实验课程名称：Java语言程序设计A

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **实验项目名称** | **实验5：接口** | | | **实验成绩** |  |
| **实 验 者** |  | **专业班级** | **信管1502** | **组 别** | **1人1组** |
| **同 组 者** | **无** | | | **开始日期** | **2018年5月15日** |
| 第一部分：实验预习报告（包括实验目的及意义，实验基本原理与方法，主要仪器设备及耗材，实验内容及要求，实验方案与技术路线等）  一．实验目的及意义  1．自定义接口。  2．自定义类实现接口。  3．接口及实现类的多态处理。  二．实验基本原理与方法  1．接口的概念。  2．接口对多态的支持。  三．主要仪器设备及耗材  1．PC及其兼容机。  2．计算机操作系统。  3．程序编辑器Vscode。  4．Java开发工具JDK。  四．实验内容及要求  实验5-1：自定义形状接口Shape，该接口声明了计算面积、周长的方法。然后，分别编写正三角形类Triangle、正六边形类Hexagon、椭圆形类Ellipse，它们都实现了Shape接口。最后，编写测试类ShapesDemo，多态地创建各种形状对象，计算面积、周长。  实验报告要求：实验5的实验报告，必须包含思路、算法（含各种形状面积、周长的计算公式）、UML类图、代码、运行结果、问题及解决过程、综合分析、总结、体会。  五．实验方案及技术路线（含各种形状面积、周长的计算公式，UML类图，注意事项）  实验5技术路线：  5-1各个形状的面积和周长公式：  5-1.1{  正三角形的周长 = 边长\*3；  正三角形的面积 = 边长的平方 \* 根号（3）/ 4；  }  5-1.2{  正六边形的周长 = 边长\*6；  正六边形的面积 = 6个相同边长的正三角形的面积之和；  }  5-1.3{  椭圆的周长 = 2 \* PI \* 短半轴长 + 4 \*（长半轴长 – 短半轴长）；  椭圆的面积 = PI \* 短半轴长 \* 长半轴长；  }  5-1UML类图：    第二部分：实验过程记录（可加页）（代码、运行结果、实验中出现的问题及解决过程）  一、实验代码：  import java.util.\*;  public class ShapesDemo  {  public static void main(String[] args)  {  // TODO Generate method automatically  Shape[] shapes = {new Triangle(1),new Hexagon(2.5),new Ellipse(5,3.2)};  for (Shape item: shapes)  {  System.out.println(item.toString());  }  }}  interface Shape  {  public String toString();  // This method is to output Name, area, perimeter  public double Calculate\_Area(double a,double b);  public double Calculate\_Perimeter(double a,double b);  // Special Description: when the abstract method is defined,  // the number of parameters should be as large as possible to satisfy the elliptical class with a slightly complicated circumference area.  //For a regular triangle and regular hexagon with only one parameter, no parameter B can be used.  }  class Triangle implements Shape  {  private double length;  public Triangle (double lengthIn)  {  length=lengthIn;  System.out.println("equilateral triangle's border length is initialized by "+length);  }  public String toString()  {  return ("equilateral triangle,the area and perimeters is:"+String.format("%.2f",Calculate\_Area(length,0.1))+","+String.format("%.2f",Calculate\_Perimeter(length,0.1)));  }  public double getLength()  {  return length;  }  public double Calculate\_Area(double length, double b)  {  return (length\*length\*Math.sqrt(3)/4);  }  public double Calculate\_Perimeter(double length,double b)  {  return (3\*length);  }  }  class Hexagon implements Shape  {  private double length;  public Hexagon (double lengthIn)  {  length = lengthIn;  System.out.println("regular hexagon's border length is initialized by "+length);  }  public String toString()  {  return ("regular hexagon,the area and perimeters is:"+String.format("%.2f",Calculate\_Area(length,0.1))+","+String.format("%.2f",Calculate\_Perimeter(length,0.1)));  }  public double getLength()  {  return length;  }  public double Calculate\_Area(double length,double b)  {  return (length\*length \* Math.sqrt(3)/4\*6);  }  public double Calculate\_Perimeter(double length,double b)  {  return (6\*length);  }  }  class Ellipse implements Shape  {  private double Half\_LongLength;  private double Half\_ShortLength;    public Ellipse (double Half\_LongLengthIn, double Half\_ShortLengthIn)  {  Half\_LongLength=Half\_LongLengthIn;  Half\_ShortLength=Half\_ShortLengthIn;  System.out.println("Ellipse's Half\_LongLength length is initialized by "+Half\_LongLength);  System.out.println("Ellipse's Half\_ShortLength length is initialized by "+Half\_ShortLength);  }  public String toString()  {  return ("Ellipse,the area and perimeters is:"+String.format("%.2f",Calculate\_Area(Half\_LongLength,Half\_ShortLength))+","+String.format("%.2f",Calculate\_Perimeter(Half\_LongLength,Half\_ShortLength)));  }  public double getLong()  {  return Half\_LongLength;  }  public double getShort()  {  return Half\_ShortLength;  }  public double Calculate\_Area(double LongLength,double ShortLength)  {  return (2\*Math.PI\*ShortLength+4\*(LongLength-ShortLength));  }  public double Calculate\_Perimeter(double LongLength,double ShortLength)  {  return (Math.PI \* LongLength \* ShortLength);  }  }  二、运行结果  equilateral triangle's border length is initialized by 1.0  regular hexagon's border length is initialized by 2.5  Ellipse's Half\_LongLength length is initialized by 5.0  Ellipse's Half\_ShortLength length is initialized by 3.2  equilateral triangle,the area and perimeters is:0.43,3.00  regular hexagon,the area and perimeters is:16.24,15.00  Ellipse,the area and perimeters is:27.31,50.27  第三部分 结果与讨论（可加页）  一、实验结果分析  第一，根据继承的定义，某个具体的类如果继承了某一个接口，那么该类必须实现接口中所有的方法，（方法的类型，方法的参数个数，形式，顺序都必须相同）；  第二，Java为了维护类的安全，不允许父类直接访问子类的方法或变量成员。在本实验中，多台数组保存Shapes对象，如果想要利用数组中每一个实例访问其子类的成员变量（比如边长）是无法做到的。为此，我将toSting()方法抽象出来，通过公共方法来让每一个实例都能够将计算所得到的面积和周长输出出来。  第三，Java是单继承语言，不允许继承多个类，但是可以继承多个接口；  二、小结、建议及体会  Java是纯面向对象的编程方式，继承和抽象的思想能够让简化代码，实现更好的逻辑。这次试验主要是对接口的理解，对接口中方法的实现，在编码中体会多态的思想。代码量相对较小，实验中我主要工作是在如何抽象类，如何高效的重复利用代码，让程序更简洁，清晰。  第四部分 实验评分标准（教师自行设计）及成绩   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | 观测点 | 考核目标 | 权重 | 得分 | | 实验预习 | 程序设计方案、技术路线（设计思路、算法、注意事项）科学、合理、可行 | 实验方案设计能力 | 20% |  | | 实验过程 | 1. 按时参加实验，不迟到，不早退 2. 源代码正确、完整、可读性强、编码规范，运行结果正确，实验中出现的问题得到解决 | 学生的实验态度、UML面向对象建模与设计能力、分析并解决问题的专业素养及编程技能、良好的编程风格与习惯 | 50% |  | | 结果分析 | 实验结果分析到位、总结完善、体会深刻 | 学生对实验数据处理与分析的能力；对专业知识的综合应用能力；事实求实的精神 | 30% |  | | 该项实验报告最终得分 | | |  | |   教师签名： | | | | | |