桂林电子科技大学

**Java应用开发** 实验报告

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | **Java面向对象编程2-1** | | | | | | | | 指导意见：    成绩  指导教师 李云辉 |
| 院 系 | 计算机与信息安全学院 | | | 专业 | | | 计算机科学与技术 | |
| 学 号 | 2000300414 | | | 姓名 | | | 黄武钊 | |
| 同作者 |  | | | | | | | |
| 实验日期 | 2022 | 年 | 5 | | 月 | 24 | | 日 |
|  |  | | | | | | | |

**实验题目： 中国人、北京人和美国人**

1. **实验目的：**本实验的目的是让学生巩固下列知识点:

子类的继承性。

子类对象的创建过程。

成员变量的继承与隐藏。

方法的继承与重写。

1. **实验要求：**

编写程序模拟中国人、美国人是人，北京人是中国人。除主类外，程序中还有4 个类：

People、ChinaPeople、AmericanPeople 和BeijingPeople 类。要求如下。

* People 类有权限是protected 的double 型成员变量height 和weight，以及public void speakHello()、public void averageHeight()和public void averageWeight()方法。
* ChinaPeople 类是People 的子类，新增了public void chinaGongfu()方法。要求：

ChinaPeople 重写父类的public void speakHello()、public void averageHeight()和public void averageWeight()方法。

* AmericanPeople 类是People 的子类，新增public void americanBoxing()方法。要求：

AmericanPeople 重写父类的public void speakHello()、public void averageHeight()和public void averageWeight()方法。

* BeijingPeople 类是ChinaPeople 的子类，新增public void beijingOpera()方法。要求ChinaPeople 重写父类的public void speakHello()、public void averageHeight()和public void averageWeight()方法。
* People、ChinaPeople、AmericanPeople 和BeijingPeople 类的UML 图如图所示。

1. **实验设计：**画出UML类图



1. **实验代码**（补全代码，代码字体 Courier 10pts，不用截图）

**//People.java**

package step1;  
public class People {  
 protected double weight,height;  
 public void speakHello() {  
 System.*out*.println("yayayaya");  
 }  
 public void averageHeight() {  
 height=173;  
 System.*out*.println("average height:"+height);  
 }  
 public void averageWeight() {  
 weight=70;  
 System.*out*.println("average weight:"+weight);  
 }  
}

**//ChinaPeople.java**

package step1;  
  
*//ChinaPeople.java*public class ChinaPeople extends People  
{  
 public void speakHello()  
 {  
 System.*out*.println("您好");  
 }  
  
 public void averageHeight()  
 {  
 height = 168.78;  
 System.*out*.println("中国人的平均身高:" + height + " 厘米");  
 }  
  
 *//【代码1】重写public void averageWeight()方法，输出:"中国人的平均体重：65公斤"* @Override  
 public void averageWeight()  
 {  
 System.*out*.println("中国人的平均体重：65公斤");  
 }  
  
 public void chinaGongfu()  
 {  
 System.*out*.println("坐如钟,站如松,睡如弓");  
 }  
}

**//AmericanPeople.java**

package step1;  
public class AmericanPeople extends People {  
 *//【代码2】重写public void speakHello()方法，输出"How do you do"  
 //【代码3】 重写public void averageHeight()方法，输出"American's average height:176 cm"* public void averageWeight() {  
 weight = 75;  
 System.*out*.println("American's average weight:"+weight+" kg");  
 }  
 public void americanBoxing() {  
 System.*out*.println("直拳、钩拳、组合拳");  
 }  
  
 @Override  
 public void speakHello()  
 {  
 System.*out*.println("How do you do");  
 }  
  
 @Override  
 public void averageHeight()  
 {  
 System.*out*.println("American's average height:176 cm");  
 }  
}

**//BeijingPeople.java**

public class BeijingPeople extends ChinaPeople  
{  
 *//【代码4】重写public void averageHeight()方法,输出:"北京人的平均身高: 172.5厘米"  
 //【代码5】重写public void averageWeight()方法,输出:"北京人的平均体重：70公斤"* public void beijingOpera()  
 {  
 System.*out*.println("花脸、青衣、花旦和老生");  
 }  
  
 @Override  
 public void averageHeight()  
 {  
 System.*out*.println("北京人的平均身高: 172.5厘米");  
 }  
  
 @Override  
 public void averageWeight()  
 {  
 System.*out*.println("北京人的平均体重：70公斤");  
 }  
}

**//Example.java**

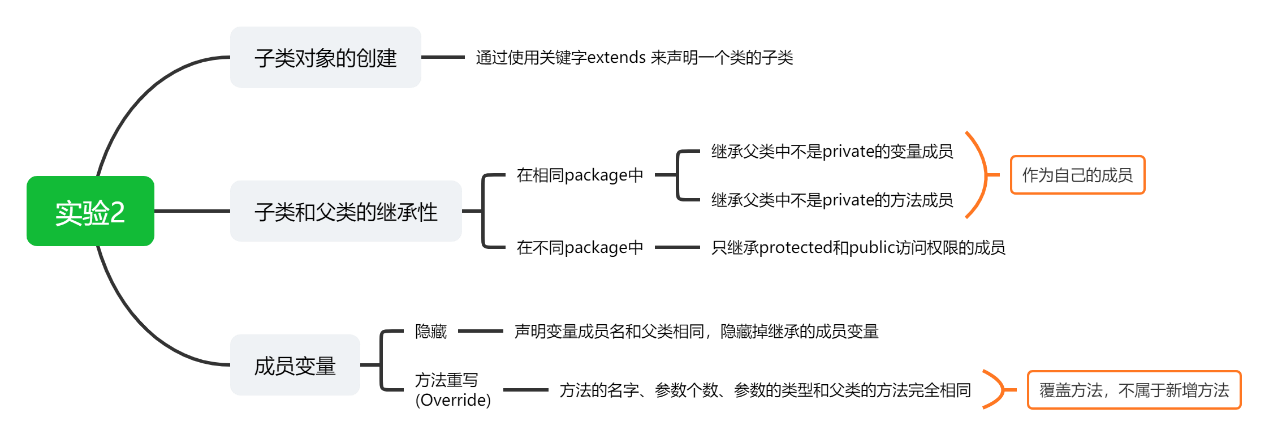
package step1;  
public class Example {  
 public static void main(String args[]) {  
 ChinaPeople chinaPeople=new ChinaPeople();  
 AmericanPeople americanPeople=new AmericanPeople();  
 BeijingPeople beijingPeople=new BeijingPeople();  
 chinaPeople.speakHello();  
 americanPeople.speakHello();  
 beijingPeople.speakHello();  
 chinaPeople.averageHeight();  
 americanPeople.averageHeight();  
 beijingPeople.averageHeight();  
 chinaPeople.averageWeight();  
 americanPeople.averageWeight();  
 beijingPeople.averageWeight();  
 chinaPeople.chinaGongfu();  
 americanPeople.americanBoxing();  
 beijingPeople.beijingOpera() ;  
 beijingPeople.chinaGongfu();  
 }  
}

1. **实验结果**

**图形用户界面, 文本, 应用程序

描述已自动生成**

1. **实验知识点总结**

****

**------------**

**扩展：**将People设计为抽象类，试着修改代码，体验多态的实现。

桂林电子科技大学

**Java应用开发** 实验报告

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | **Java面向对象编程2-2** | | | | | | | | 指导意见：    成绩  指导教师 李云辉 |
| 院 系 | 计算机与信息安全学院 | | | 专业 | | 计算机科学与技术 | | |
| 学 号 | 2000300414 | | | 姓名 | | 黄武钊 | | |
| 同作者 |  | | | | | | | |
| 实验日期 | 2022 | 年 | 5 | | 月 | | 24 | 日 |
|  |  | | | | | | | |

1. **实验目的：**设计和实现 comparable 接口[[1]](#footnote-2)
2. **实验要求：**

* 修改 CeometricObject 类以实现 Comparable 接口；
* 并且在 GeometricObject 类中定义一个静态的max方法，用该方法求两个圆中的较大者。

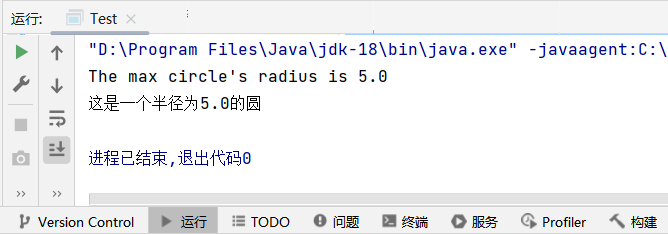
1. **实验设计：**画出UML类图



1. **实验代码**（补全代码，代码字体 Courier 10pts，不用截图）

package step2;  
  
public class Test  
{  
 *// Main method* public static void main(String[] args)  
 {  
 *// Create two comparable circles* Circle1 circle1 = new Circle1(5);  
 Circle1 circle2 = new Circle1(4);  
  
 *// Display the max circle* Circle1 circle = (Circle1) GeometricObject1.*max*(circle1, circle2);  
 System.*out*.println("The max circle's radius is " +  
 circle.getRadius());  
 System.*out*.println(circle);  
 }  
}  
  
abstract class GeometricObject1 implements Comparable  
{  
 public static Object max(Circle1 circle1, Circle1 circle2)  
 {  
 if (circle1.compareTo(circle2) > 0)  
 {  
 return circle1;  
 } else if (circle1.compareTo(circle2) < 0)  
 {  
 return circle2;  
 }  
 return 0;  
 }  
}  
  
*// Circle.java: The circle class that extends GeometricObject*class Circle1 extends GeometricObject1  
{  
 *// Implement it* double radius;  
  
 public Circle1(double i)  
 {  
 this.radius = i;  
 }  
  
 public double getRadius()  
 {  
 return this.radius;  
 }  
  
 @Override  
 public int compareTo(Object o)  
 {  
 Circle1 c = (Circle1) o;  
 if (this.radius > c.radius)  
 {  
 return 1;  
 } else if (this.radius < c.radius)  
 {  
 return -1;  
 } else  
 {  
 return 0;  
 }  
 }  
  
 @Override  
 public String toString()  
 {  
 String s = "这是一个半径为" + this.radius + "的圆";  
 return s;  
 }  
}

1. **实验结果**

****

1. **实验知识点总结**

图示, Teams

描述已自动生成

1. **如果你想1个类的对象支持比较(排序), 就必须实现Comparable接口.** [↑](#footnote-ref-2)