Лабораторная работа №11

Вариант 11

Выполнил: Олишкевич Игорь;

Тема: разработка программ, реализующих механизм наследования и интерфейсов;

Цель: сформировать понятия интерфейса, наследования, потомка класса и их реализации в С#;

Задание: построить иерархию классов в соответствии с вариантом задания: транспорт, экспресс, автомобиль, поезд. Расширить иерархию классов из с использованием виртуального класса в качестве основы иерархии. Показать пример использования полиморфизма методов.

**Задание 1:**

КАРТИНА: ФИО автора, название, жанр, список владельцев.

**Решение:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

interface Realese

{

void FIO(string F);

void Name(string N);

void Zanr(string Z);

void List(string L);

}

class Picture : Realese

{

public string fio;

public string name;

public string zanr;

public string list;

public Picture(string fio, string name, string zanr, string list)

{

this.fio = fio;

this.name = name;

this.zanr = zanr;

this.list = list;

}

public void FIO(string F)

{

Console.WriteLine(F);

}

public void Name(string N)

{

Console.WriteLine(N);

}

public void Zanr(string Z)

{

Console.WriteLine(Z);

}

public void List(string L)

{

Console.WriteLine(L);

}

}

static void Main(string[] args)

{

Picture[] pic = new Picture[2];

for (int i = 0; i < 2; i++)

{

string fio = Console.ReadLine();

string name = Console.ReadLine();

string zanr = Console.ReadLine();

string list = Console.ReadLine();

pic[i] = new Picture(fio, name, zanr, list);

pic[i].FIO(pic[i].fio);

pic[i].Name(pic[i].name);

pic[i].Zanr(pic[i].zanr);

pic[i].List(pic[i].list);

}

}

}

}

**Задание 2:**

1. Добавьте в интерфейс IDemo свойство Y, которое позволит обращаться для чтения к значению поля у. Реализуйте работу с данным свойством в классах DemoPoint и DemoShape
2. Добавьте свойство Z для обращения к полю z класса DemoShape. Подумайте, куда именно нужно добавить определение данного свойства и почему.

**Решение:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp24

{

//определение интерфейса

interface IDemo

{

void Show(); //объявление метода

double Dlina(); //объявление метода

int X { get; } //объявление свойства, доступного только для чтения

int Y { get; }

int Z { get; }

int this[int i] { get; set; } //объявление индексатора, доступного для чтения-записи

}

//класс DemoPoint наследует интерфейс IDemo

class DemoPoint : IDemo

{

protected int x;

protected int y;

protected int z = 0;

public int Z

{

get

{

return z;

}

}

public DemoPoint(int x, int y)

{

this.x = x; this.y = y;

}

public void Show() //реализация метода, объявленного в интерфейсе

{

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

}

public double Dlina() //реализация метода, объявленного в интерфейсе

{

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

}

public int X //реализация свойства, объявленного в интерфейсе

{

get

{

return x;

}

}

public int Y

{

get

{

return y;

}

}

public int this[int i] //реализация индексатора, объявленного в интерфейсе

{

get

{

if (i == 0) return x;

else if (i == 1) return y;

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

}

set

{

if (i == 0) x = value;

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

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

}

}

}

//класс DemoShape наследует класс DemoPoint и интерфейс IDemo

class DemoShape : DemoPoint, IDemo

{

protected int z;

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

{

this.z = z;

}

// реализация метода, объявленного в интерфейсе, с сокрытием одноименного метода из

//базового класса

public new void Show()

{

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

z);

}

// реализация метода, объявленного в интерфейсе, с сокрытием одноименного метода из

//базового класса

public new double Dlina()

{

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

}

// реализация индексатора, объявленного в интерфейсе, с сокрытием одноименного

// индексатора из базового класса

public int Z

{

get

{

return 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("недопустимое значение индекса");

}

set

{

if (i == 0) x = value;

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

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

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

}

}

}

class Program

{

static void Main(string[] args)

{

//создание массива интерфейсных ссылок

IDemo[] a = new IDemo[4];

//заполнение массива

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

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

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

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

//просмотр массива

foreach (IDemo x in a)

{

x.Show();

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

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

x[1] += x[0];

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

Console.WriteLine("z=" + x.Z);

Console.Write("новые координаты - ");

x.Show();

Console.WriteLine();

}

Console.ReadKey();

}

}

}

**Задание 3:**

Измените метод CompareTo так, чтобы метод Sort сортировал массив точек по убыванию расстояния между точкой и началом координат.

**Решение:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp25

{

interface IComparable

{

int CompareTo(object T);

}

class DemoPoint : IComparable<DemoPoint>

{

protected int x;

protected int y;

public DemoPoint(int x, int y)

{

this.x = x; this.y = y;

}

public void Show()

{

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

}

public double Dlina()

{

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

}

//реализация метода CompareTo

public int CompareTo(DemoPoint obj)

{

//определяем критерии сравнения текущего объекта с параметром в

// зависимости от удаленности точки от начала координат

if (this.Dlina() == obj.Dlina()) return 0;

else if (this.Dlina() < obj.Dlina()) return 1;

else return -1;

}

}

class Program

{

static void Main(string[] args)

{

DemoPoint[] a = new DemoPoint[4];

a[0] = new DemoPoint(5, -1);

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

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

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

Console.WriteLine("Не отсортированный!");

Console.WriteLine();

foreach (DemoPoint x in a)

{

x.Show();

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

}

Array.Sort(a);

Console.WriteLine();

Console.WriteLine("Отсортированный по убыванию");

Console.WriteLine();

foreach (DemoPoint x in a)

{

x.Show();

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

}

Console.ReadLine();

}

}

}

Тестирование:

