**实验二 面向对象程序设计**

**一、实验目的**  
1. 理解类的定义、继承等面向对象的的基本概念；  
2. 掌握 C#语言定义类及其各种成员（字段，属性，方法） 的方法；  
3. 掌握方法覆盖的应用；  
4. 掌握接口的定义和实现方法。

**二、 实验要求**

根据要求，编写 C#程序，并将程序代码和运行结果写入实验报告。

**三、 实验内容**  
1. 设计编写一个控制台应用程序， 练习类的继承。  
 (1) 编写一个抽象类 People，具有”姓名”， ”年龄”字段， ”姓名”属性， Work 方法。  
 (2) 由抽象类 People 派生出学生类 Student 和职工类 Employer， 继承 People 类，并  
 覆盖 Work 方法。  
 (3) 派生类 Student 增加”学校”字段，派生类 Employer 增加”工作单位”字段。  
 (4) 在 Student 和 Employer 实例中输出各自不同的信息。  
2. 编写一个控制台应用程序，输入正方形边长或者半径，计算其周长和面积并输出。  
 (1) 编写两个接口，接口 IShape 包含三个方法： initialize, getPerimeter, getArea。 分别进行初始化、获取边长和面积， 其返回值均为 decimal。 接口 IDisplayresult 显示计算结果。

1. **设计编写一个控制台应用程序， 练习类的继承。**

**源程序：**

**主函数：**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace renwu1

{

class Program

{

static void Main(string[] args)

{

Student stu = new Student("孟祥通",22,"石家庄铁道大学");

stu.Work();

Employer em = new Employer("羊城新",21,"阿里粑粑");

em.Work();

Console.ReadLine();

}

}

}

**People.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace renwu1

{

abstract class People

{

public string name;

public int age;

public People()

{

}

public People(string name,int age)

{

this.name = name;

this.age = age;

}

public void Work()

{

Console.WriteLine("姓名：{0} 年龄：{1}",name,age);

}

}

}

**Student.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace renwu1

{

class Student:People

{

public string school;

public Student(string name, int age, string school)

{

this.name = name;

this.age = age;

this.school = school;

}

public new void Work()

{

Console.WriteLine("姓名：{0} 年龄：{1} 学校：{2}",name,age,school);

}

}

}

**Employer.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace renwu1

{

class Employer:People

{

public string workaddress;

public Employer(string name,int age,string workaddress)

{

this.name = name;

this.age = age;

this.workaddress = workaddress;

}

public new void Work()

{

Console.WriteLine("姓名：{0} 年龄：{1} 工作单位：{2}", name, age, workaddress);

}

}

}

**程序结果截图：**

****

1. **编写一个控制台应用程序，输入正方形边长或者半径，计算其周长和面积并输出。  
   (1) 编写两个接口，接口 IShape 包含三个方法： initialize, getPerimeter, getArea。 分  
   别进行初始化、获取边长和面积， 其返回值均为 decimal。 接口 IDisplayresult 显示计算结  
   果。  
   (2) 编写两个类， Square（正方形） 和 Circle（圆形） ，实现 IShape 和 IDisplayresult  
   接口。**

**源程序：**

**主函数：**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace renwu2

{

class Program

{

static void Main(string[] args)

{

Circle c = new Circle();

c.initialize(3);

c.Display();

Square s = new Square();

s.initialize(4);

s.Display();

Console.ReadLine();

}

}

}

**Ishape.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace renwu2

{

public interface IShape

{

decimal initialize(double l);

decimal getPerimeter();

decimal getArea();

}

}

**Idisplayresult.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace renwu2

{

public interface IDisplayresult

{

void Display();

}

}

**Circle.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace renwu2

{

class Circle : IShape, IDisplayresult

{

public double radius;

public decimal initialize(double radius)

{

this.radius = radius;

return (decimal)(this.radius);

}

public decimal getPerimeter()

{

return (decimal)(2 \* 3.14 \* radius);

}

public decimal getArea()

{

return (decimal)(3.14 \* radius \* radius);

}

public void Display()

{

Console.WriteLine("周长：{0} 面积：{1}", getPerimeter(), getArea());

}

}

}

**Square.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace renwu2

{

class Square:IShape,IDisplayresult

{

public double len;

public decimal initialize(double len)

{

this.len = len;

return (decimal)(this.len);

}

public decimal getPerimeter()

{

return (decimal)(4 \* len);

}

public decimal getArea()

{

return (decimal)(len \* len);

}

public void Display()

{

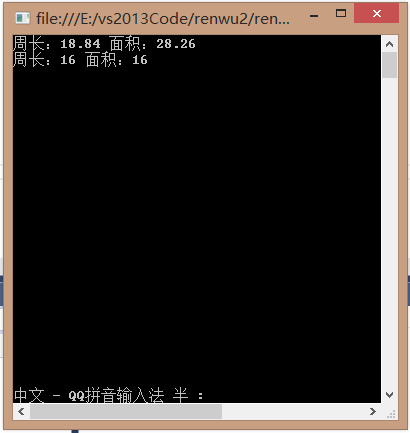
Console.WriteLine("周长：{0} 面积：{1}",getPerimeter(),getArea());

}

}

}

**程序结果截图：**

****

**四、实验总结：**

通过此次实验我了解了在c#中抽象和接口的使用，与c++相较而言，我感觉是区别不大，虽然在有些地方也有差异，但是机制还是一样的，不同的只不过是语言的语法差异罢了。