**《面向对象程序设计(Java)》**

**实验报告**

姓 名 石家晖

学 号 2019081238

学 院 软件工程学院

专 业 软件工程

年 级 2019级

指导教师 文立玉

**成都信息工程大学软件工程学院**

**2020年9月**

# 实验3：

1. **实验目的**

1. 通过编程和上机实验理解 Java 语言是如何体现面向对象编程基本思想，了解类的封装方法，以及 如何创建类和对象，了解成员变量和成员方法的特性，掌握 OOP 方式进行程序设计的方法 。

1. **实验要求**

1. 实验在2学时内完成。

2. 1人1组独立完成。

1. **实验设备与环境**

JDK13, NEOVIM，Linux

1. **设计思路与具体实现**

1. 编写一个圆环类 Ring 的 Java 程序。圆环类有 3 个数据成员 , 分别是内半径 innerRadius, 外半径 outerRadius 和颜色 color, 这些属性可以查看 get 也可以重新设置 set, 另外 , 圆环还可以返回其面积 area 。

(1) 设计思路

编写一个圆环类 Ring 的 Java 程序。圆环类有 3 个数据成员 , 分别是内半径 innerRadius, 外半径 outerRadius 和颜色 color, 这些属性可以查看 get 也可以重新设置 set, 另外 , 圆环还可以返回其面积 area 。

(2) 实验步骤

①、创建类Ring

②、在类中创建 3 个数据成员 , 分别是内半径 innerRadius, 外半径 outerRadius 和颜色 color。并编写get方法和set方法以查看和修改这些属性。最后 , 编写方法返回其面积 area 。 并设计Ring的构造函数。

③、在main函数中创建Ring对象，并且对其get方法、set方法、和返回其area的方法进行测试。

(3) 关键代码

class Ring {

private double innerRadius;

private double outerRadius;

public String color;

public static double PI = 3.1415926;

public Ring (double iRadius, double oRadius, String c) {

this.innerRadius = iRadius;

this.outerRadius = oRadius;

this.color = c;

num++;

}

public double getInnerRadius() {

return innerRadius;

}

public void setInnerRadius (double innerRadius) {

this.innerRadius = innerRadius;

}

public double getOuterRadius() {

return outerRadius;

}

public void setOuterRadius (double outerRadius) {

this.outerRadius = outerRadius ;

}

public String getColor() {

return color;

}

public void setColor (String color) {

this.color = color;

}

public double getArea() {

return PI \* (getOuterRadius() \* getOuterRadius() - getInnerRadius() \* getInnerRadius() );

}

public double getInCircle() {

return 2 \* PI \* getInnerRadius();

}

public double getOutCircle() {

return 2 \* PI \* getOuterRadius();

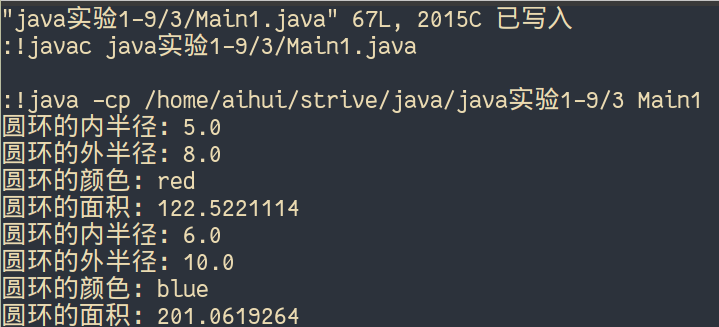
}

}

(4) 程序测试过程

在main函数中输入测试数据，调用Ring类的所有方法，观察输出的正确与否。

1. 运行结果（运行效果）



2. 第1题中增加两个static 成员 ：圆周率和圆对象个数, 增加两个 static 方法，分别是设置圆周率和显示当前圆对象个数的, 仔细体会静态成员与实例成员的使用方法和区别。

(1) 设计思路

第1题中增加两个static 成员 ：圆周率和圆对象个数, 增加两个 static 方法，分别是设置圆周率和显示当前圆对象个数的, 仔细体会静态成员与实例成员的使用方法和区别。

(2) 实验步骤

①、在Ring类中增加两个static 成员 ：圆周率PI和圆对象个数num。

②、增加两个 static 方法，分别是设置圆周率和显示当前圆对象个数的。

③、在主函数中进行测试。

(3) 关键代码

public static double PI = 3.1415926;

public static int num = 0;

//static

public static void setPI (double pi) {

PI = pi;

}

public static int getNum() {

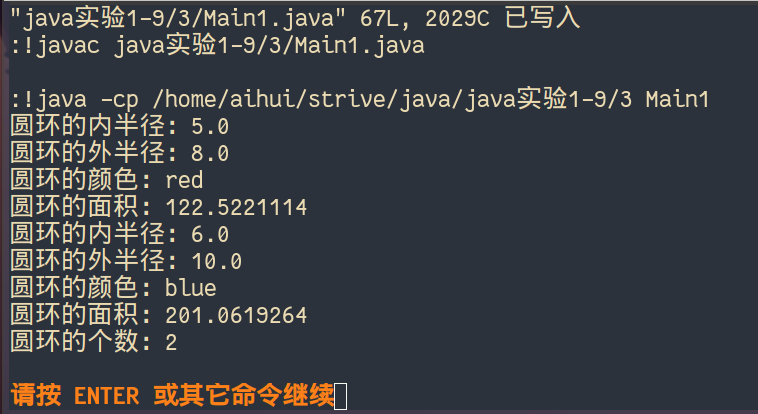
return num;

}

(4) 程序测试过程

在main函数中输入测试数据，调用Ring类的两个static方法，观察输出的正确与否。

(5) 运行结果（运行效果）



3.设计一个教材类，一个课程类，及一个为某门课程指定参考教材的类。一门课程可以有多本参考教材，再设计一个测试类。

(1) 设计思路

设计一个教材类，一个课程类，及一个为某门课程指定参考教材的类。一门课程可以有多本参考教材，再设计一个测试类。

(2) 实验步骤

①、创建4个类，分别是教材类、课程类、为某门课程指定参考教材的类、测试类。

②、确定好四个类的关系，设计实现这四个类。

③、创建测试类对象进行类的测试。

(3) 关键代码

class Classes {

private String name;

private NameBooks nameBooks = new NameBooks();

public Classes (String name) {

this.name = name;

}

public String getName() {

return name;

}

public void addBooks (String booksName) {

nameBooks.addBooks (booksName);

}

public void putAllBooks() {

for (int i = 0; i < nameBooks.getNum() ; i++ )

System.out.println ("第" + (i + 1) + "教材:" + nameBooks.getBooks (i) );

}

}

class Books {

private String name;

public void setName (String name) {

this.name = name;

}

public String getName() {

return name;

}

}

class NameBooks {

Books books[] = new Books[100];

int num = 0;

public void addBooks (String booksName) {

books[num] = new Books();

books[num].setName (booksName);

num++;

}

public String getBooks (int num) {

return books[num].getName();

}

public int getNum() {

return num;

}

}

class Test {

public void testClasses() {

Classes a = new Classes ("Chinese");

a.addBooks ("Chinese Book I");

a.addBooks ("Chinese Book II");

a.addBooks ("Chinese Book III");

a.putAllBooks();

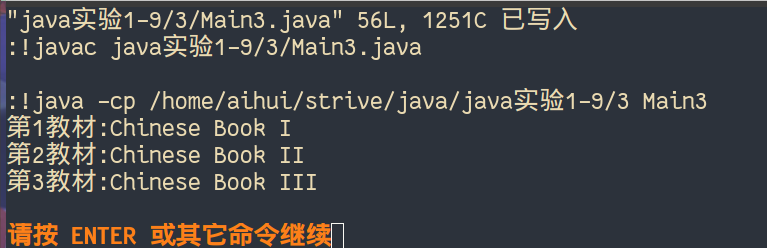
}

}

(4) 程序测试过程

由于设计了测试类，可以直接在main函数中创建测试类对象进行类的测试。

(5) 运行结果（运行效果）



1. **实验总结**

理解了 Java 语言是如何体现面向对象编程基本思想，了解了类的封装方法，以及如何创建类和对象。

了解了成员变量和成员方法的特性，基本掌握了OOP 方式进行程序设计的方法 。

1. **附录（程序代码）**

1&2.

public class Main1 {

public static void main (String[] args) {

Ring ring1 = new Ring (5, 8, "red");

System.out.println ("圆环的内半径：" + ring1.getInnerRadius() );

System.out.println ("圆环的外半径：" + ring1.getOuterRadius() );

System.out.println ("圆环的颜色：" + ring1.getColor() );

System.out.println ("圆环的面积：" + ring1.getArea() );

Ring ring2 = new Ring (5, 8, "red");

ring2.setInnerRadius (6);

ring2.setOuterRadius (10);

ring2.setColor ("blue");

System.out.println ("圆环的内半径：" + ring2.getInnerRadius() );

System.out.println ("圆环的外半径：" + ring2.getOuterRadius() );

System.out.println ("圆环的颜色：" + ring2.getColor() );

System.out.println ("圆环的面积：" + ring2.getArea() );

System.out.println ("圆环的个数：" + ring2.getNum() );

}

}

class Ring {

private double innerRadius;

private double outerRadius;

public String color;

public static double PI = 3.1415926;

public static int num = 0;

public Ring (double iRadius, double oRadius, String c) {

this.innerRadius = iRadius;

this.outerRadius = oRadius;

this.color = c;

num++;

}

public double getInnerRadius() {

return innerRadius;

}

public void setInnerRadius (double innerRadius) {

this.innerRadius = innerRadius;

}

public double getOuterRadius() {

return outerRadius;

}

public void setOuterRadius (double outerRadius) {

this.outerRadius = outerRadius ;

}

public String getColor() {

return color;

}

public void setColor (String color) {

this.color = color;

}

public double getArea() {

return PI \* (getOuterRadius() \* getOuterRadius() - getInnerRadius() \* getInnerRadius() );

}

public double getInCircle() {

return 2 \* PI \* getInnerRadius();

}

public double getOutCircle() {

return 2 \* PI \* getOuterRadius();

}

//static

public static void setPI (double pi) {

PI = pi;

}

public static int getNum() {

return num;

}

}

3.

public class Main3 {

public static void main (String[] args) {

Test test = new Test();

test.testClasses();

}

}

class Classes {

private String name;

private NameBooks nameBooks = new NameBooks();

public Classes (String name) {

this.name = name;

}

public String getName() {

return name;

}

public void addBooks (String booksName) {

nameBooks.addBooks (booksName);

}

public void putAllBooks() {

for (int i = 0; i < nameBooks.getNum() ; i++ )

System.out.println ("第" + (i + 1) + "教材:" + nameBooks.getBooks (i) );

}

}

class Books {

private String name;

public void setName (String name) {

this.name = name;

}

public String getName() {

return name;

}

}

class NameBooks {

Books books[] = new Books[100];

int num = 0;

public void addBooks (String booksName) {

books[num] = new Books();

books[num].setName (booksName);

num++;

}

public String getBooks (int num) {

return books[num].getName();

}

public int getNum() {

return num;

}

}

class Test {

public void testClasses() {

Classes a = new Classes ("Chinese");

a.addBooks ("Chinese Book I");

a.addBooks ("Chinese Book II");

a.addBooks ("Chinese Book III");

a.putAllBooks();

}

}