.Formatters.Binary;

using System.IO;

using System.Runtime.Serialization;

namespace Pervaya\_popitka

{

public class Point

{

public Point() //конструктор по умолчанию

{

}

public Point(int x, int y) //конструктор

{

this.X = x;

this.Y = y;

}

private int \_X;

public int X

{

get //способ пролучения свойства

{

return this.\_X;

}

set //способ задания свойства

{

\_X = value;

}

}

private int \_Y;

public int Y

{

get

{

return this.\_Y;

}

set

{

\_Y = value;

}

}

public string PrintP()

{

String a = ("Point's x: " + this.X + " y: " + this.Y + " ");

return a;

}

}

public class ColoredPoint : Point

{

public ColoredPoint()

{ }

public ColoredPoint(int x, int y, String color)

: base(x, y)

{

this.Color = color;

}

private String \_Color;

public String Color

{

get

{

return this.\_Color;

}

set

{

\_Color = value;

}

}

public string PrintCP()

{

String a = this.PrintP() + "color: " + this.Color;

return a;

}

}

public class Line : Point

{

public Line()

{ }

public Line(Point start, Point end)

{

this.Start = start;

this.End = end;

}

public Line(int x1, int y1, int x2, int y2)

: this(new Point(x1, y1), new Point(x2, y2))

{ }

private Point \_Start;

public Point Start

{

get { return \_Start; }

set { \_Start = value; }

}

private Point \_End;

public Point End

{

get { return \_End; }

set { \_End = value; }

}

public string PrintL()

{

String a = ("Start " + this.Start.PrintP() + "End " + this.End.PrintP());

return a;

}

public string PrintL(Line temp)

{

String a = ("Start " + temp.Start.PrintP() + "End " + temp.End.PrintP());

return a;

}

}

public class ColoredLine : Line

{

public ColoredLine(Point start, Point end, String color)

: base(start, end)

{

this.Color = color;

}

public ColoredLine(int x1, int y1, int x2, int y2, String color)

: this(new Point(x1, y1), new Point(x2, y2), color)

{ }

private String \_Color;

public String Color

{

get

{

return this.\_Color;

}

set

{

\_Color = value;

}

}

public string PrintCL()

{

String a = this.PrintL() + "color: " + this.Color;

return a;

}

}

public class Polygon : Line

{

public int numOfLines;

public Polygon()

{ }

public Polygon(List<Line> frieng, int lines)

{

numOfLines = lines;

this.Frieng = frieng;

}

private List<Line> \_Frieng;

public List<Line> Frieng

{

get { return \_Frieng; }

set { \_Frieng = value; }

}

public String PrintPoly()

{

String a = "";

for (int i = 0; i < this.numOfLines; i++)

{

a += "Number of line: " + (i + 1) + ". " + this.PrintL(this.Frieng[i]) + "\n";

}

return a;

}

public override bool Equals(object obj)

{

if (!(obj is Polygon))

return false;

else

return ((Polygon)obj).Frieng == this.Frieng;

}

public Polygon DeepCopy()

{

Polygon obj = new Polygon(this.Frieng, numOfLines);

return obj;

}

public override int GetHashCode()

{

try

{

return Convert.ToInt32(this.Frieng);

}

catch (Exception e)

{

Console.WriteLine(e.Message);

return -1;

}

}

public static bool operator ==(Polygon one, Polygon two)

{

if ((Object)one == null || (Object)two == null)//проверить на null

return false;

return one.Equals(two);

}

public static bool operator !=(Polygon one, Polygon two)

{

if ((Object)one == null || (Object)two == null)//проверить на null

return true;

return !one.Equals(two);

}

}

public class Picture : List<Point>

// where T : class, IComparable<T>

{

private int size;

private int index = 0;

private static void Show\_Message(string message)

{

Console.WriteLine(message);

}

public Point[] array;

public Picture(int sizet)

{

size = sizet;

array = new Point[size];

//this.Added += Show\_Message;

//array = new Point[size];

}

public new Boolean Add(Point x)

{

array[index] = x;

index++;

return true;

}

public Boolean Add(Line x)

{

array[index] = x;

index++;

return true;

}

public int Length

{

get { return array.Length; }

}

public override string ToString()

{

string res = "";

for (var i = 0; i < Length; i++)

res = string.Concat(res, string.Concat(array[i] + "\n"));

return res;

}

}

class Program

{

static void Main(string[] args)

{

Picture p = new Picture(5);

Point First = new Point(1, 1); //создаем обьекты

ColoredPoint Second = new ColoredPoint(1, 2, "black");

Line Third = new Line(0, 0, 1, 1);

ColoredLine Fourth = new ColoredLine(0, 0, 1, 1, "black");

Polygon Triangle = new Polygon(new List<Line>() { new Line(0, 0, 0, 1), new Line(1, 0, 1, 1), new Line(0, 0, 1, 1) }, 3);

p.Add(First);

p.Add(Second);

p.Add(Third);

p.Add(Fourth);

p.Add(Triangle);

First.X = 3; //меняем некоторые параметры

Second.Color = "blue";

Third.End.Y = 2;

Fourth.Start.X = 3;

Triangle.Frieng[0].Start.X = 1;

p.array[0].X = 0;

Console.WriteLine(First.PrintP()); //выводим обьекты

Console.WriteLine(Second.PrintCP());

Console.WriteLine(Third.PrintL());

Console.WriteLine(Fourth.PrintCL());

Console.WriteLine(Triangle.PrintPoly());

Console.ReadKey();

}

}

class MyInvalidCastException : InvalidCastException

{

public MyInvalidCastException()

: base() { }

public MyInvalidCastException(string message)

: base(message) { }

public MyInvalidCastException(string format, params object[] args)

: base(string.Format(format, args)) { }

public MyInvalidCastException(string message, Exception innerException)

: base(message, innerException) { }

public MyInvalidCastException(string format, Exception innerException, params object[] args)

: base(string.Format(format, args), innerException) { }

}

}

}

}

**Скріншот програми:**

