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。

源代码：

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace test2.\_1

{

class Employer : People

{

String Place;

public Employer(String name, String age, String place)

{

this.Name = name;

this.Age = age;

this.Place = place;

}

public Employer() { }

public new void work()

{

Console.WriteLine("职工的名字{0}，职工的年龄{1}，职工的地址{2}", Name, Age, Place );

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace test2.\_1

{

abstract class People

{

protected String Name;

protected String Age;

public void work(){}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace test2.\_1

{

class Program

{

static void Main(string[] args)

{

Student student = new Student("小阿布", "8", "石家庄铁道大学");

student.work();

Employer employer = new Employer("阿布","18","石家庄北二环");

employer.work();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace test2.\_1

{

class Student:People

{

String School;

public Student(String name, String age, String school)

{

this.Name = name;

this.Age = age;

this.School = school;

}

public Student() { }

public new void work()

{

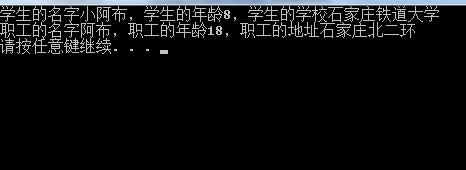
Console.WriteLine("学生的名字{0}，学生的年龄{1}，学生的学校{2}",Name,Age,School);

}

}

}

程序结果截图：



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

(2) 编写两个类，Square（正方形）和 Circle（圆形），实现 IShape 和 IDisplayresult 接口。

源代码：

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace test2.\_2

{

public interface IShape

{

double Gramlength();

double Area();

int Sides{get;}

void displayresult();

}

public interface Idisplayresult

{

void displayresult();

}

public class Square : IShape, Idisplayresult

{

private int sides;

public int sidelength;

public Square()

{

sides = 4;

}

public int Sides

{

get { return sides; }

}

public double Area()

{ return ((double)(sidelength \* sidelength)); }

public double Gramlength()

{ return ((double)(Sides \* sidelength)); }

void IShape.displayresult()

{

Console.WriteLine("\n计算正多边形的周长面积：");

Console.WriteLine("正{0}边形", this.Sides);

Console.WriteLine("边长是：{0}", this.sidelength);

Console.WriteLine("周长是：{0}", this.Gramlength());

Console.WriteLine("面积是：{0}", this.Area());

}

void Idisplayresult.displayresult()

{

Console.WriteLine("显示图形");

}

}

public class Circle : IShape, Idisplayresult

{

private int sides;

public int sidelength;

public Circle()

{

sides = 0;

}

public int Sides

{

get { return sides; }

}

public double Area()

{ return ((double)(sidelength \* sidelength\*3.14)); }

public double Gramlength()

{ return ((double)(2 \* sidelength\*3.14)); }

void IShape.displayresult()

{

Console.WriteLine("\n计算圆的周长面积：");

Console.WriteLine("正{0}边形", this.Sides);

Console.WriteLine("边长是：{0}", this.sidelength);

Console.WriteLine("周长是：{0}", this.Gramlength());

Console.WriteLine("面积是：{0}", this.Area());

}

void Idisplayresult.displayresult()

{

Console.WriteLine("显示图形");

}

}

class Program

{

static void Main(string[] args)

{

Square tt = new Square();

tt.sidelength = 6;

IShape sh = ( IShape)tt;

sh.displayresult();

Idisplayresult dis1 = (Idisplayresult)tt;

dis1.displayresult();

Circle ll = new Circle();

ll.sidelength = 6;

IShape al = (IShape)ll;

al.displayresult();

Idisplayresult di = (Idisplayresult)ll;

di.displayresult();

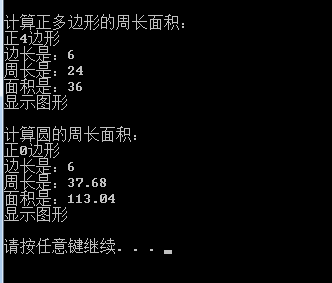
Console.ReadLine();

}

}

}

程序结果截图：



**四、实验总结**

通过本次试验，我练习类的继承，以及类的接口和类与类之间的联系，认识到自己在这一方面还有一些不足，需要更加努力的学习这方面的知识。