

## 实验报告

实验名称	实验3 抽象类和接口			实验时间	第 10 周		
实验环境							
Eclipse/Jcreator Pro/JDK。							
实验目的和内容要求							
<p>一、实验目的</p> <p>1. 掌握抽象类和接口的设计方法。</p> <p>二、实验要求</p> <p>1. 提供所完成的各道题主要实验代码和运行结果的界面截图。</p> <p>2. 简述在实验过程中遇到的问题与解决方法。</p> <p>3. 简述实验过程中的发现与收获，未解决或需进一步解决的问题。</p> <p>三、实验内容</p> <p>1. 定义一个抽象类 Shape，它包含一个抽象方法 getArea()，从 Shape 类派生出 Rectangle 和 Circle 类，这两个类都用 getArea()方法计算对象的面积。编写应用程序使用 Rectangle 和 Circle 类。实验要求：定义 Rectangle 和 Circle 类的构造方法，初始化成员变量。</p> <p>2. 学校中有老师和学生两类人，而在职研究生既是老师又是学生。实验要求：在 Graduate 中实现各个接口声明的抽象方法。</p> <p>(1) 设计两个信息管理接口 StudentInterface 和 TeacherInterface，其中 StudentInterface 接口包括 setFee()方法和 getFee()方法，分别用于设置和获取学生的学费，TeacherInterface 接口包括 setPay()和 getPay()方法，分别用于设置和获取老师的工资。</p> <p>(2) 定义一个研究生类 Graduate，实现 StudentInterface 接口和 TeacherInterface 接口，它定义的成员变量有 name（姓名），sex（性别），age（年龄），fee（每学期学费） pay（月工资）。</p> <p>(3) 创建一个姓名为“zhangsan”的研究生，统计他的年收入和学费，如果收入减去学费不足 2000 元，则输出“You need a loan!”（需要贷款）的信息，否则输出“You income is enough!”信息。</p>							
实验过程、结果分析与总结							
<p>第 1 题：</p> <p>源代码如下：</p> <p>Shape.java</p> <pre>public abstract class Shape {     public abstract double getArea(); }</pre> <p>Rectangle.java</p> <pre>public class Rectangle extends Shape {</pre>							

```
private double length;
private double width;

public Rectangle() {

}

public Rectangle(double length, double width) {
    this.length = length;
    this.width = width;
}

@Override
public double getArea() {
    return length * width;
}
}
```

#### Circle.java

```
public class Circle extends Shape {
    private double radius;

    public Circle() {

    }

    public Circle(double radius) {
        this.radius = radius;
    }

    @Override
    public double getArea() {
        return Math.PI * radius * radius;
    }
}
```

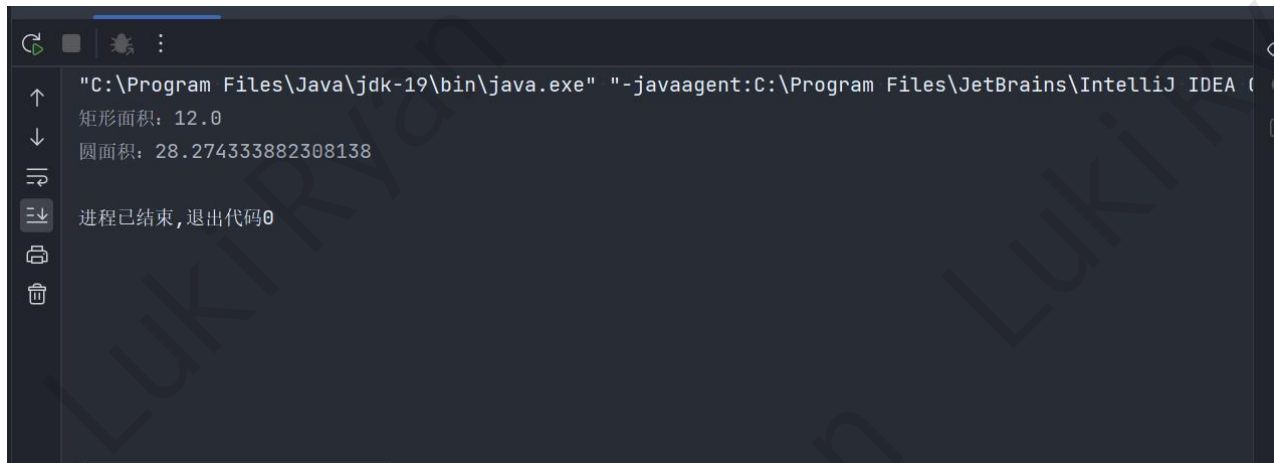
- # 多态的三个必要条件
- ## 继承
- ## 方法的重写
- ## 父类引用子类对象

#### test.java

```
public class test {
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle(3, 4);
    }
}
```

```
Circle circle = new Circle(3);  
System.out.println("矩形面积: " + rectangle.getArea());  
System.out.println("圆面积: " + circle.getArea());  
}  
}
```

运行结果:



```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA (
矩形面积: 12.0
圆面积: 28.274333882308138
进程已结束,退出代码0
```

## 第 2 题:

源代码如下:

StudentInterface.java

```
public interface StudentInterface {  
    public abstract void setFee(int fee);  
    public abstract int getFee();  
}
```

TeacherInterface.java

```
public interface TeacherInterface {  
    public abstract void setPay(int pay);  
    public abstract int getPay();  
}
```

Graduate.java

```
public class Graduate implements StudentInterface, TeacherInterface{  
    private String name;  
    private String sex;  
    private int age;  
    private int fee;  
    private int pay;  
  
    public Graduate() {  
  
    }  
  
    public Graduate(String name, String sex, int age){  
        this.name = name;  
        this.sex = sex;  
        this.age = age;  
    }  
  
    @Override  
    public void setFee(int fee){  
        this.fee = fee;  
    }  
  
    @Override  
    public int getFee() {  
        return this.fee;  
    }  
}
```

```
@Override
public void setPay(int pay) {
    this.pay = pay;
}

@Override
public int getPay() {
    return this.pay;
}
}
```

# 多态的实现方式

## 接口实现

## 继承父类进行方法重写

## 同一个类中进行方法重载

## 父类引用指向子类对象

test.java

```
import java.util.Scanner;

public class test {
    public static void main(String[] args) {

        Graduate zhangsan = new Graduate("zhangsan", "男", 20);

        System.out.println("请输入每月工资:");
        Scanner sc1 = new Scanner(System.in);
        zhangsan.setPay(sc1.nextInt());

        System.out.println("请输入每学期学费:");
        Scanner sc2 = new Scanner(System.in);
        zhangsan.setFee(sc2.nextInt());

        int pays = zhangsan.getPay() * 12;
        int fees = zhangsan.getFee() * 2;
        int earn = pays - fees;
        // 测试
        System.out.println(pays + "-" + fees + "=" + earn);

        if(earn <= 2000) {
            System.out.println("You need a loan!");
        } else {
            System.out.println("You income is enough!");
        }
    }
}
```

```
}
```

```
运行 test x
请输入每月工资:
3000
请输入每学期学费:
8000
36000-16000=20000
You income is enough!

进程已结束,退出代码0
```

```
运行 test x
请输入每月工资:
1500
请输入每学期学费:
8000
18000-16000=2000
You need a loan!

进程已结束,退出代码0
```

```
运行 test x
请输入每月工资:
250
请输入每学期学费:
8000
3000-16000=-13000
You need a loan!

进程已结束,退出代码0
```

在完成这个实验过程中，我遇到了一些问题，并且学到了一些解决方法。这些问题和解决方法包括：

**问题 1：**如何定义抽象类和接口？

**解决方法：**我学到了抽象类是通过使用关键字“abstract”来定义的，并且可以包含抽象方法。接口是通过使用关键字“interface”来定义的，并且接口中的方法默认是抽象的。在实验中，我定义了一个名为 Shape 的抽象类，并在其中声明了一个抽象方法 getArea()。我还定义了两个接口 StudentInterface 和 TeacherInterface，并在 Graduate 类中实现了这两个接口的抽象方法。

**问题 2：**如何创建类的构造方法并初始化成员变量？

**解决方法：**我学到了构造方法是用于创建对象时初始化对象的方法。在实验中，我为 Rectangle 和 Circle 类定义了构造方法，并在构造方法中初始化了成员变量。构造方法的命名与类名相同，并且可以带有参数，用于接收对象的初始化值。

在实验过程中，我也有一些发现和收获。其中包括：

**发现 1：**抽象类和接口提供了一种良好的代码组织和设计方式。通过定义抽象类，我可以将通用的方法和属性提取到一个基类中，并让子类实现自己的特定逻辑。接口则提供了一种合同式的实现方式，确保实现了接口的类都要实现接口中定义的方法。

**收获 1：**我进一步理解了面向对象编程的概念和原则。抽象类和接口的使用使得代码更具可扩展性和可维护性，提高了代码的重用性和灵活性。

总体而言，通过完成这个实验，我学到了关于抽象类、接口、构造方法和面向对象编程的一些基本知识。我也意识到了自己在某些方面仍然需要进一步学习和实践，以提高自己的编程技能和理解能力。

指导老师评阅意见

指导老师： 年 月 日