**《C#程序设计》课程实验报告**

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

**实验十一、抽象类与密封类**

**一、实验目的**

1、了解抽象类和抽象方法。

2、了解密封类和密封方法。

3、将实验报告云盘，文件名是**学号姓名实验11**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-11**”。

1. **基本实验**

**任务1、抽象类与抽象方法，【项目名P1101】**

**要求：根据以下类，完成类的设计**。

|  |
| --- |
| 抽象类Animal |
| - name：string //名字  - age：int //年龄  定义name和age的读写属性 |
| + Animal() { } //无参构造  + Animal(String name, int age)//带参构造  + Born()：void //显示姓名和年龄  + Speak(string vioce)：void //虚拟方法，输出叫声  + Eat(string food)：void; //抽象方法，吃的食物 |

Cat类 // Animal派生类，重写构造及方法，并实例化，测试

Dog类 // Animal派生类，重写构造及方法，并实例化，测试

Cow类 // Animal派生的抽象类，重写抽象方法，无需实例化。

**程序源程序代码：**粘贴program.cs中代码

abstract class Animal

{

private string name;

public string Name

{

get { return name; }

set { name = value; }

}

private int age;

public int Age

{

get { return age; }

set { age = value; }

}

public Animal() { }

public Animal(string name, int age)

{

this.name = name;

this.age = age;

}

public void Born()

{

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

}

public virtual void Speak(string voice)

{

Console.WriteLine("发出了" + voice + "的声音");

}

public abstract void Eat(string food);

}

class Cat : Animal

{

public Cat() { }

public Cat(string name, int age) : base(name, age)

{

}

public new void Born()

{

base.Born();

}

public override void Speak(string voice)

{

base.Speak(voice);

}

public override void Eat(string food)

{

Console.WriteLine("{0}爱吃{1}", Name, food);

}

}

class Dog : Animal

{

public Dog() { }

public Dog(string name, int age) : base(name, age)

{

}

public new void Born()

{

base.Born();

}

public override void Speak(string voice)

{

base.Speak(voice);

}

public override void Eat(string food)

{

Console.WriteLine("{0}爱吃{1}", Name, food);

}

}

abstract class Cow : Animal

{

public Cow() { }

public Cow(string name, int age) : base(name, age)

{

}

public new void Born()

{

base.Born();

}

public override void Speak(string voice)

{

base.Speak(voice);

}

public override abstract void Eat(string food);

}

class Program

{

static void Main(string[] args)

{

Animal a1 = new Cat() { Name = "招财", Age = 4 };

a1.Born();

a1.Speak("喵喵...");

a1.Eat("鱼");

Animal a2 = new Dog() { Name = "卡卡", Age = 6 };

a2.Born();

a2.Speak("汪汪...");

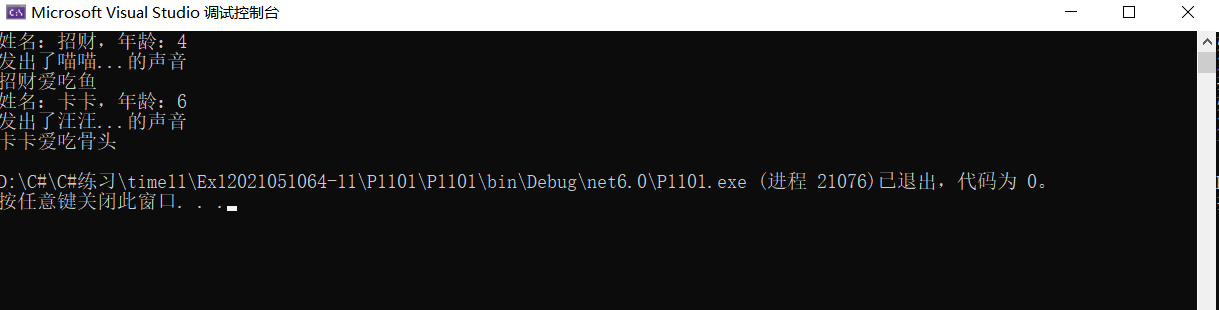
a2.Eat("骨头");

Console.ReadKey();

}

}

//粘贴运行结果界面截图



**任务2、抽象类的多态【项目名P1102】**

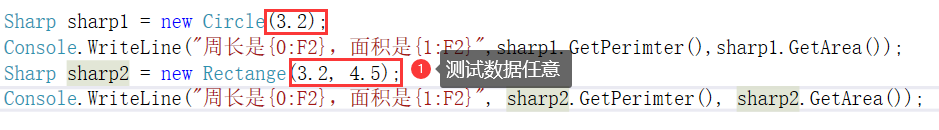
要求：C#使用多态（抽象类）求面积和周长

|  |
| --- |
| 抽象类Sharp |
| + GetArea()：double //抽象方法，面积  + GetPerimter();double //抽象方法，周长 |

|  |
| --- |
| Circle(Sharp派生类) |
| - r：double //半径  定义r的读写属性 |
| + Circle(double r) //构造函数，1个参数  + GetArea()：double //重写抽象方法，计算面积  + GetPerimter();double //重写抽象方法，计算周长 |

|  |
| --- |
| Circle(Sharp派生类) |
| - width：double //宽  - height：double //高  定义width和height的读写属性 |
| + Circle(double width,double height) //构造函数，1个参数  + GetArea()：double //重写抽象方法，计算面积  + GetPerimter();double //重写抽象方法，计算周长 |

**主函数部分**



**程序源程序代码：**粘贴program.cs中代码

abstract class Sharp

{

public abstract double GetArea();

public abstract double GetPerimeter();

}

class Cirlce : Sharp

{

private double r;

public double R

{

get { return r; }

set { r = value; }

}

public Cirlce(double r)

{

this.r = r;

}

public override double GetArea()

{

return Math.PI \* r \* r;

}

public override double GetPerimeter()

{

return 2 \* Math.PI \* r;

}

}

class Rectangle : Sharp

{

private double width;

public double Width

{

get { return width; }

set { Width = value; }

}

private double height;

public double Height

{

get { return height; }

set { height = value; }

}

public Rectangle(double width, double height)

{

this.width = width;

this.height = height;

}

public override double GetArea()

{

return width \* height;

}

public override double GetPerimeter()

{

return 2 \* (width + height);

}

}

class Program

{

static void Main(string[] args)

{

Sharp sharp1 = new Cirlce(3.2);

Console.WriteLine("周长是{0:F2},面积是{1:F2}", sharp1.GetPerimeter(), sharp1.GetArea());

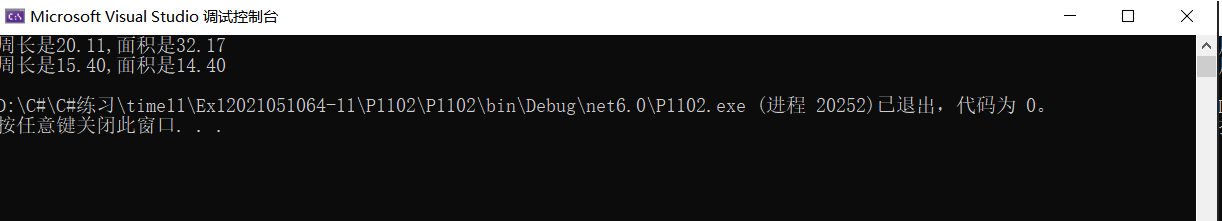
Sharp sharp2 = new Rectangle(3.2, 4.5);

Console.WriteLine("周长是{0:F2},面积是{1:F2}", sharp2.GetPerimeter(), sharp2.GetArea());

}

}

//粘贴运行结果界面截图

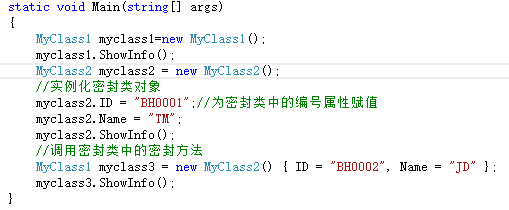


**任务3、虚方法，主函数【项目名P1103】**

**要求：**声明一个类MyClass1，该类中声明了一个虚方法ShowInfo，用来显示信息。然后声明一个密封类MyClass2，继承自MyClass1类，在MyClass2密封类中声明两个公共属性，分别用来表示用户编号（id）和名称（name），然后密封并重写MyClass1基类中的虚方法ShowInfo，并提供具体的实现。

最后在主程序类Program的Main方法中实例化MyClass2密封类的一个对象，然后使用该对象访问myClass2密封类中的公共属性和密封方法。

主函数参考：



补充要求：1、声明非密封类MyClass3，继承自MyClass2，并实例化mycalss3，分析错误。

2、调用密封方法ShowInfo，分析错误。

**源程序代码：**粘贴program.cs中代码

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication3

{

class MyClass1

{

public virtual void ShowInfo()

{

}

}

sealed class MyClass2:MyClass1

{

private string id;

public string Id

{

get { return id; }

set { id = value; }

}

private string name;

public string Name

{

get { return name; }

set { name = value; }

}

public sealed override void ShowInfo()

{

Console.WriteLine("我的学号为{0},我的名字叫做{1}",id,name);

}

}

class Program

{

static void Main(string[] args)

{

MyClass1 myclass1 = new MyClass1();

myclass1.ShowInfo();

MyClass2 myclass2 = new MyClass2();

myclass2.Id = "12021051064";

myclass2.Name = "华心童";

myclass2.ShowInfo();

MyClass1 myclass3 = new MyClass2() {Id="12021051064",Name="Jason" };

myclass3.ShowInfo();

Console.ReadKey();

}

}

}

//粘贴运行结果界面截图

