**《C#程序设计》课程实验报告**

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

**实验十二、接口与集合**

**一、实验目的**

1、了解接口的定义与使用。

2、了解集合接口与实现。

3、将实验报告云盘，文件名是**学号姓名实验12**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-12**”。

1. **基本实验**

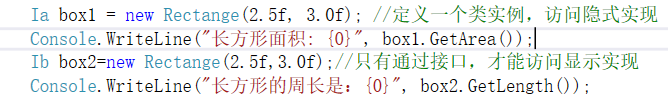
**任务1、接口的定义与使用，【项目名P1201】**

**要求：根据以下描述，完成接口的定义与实现**。

1. 接口Ia，成员GetArea
2. 接口Ib，成员GetLength
3. 类Rectangle，继承自Ia，Ib

|  |
| --- |
| Rectangle |
| - x：float //长  - y：float //宽 |
| + Rectangle(float x, float y) //带参构造  + GetArea：float　　　　//隐式接口成员实现  GetLength：float　　　//显式接口成员实现 |

主代码部分样例



**程序源程序代码：**粘贴program.cs中代码

interface Ia

{

float GetArea();

}

interface Ib

{

float GetLength();

}

class Rectangle : Ia, Ib

{

private float x;

private float y;

public Rectangle(float x, float y)

{

this.x = x;

this.y = y;

}

public float GetArea()

{

return x \* y;

}

float Ib.GetLength()

{

return 2 \* (x + y);

}

}

class Program

{

static void Main(string[] args)

{

Rectangle r1 = new Rectangle(2.5f, 3.0f);

Console.WriteLine("长方形的面积为：" + r1.GetArea());

Ia box1 = r1;

Console.WriteLine("长方形的面积为：" + box1.GetArea());

Ib box2 = new Rectangle(2.5f, 3.0f);

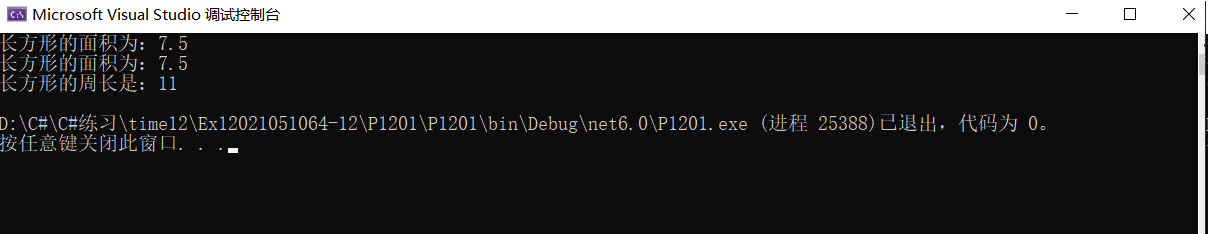
Console.WriteLine("长方形的周长是：" + box2.GetLength());

Console.ReadKey();

}

}

//粘贴运行结果界面截图



**任务2、ArrayList集合类【项目名P1202】**

要求：使用ArrayList集合类对姓名进行排序。

了解ArrayList的Add，Insert，AddRange的使用。

**程序源程序代码：**粘贴program.cs中代码

class Program

{

static void Main(string[] args)

{

ArrayList a1 = new ArrayList();

a1.Add("华心童");

//a1.Add("李宽");

//a1.Add("王嘉俊");

a1.Insert(0, "Messi");

//a1.Add("焦心印");

a1.Add("马磊");

a1.Add("孙钰程");

a1.Insert(4, "Jason");

a1.AddRange(new ArrayList() { "李四", "张三" });

Console.Write("排序前：");

foreach (object obj in a1)

{

Console.WriteLine(obj);

}

a1.Sort();

Console.Write("排序后：");

foreach (object obj in a1)

{

Console.WriteLine(obj);

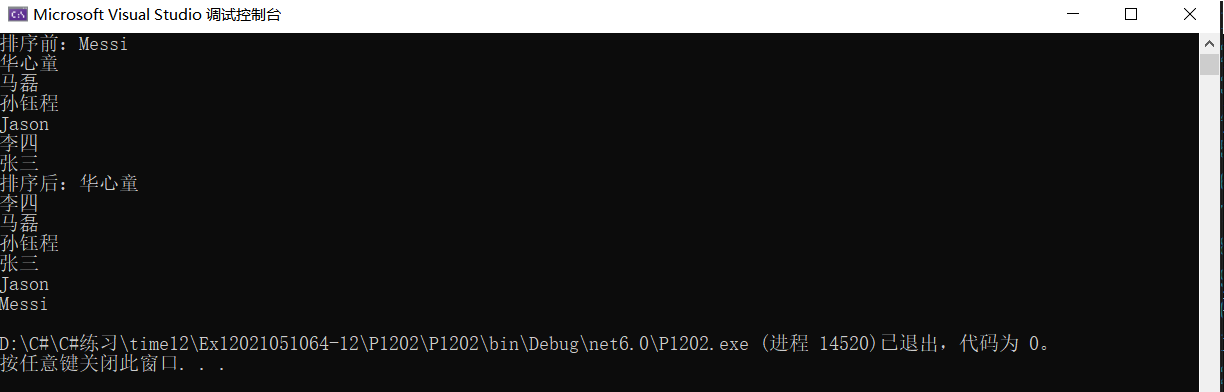
}

Console.ReadKey();

}

}

//粘贴运行结果界面截图



**任务3、List集合类【项目名P1203】**

要求：使用List集合类对水果进行排序。

了解List的Add，Insert，AddRange的使用。

**源程序代码：**粘贴program.cs中代码

class Program

{

static void Main(string[] args)

{

List<string> a1 = new List<string>();

a1.Add("apple");

a1.Add("orange");

a1.Insert(0, "cherry");

a1.Add("watermelon");

a1.Add("pear");

a1.Insert(2, "banana");

a1.AddRange(new List<string>() { "pineapple", "grape" });

Console.Write("排序前：");

foreach (object obj in a1)

{

Console.WriteLine(obj);

}

a1.Sort();

Console.Write("排序后：");

foreach (object obj in a1)

{

Console.WriteLine(obj);

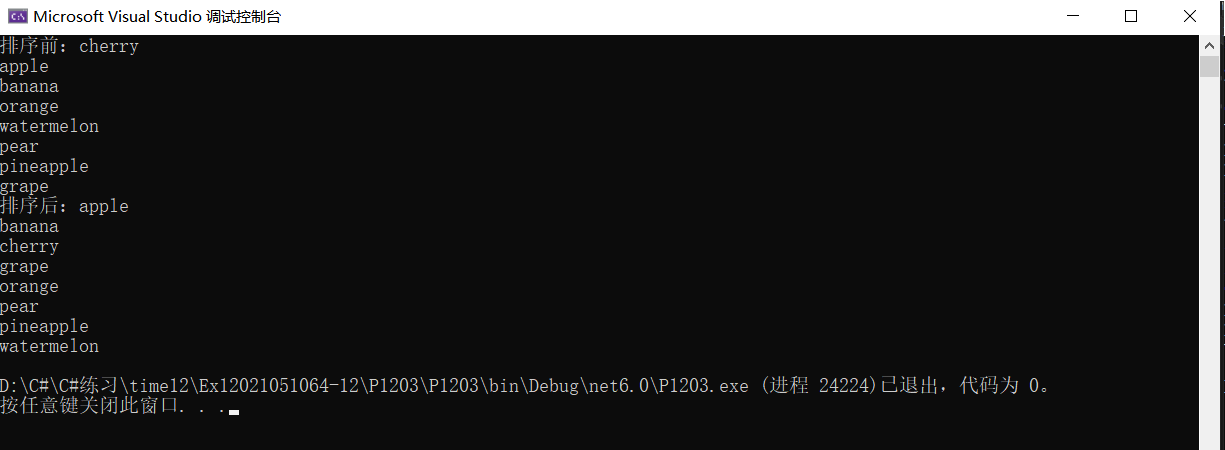
}

Console.ReadKey();

}

}

//粘贴运行结果界面截图



**任务4、ArrayList集合类【项目名P1204】**

要求：Student从IComparable接口派生，实现接口方法CompareTo，对数据按分数、学号和姓名的排序输出。

|  |
| --- |
| Student |
| - no：int;  - name：string;  - score：float;  +No：int、+Name：string，+Score：float//实现读属性 |
| + Student(int no, string name, float score) //带参构造  + Display()：void　　　　//显示学生信息  + CompareTo(object obj)：int//实现隐式接口方法 |

static void DispStu(ArrayList myarr, string str)//静态方法，显示信息

**main函数**



**源程序代码：**粘贴program.cs中代码

class Student : IComparable

{

private int no;

public int No

{

get { return no; }

set { no = value; }

}

private string name;

public string Name

{

get { return name; }

set { name = value; }

}

private float score;

public float Score

{

get { return score; }

set { score = value; }

}

public Student(int no, string name, float score)

{

this.no = no;

this.name = name;

this.score = score;

}

public void Display()

{

Console.WriteLine("\t{0}\t{1}\t{2}", No, Name, Score);

}

public int CompareTo(object obj)

{

Student s1 = (Student)obj;

//只排分数

//if (this.score > s1.score) return -1;

//else if (this.score == s1.score) return 0;

//else return 1;

if (this.score != s1.score)

return s1.score.CompareTo(this.score);

//排的分数和姓名

else if (this.no != s1.no && !this.name.Equals(s1.name))

return this.name.CompareTo(s1.name);

//排的分数和学号

//else if (this.no != s1.no && !this.name.Equals(s1.name))

// return this.no.CompareTo(s1.no);

return 0;

}

}

class Program

{

static void DispStu(ArrayList myarr, string str)

{

Console.WriteLine(str);

Console.WriteLine("————————————————");

Console.WriteLine("\t学号\t姓名\t分数");

foreach (Student s in myarr)

{

s.Display();

}

}

static void Main(string[] args)

{

int n = 4;

ArrayList myarr = new ArrayList();

Student[] ST = new Student[4] { new Student(1, "Daniel", 85), new Student(5, "Keven", 78), new Student(2, "John", 100), new Student(8, "Jason", 100) };

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

{

myarr.Add(ST[i]);

}

DispStu(myarr, "排序前：");

myarr.Sort();

DispStu(myarr, "按分数排序后：");

Console.ReadKey();

}

}

//粘贴运行结果界面截图

