**Завдання 1**

Додати в інтерфейс «IDemo» властивість Y, яка дозволить звертатися для читання до значення поля у. Реалізувати роботу з даною властивістю в класах «DemoPoint» і «DemoShape». Описати клас «Square» (геометрична фігура квадрат) та навести в цьому класі реалізацію інтерфейса «IMeasurable».

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab\_6

{

// визначення інтерфейсу

interface IDemo

{

void Show(); //оголошення методу

double Dlina(); //оголошення методу

int X { get; } //оголошення властивості, доступного тільки для читання

int Y { get; }

int this[int i] { get; set; } //оголошення індексатора, доступного для читання-запису

}

// клас DemoPoint успадковує інтерфейс IDemo

class DemoPoint : IDemo

{

protected int x;

protected int y;

public DemoPoint(int x, int y)

{

this.x = x;

this.y = y;

}

public void Show() //реалізація методу, оголошеного в інтерфейсі

{

Console.WriteLine("точка на плоскости: ({0}, {1})", x, y);

}

public double Dlina() //реалізація методу, оголошеного в інтерфейсі

{

return Math.Sqrt(x \* x + y \* y);

}

public int X //реалізація властивості, оголошеної в інтерфейсі

{

get

{

return x;

}

}

public int Y

{

get

{

return y;

}

}

public int this[int i] //реалізація індексатора, оголошеного в інтерфейсі

{

get

{

if (i == 0) return x;

else if (i == 1) return y;

else throw new Exception("неприпустимое значение индекса");

}

set

{

if (i == 0) x = value;

else if (i == 1) y = value;

else throw new Exception("неприпустимое значение индекса");

}

}

}

class DemoShape : DemoPoint, IDemo

{

protected int z;

public DemoShape(int x, int y, int z) : base(x, y)

{

this.z = z;

}

public new void Show()

{

Console.WriteLine("точка в пространстве: ({0}, {1}, {2})", x, y,

z);

}

public new double Dlina()

{

return Math.Sqrt(x \* x + y \* y + z \* z);

}

public new int this[int i]

{

get

{

if (i == 0) return x;

else if (i == 1) return y;

else if (i == 2) return z;

else throw new Exception("неприпустимое значение индекса");

}

set

{

if (i == 0) x = value;

else if (i == 1) y = value;

else if (i == 2) z = value;

else throw new Exception("неприпустимое значение индексв");

}

}

}

interface IMeasurable

{

double Perimetr();

double Plosha();

void Show();

}

class Square:IMeasurable

{

private double side;

public Square(double side)

{

{

this.side = side;

}

}

public void Show()

{

Console.WriteLine("Квадрат со стороной {0} ", side);

}

public double Perimetr()

{

return (4 \* side);

}

public double Plosha()

{

return (side \* side);

}

}

class Program

{

static void Main()

{

// створення масиву інтерфейсних посилань

IDemo[] a = new IDemo[4];

// заповнення масиву

a[0] = new DemoPoint(2, 1);

a[1] = new DemoPoint(-3, 0);

a[2] = new DemoShape(3, 4, 0);

a[3] = new DemoShape(0, 5, 6);

// перегляд масиву

foreach (IDemo x in a)

{

x.Show();

Console.WriteLine("Длинна={0:f2}", x.Dlina());

Console.WriteLine("x=" + x.X);

Console.WriteLine("y=" + x.Y);

x[1] += x[0];

Console.Write("новые координаты - ");

x.Show();

Console.WriteLine();

IMeasurable[] b = new IMeasurable[4];

b[0] = new Square(2);

b[1] = new Square(2.5);

b[2] = new Square(3);

b[3] = new Square(6);

foreach(IMeasurable y in b)

{

y.Show();

Console.WriteLine("Периметр квадрата = {0}, площадь = {1}", y.Perimetr(), y.Plosha());

}

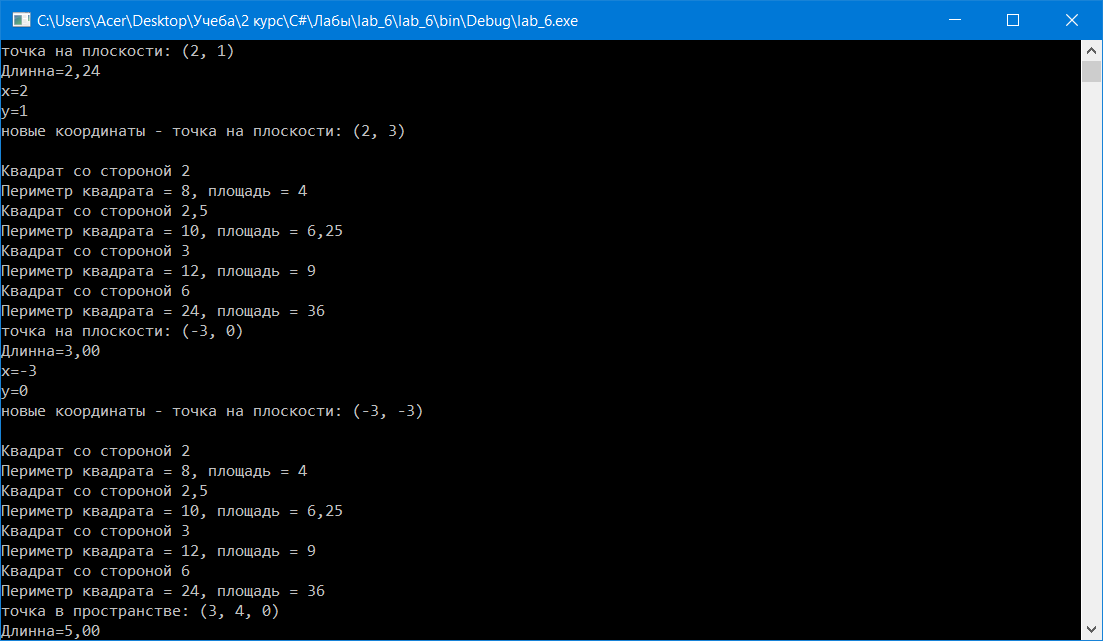
}

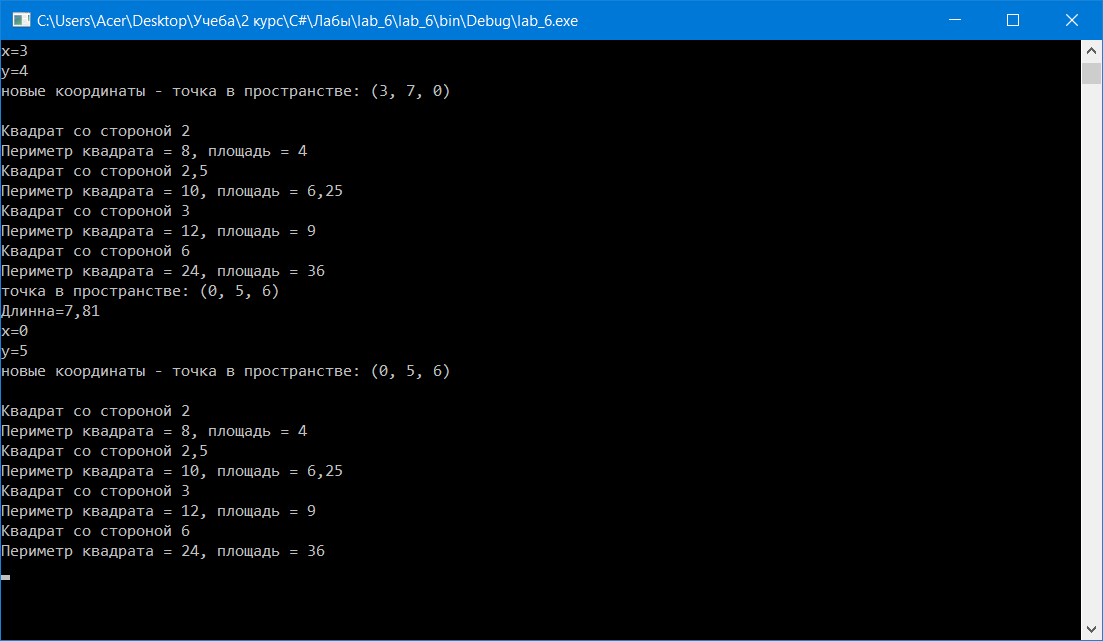
Console.ReadKey();

}

}

}





Завдання 2

Реалізувати підтримку інтерфейсу «System.IСomparable» у класі «Square» із завдання вище. Слід мати на увазі, що при реалізації методу «CompareTo» програміст самостійно визначає критерій порівняння об’єктів. Порівнювати квадрати за величиною периметру.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task2

{

interface IMeasurable

{

double Perimetr();

double Plosha();

void Show();

}

class Square : IMeasurable, IComparable<Square>

{

private double side;

public Square(double side)

{

this.side = side;

}

public void Show()

{

Console.WriteLine("Квадрат со стороной {0} (side);", side);

}

public double Perimetr()

{

return (4 \* side);

}

public double Plosha()

{

return (side \* side);

}

public int CompareTo(Square a)

{

return this.Perimetr().CompareTo(a.Perimetr());

}

}

class Program

{

static void Main(string[] args)

{

Square[] b = new Square[4];

b[0] = new Square(7);

b[1] = new Square(3);

b[2] = new Square(8);

b[3] = new Square(2);

foreach (Square x in b)

{

x.Show();

Console.WriteLine("Периметр квадрата = {0}, площадь = {1};", x.Perimetr(), x.Plosha());

Console.WriteLine();

}

Console.WriteLine("-----СОРТИРОВКА-----");

Console.WriteLine("Сортировка за значением периметра квадрата:");

Array.Sort(b);

foreach (Square x in b)

{

x.Show();

Console.WriteLine("Периметр квадрата = {0}, площадь = {1};", x.Perimetr(), x.Plosha());

Console.WriteLine();

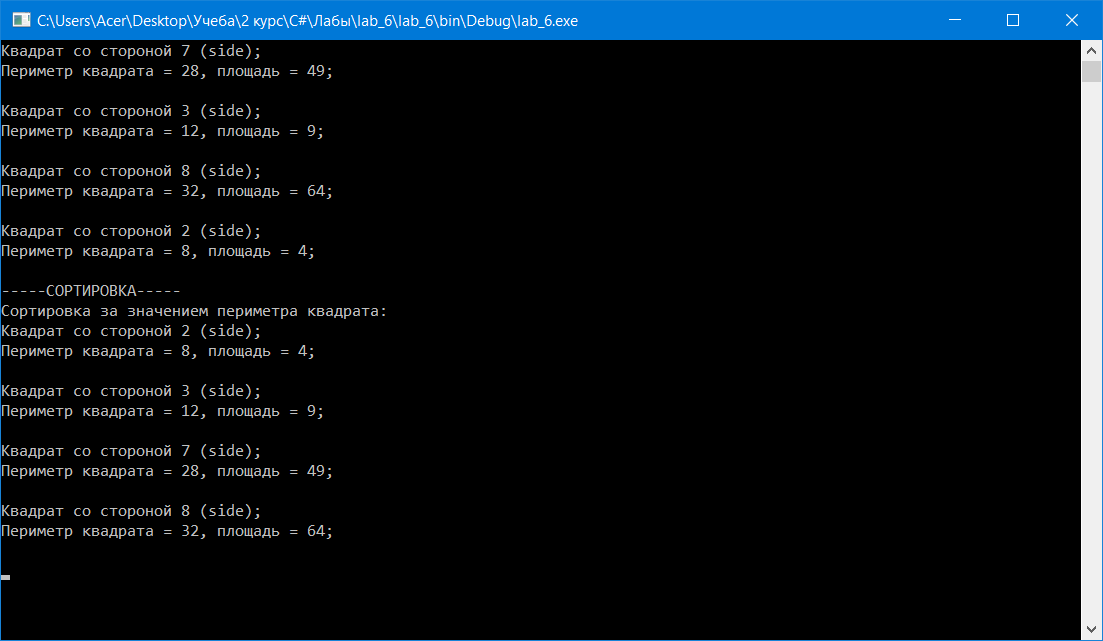
}

Console.ReadKey();

}

}

}



Завдання 3

Створити інтерфейс «Room». Реалізувати його в класах «ClassRoom», «MyRoom». Реалізувати інтерфейс «IComparable». В якості параметра для сортування використати площу кімнати. Реалізувати інтерфейс «IComparer», в якості порівняння використати поле «Кількість вікон».

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task2

{

interface IRoom:IComparable<IRoom>

{

double Area();

void Show();

int Windows { get; }

}

class RoomComparer : IComparer<IRoom>

{

public int Compare(IRoom a, IRoom b)

{

return a.Windows.CompareTo(b.Windows);

}

}

class MyRoom : IRoom

{

protected double width;

protected double length;

protected int windows;

public MyRoom(double x, double y, int w)

{

this.width = x;

this.length = y;

this.windows = w;

}

public double Area()

{

return (width \* length);

}

public int Windows

{

get { return windows; }

}

public void Show()

{

Console.WriteLine("Ширина = {0}, длина = {1}", width, length);

Console.WriteLine("Площадь комнаты = {0}", Area());

Console.WriteLine("Количество окон = {0}", windows);

}

public int CompareTo(IRoom a)

{

return this.Area().CompareTo(a.Area());

}

}

class ClassRoom : MyRoom, IRoom

{

private int seats;

public ClassRoom(double x, double y, int w, int s) : base (x, y, w)

{

this.seats = s;

}

public new double Area()

{

return (width \* length);

}

public new int Windows

{

get { return windows; }

}

public new void Show()

{

Console.WriteLine("Ширина = {0}, длина = {1}", width, length);

Console.WriteLine("Площадь комнаты = {0}", Area());

Console.WriteLine("Количество окон = {0}", windows);

Console.WriteLine("Количество мест для сидения = {0}", seats);

}

public new int CompareTo(IRoom a)

{

return this.Area().CompareTo(a.Area());

}

}

class Program

{

static void Main(string[] args)

{

IRoom[] array = new IRoom[4];

array[0] = new MyRoom(5, 7, 5);

array[1] = new MyRoom(2, 4, 1);

array[2] = new ClassRoom(6, 10, 4, 15);

array[3] = new ClassRoom(5, 8, 3, 12);

Console.WriteLine("Информация про комнаты:");

int number = 0;

foreach(IRoom x in array)

{

number++;

Console.WriteLine("Комната №{0}:", number);

x.Show();

Console.WriteLine();

}

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("Сортировка комнат по площади:");

Array.Sort(array);

number = 0;

foreach(IRoom x in array)

{

number++;

Console.WriteLine("Комната №{0}", number);

x.Show();

Console.WriteLine();

}

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("Сортировка комнат по количеству окон");

Array.Sort(array, new RoomComparer());

number = 0;

foreach (IRoom x in array)

{

number++;

Console.WriteLine("Комната №{0}", number);

x.Show();

Console.WriteLine();

}

Console.ReadKey();

}

}

}

