**《C#程序设计》课程实验报告**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **学号：** | **12021051064** |  | **姓名：** | **华心童** |

**实验六、C#的类和对象**

**一、实验目的**

1、定义类。

2、访问类的成员及方法，析构函数，输出型参数的使用。

3、将实验报告云盘，文件名是**学号姓名实验6**命名。

**二、实验内容**

实验准备：在F盘创建学号文件夹，用于存放实验结果文件。

运行VS2013，创建解决方案是“**Ex学号-6**”。

1. **基本实验**

**任务1、创建类，输出类的成员【项目名P601】**

**要求：**按类图完成Point类的定义。

在Main中创建两个点对象，使用构造函数初始化两点的坐标，计算这两点间的距离并输出。同时观察构造函数和析构函数的自动调用过程。

|  |
| --- |
| Point |
| - x ：int //横坐标值  - y ：int //纵坐标值  +PX ：int //公有属性，用于读写字段x  +PY ：int //公有属性，用于读写字段y |
| +Point(); //无参构造，初始化点为原点（0,0）  +Point(int x,int y); //用形参x和y初始化点的坐标  +~Point(); //析构函数，函数体内输出提示点的坐标  +ShowPoint() ：void //显示点的坐标 |

**提示：在主函数中完成：坐标距离计算：d=√[(x2-x1)²+(y2-y1)²]，并理解主函数中如何读取坐标值（私有成员），并将距离并显示。**

**程序源程序代码：**粘贴program.cs中代码

using System;

namespace P601

{

class Point

{

private readonly int x;

private int X

{

get { return x; }

}

private readonly int y;

private int Y

{

get { return y; }

}

public int PX;

public int PY;

public Point()

{

this.x = x;

this.y = y;

}

public Point(int x, int y)

{

this.PX = x;

this.PY = y;

}

public void ShowPoint()

{

Console.WriteLine("该点的坐标是：({0},{1})",PX,PY);

}

~Point()

{

Console.WriteLine("点({0},{1})被析构", PX, PY);

}

class Program

{

static void Main(string[] args)

{

Point P1 = new Point();

Console.Write("请输入第一个坐标：");

P1.PX = int.Parse(Console.ReadLine());

P1.PY = int.Parse(Console.ReadLine());

P1.ShowPoint();

Point P2 = new Point();

Console.Write("请输入第二个坐标：");

P2.PX = int.Parse(Console.ReadLine());

P2.PY= int.Parse(Console.ReadLine());

P2.ShowPoint();

double d = Math.Sqrt(Math.Pow(P2.PX - P1.PX, 2)+Math.Pow(P2.PY-P1.PY,2));

Console.WriteLine("两点之间的距离为：{0:F2}",d);

Console.ReadKey();

}

}

}

}

//粘贴运行结果界面截图



**任务2、创建类，理解成员输出型参数【项目名P602】**

设计一个学生类如下，按要求补充该类方法的定义，并为类增加索引函数用于访问成绩数组元素，最后设计Main测试该类各个方法以及索引函数的调用。

class Student

{ private string stuNo,stuName;

private int[] score ;

public Student()

{

stuNo = "";

stuName = "";

score = new int[5];

}

public Student(string stuNo,string stuName)

{

this.stuNo = stuNo;

this.stuName = stuName;

this.score = new int[5]{0,0,0,0,0};

}

public void Display(){

//显示学生所有信息

}

public void SetScore(params int[] score){

//为所有成绩赋值

}

public void GetMaxAndMin(out int max,out int min){

//查找最高及最低分数

}

public float GetAvg(out int count){

//返回该生的平均分数，并统计不及格门数

}

}

**源程序代码：**粘贴program.cs中代码

using System;

namespace P602

{

class Student

{

private string stuNo, stuName;

private int[] score;

public int[] Score

{

get { return score; }

set { score = value; }

}

public Student()

{

stuNo = "64";

stuName = "Jason";

score = new int[5];

}

public Student(string stuNo, string stuName)

{

this.stuNo = stuNo;

this.stuName = stuName;

this.score = new int[5] { 0, 0, 0, 0, 0 };

}

public int this[int index]

{

get { return score[index]; }

set { score[index] = value; }

}

public void Display()

{

Console.Write("学号：{0} ", stuNo);

Console.WriteLine("姓名：{0}", stuName);

Console.Write("五门科目分数分别为：");

for(int i = 0; i < score.Length; i++)

{

Console.Write(this.score[i]);

Console.Write(' ');

}

GetMaxAndMin(out int max, out int min);

Console.Write("平均分：{0}", GetAvg(out int count));

//显示学生所有信息

}

public void SetScore(params int[] score)

{

for (int i = 0; i < score.Length; i++)

this.score[i] = score[i];

//为所有成绩赋值

}

public void GetMaxAndMin(out int max, out int min)

{

max = 0; min = 100;

for (int i = 0; i < score.Length; i++)

{

if (score[i] > max) max = score[i];

if (score[i] < min) min = score[i];

}

Console.WriteLine("最高分：{0}，最低分：{1}", max, min);

//查找最高及最低分数

}

public float GetAvg(out int count)

{

float avg = 0;

int sum = 0;

count = 0;

for (int i = 0; i < score.Length; i++)

{

if (score[i] < 60) count++;

sum += score[i];

}

avg = sum / score.Length;

Console.Write("不及格门数：{0} ", count);

return avg;

//返回该生的平均分数，并统计不及格门数

}

}

class Program

{

static void Main(string[] args)

{

int count = 0;

Student s1 = new Student();

Console.Write("请输入五门课程的成绩：");

string value = Console.ReadLine();

string[] val = value.Split(' ');

int[] score = new int[val.Length];

for (int i = 0; i < val.Length; i++)

score[i] = int.Parse(val[i]);

s1.SetScore(score);

s1.Display();

Console.ReadKey();

}

}

}

//粘贴运行结果界面截图

