



8.4

Java 类的复用

基本信息

教学主题	Java 类的复用	课时安排	1 课时(45 分钟)
所在章节	第 9 章 Java 类的继承		

【教学目标】

❖ 知识目标

- (1) 学生能够了解 Java 类的复用，知晓聚合与继承、父类与子类的区别。
- (2) 学生能够了解 Java 访问控制的作用。
- (3) 学生能够掌握 Java 方法覆盖性质。

❖ 能力目标

- (1) 学生能够掌握聚合和继承的应用。
- (2) 学生能够编写具有覆盖方法的程序。

❖ 创新性学习目标

通过讲解 Java 语言重写与继承方法，培养学生理解和动手能力；讲解不同的修饰符的用途，引导学生进行对比分析，使学生深入理解 Java 修饰符的理解，同时培养其归纳总结和对比分析能力。本堂课结合国家正在实施的扶贫措施，提出问题——如何分析和设计现实场景的扶贫系统，理论与实践相结合。

【教学内容】

❖ 主要内容

- (1) Java 类的继承与聚合。
- (2) Java 访问控制。

❖ 地位作用

本堂课是 Java 语言面向对象继承的知识内容，是 Java 面向对象编程技术的基石。本堂课以 Java 面向对象思想为基础，对 Java 面向对象的知识进行更深入的学习，解释何为 Java 面向对象语言，让学生全面掌握 Java 面向对象原理，引导学生采用 Java 语言进行程序设计，使其具备编写 Java 面向对象程序的能力。

【教学重点、难点、创新点及解决措施】

❖ 教学重点及解决措施

- (1) Java 类的复用。
- (2) Java 访问控制符。
- (3) Java 方法重写。



问题牵引教学方法：对于较难理解的基本概念，主要采用提出问题的方式，启发学生积极思考、分析、讨论，寻找解决途径和问题的答案的方法。

❖ 教学难点及解决措施

(1) Java 类的复用——理解 Java 继承与复用的区别。

(2) 掌握 Java 访问控制符的作用范围，能够分别控制符的用途。

“讲、学、练”相结合：采用演示、讲解和练习等方式使学生了解 Java 语言概念、理解 Java 聚合与继承的异同，掌握 Java 控制符作用范围，熟练区分不同修饰符的用途。

❖ 教学创新点及解决措施

本堂课教学创新点包括：针对学生以前学习先导课程 C 语言，跳出教材，通过对比 C 语言与 Java 语言，探讨继承思想的由来与发展，深入讲解 Java 继承方法。这样既能使学生了解到 Java 语言的发展历史，又能够深刻理解 Java 继承的概念。

本堂课通过一个完整实例，讲解 Java 继承、方法控制、覆盖的概念，并全程配图演示，加强学生掌握 Java 语言实践操作技能。

针对学生关注的如何解决现实场景的问题，跳出教材，通过引入现实场景扶贫措施，深入讲解 Java 语言采用面向对象的方法设计现实场景中的类、对象、属性、方法。同时，使用继承、重写方法结合实际案例，这样既使学生了解了国家扶贫政策，又能深刻理解 Java 语言面向对象概念。

▶ 【教学方法】

❖ 思维导图

Java 类的继承、访问控制、覆盖思维导图如图 8.47 所示。

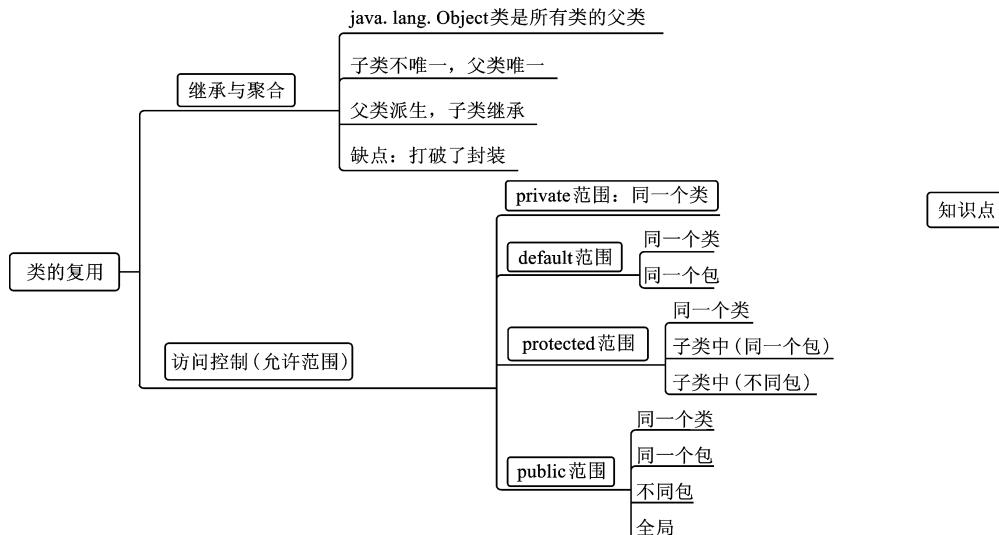


图 8.47 Java 类的继承、访问控制、覆盖思维导图

❖ 教学模式及方法

本堂课采用“课堂授课模式”进行教学，主要环节包括：授课 理解 巩固 运用 总结。

❖ 教学内容重组与加工

本堂课主要涉及的教学内容包括继承思想发展历史、Java 类的复用概念、Java 访问控制符的作用范围、Java 重写方法的应用。考虑到学生只是简单地了解面向对象的概念，为了更好的阐述 Java 语言中继承这一面向对象特性，掌握访问控制符、方法重写，本堂课对教学内容进行了重组，探讨继承思想的由



来与现实应用，深入了解继承思想。同时，为了让学生更好地理解访问控制、重写方法，在课堂教学中以知识为本，实际开发为参考，由浅入深地讲解原理与应用，使同学进一步巩固课堂知识内容。

❖ 教学资源与技术手段

采用“雨课堂”随堂练习实时监控教学进程与课后反思；建设“网上课堂”实现资源共享；结合多媒体教学与传统黑板板书展示多元素教学内容；创设小组讨论、师生问答的互动情境激发课堂活力，提高学生的课堂参与度；设置计算机编程演示环节来帮助学生突破理论与实践之间的藩篱；通过布置课外论文研读任务来培养学生研究性学习的习惯。

❖ 教学信息收集与处理

首先收集目前国家正在实施的扶贫政策和视频，举例展示目前扶贫的理念，抽取扶贫政策与面向对象编程思想相同的关键信息点。同时，本着解决实际问题的思路出发，引导学生思考如何设计扶贫人员和扶贫对象的类，通过图片展示扶贫人员和扶贫对象信息，描述它们所对应的类、属性和方法，给学生直观的展示采用面向对象思想分析、设计的效果，并结合实际情况探讨 Java 修饰符与方法重写的应用。基于上述原理，教师演示自主开发的扶贫系统，使学生既学习 Java 语言面向对象知识，又对国家扶贫攻坚伟大壮举有直观的体会。最后，精选项目源代码案例介绍给学生，引导学生通过研读课外源代码来拓宽自己的知识深度与广度，培养研究性学习习惯以增强创新意识。

❖ 教学参与的切入

本堂课在面向学生主体进行教学过程中，在课前让学生复习 Java 面向对象基础知识，为本课堂的知识学习打下基础，然后探究面向对象编程；在上课时展开分组讨论 C 语言结构体与 Java 类的异同，并进一步引导小组学生代表分享探究时查询到的观点，让他们大胆发表看法，师生共同分析。分析完后，教师讲述 Java 类的特点，调动学生的积极性，给机会、给平台让他们大胆发表对 Java 类的看法，敢向老师质疑，分享探究时查询到的观点，师生共同分析，让学生思想与思想碰撞，智慧与智慧较量。当学生在学习中意识到自己的智慧和力量，既体验到了创造的欢乐，又充分地显露了自己的个性时，就能实现师生间的知识同步、思维共振、感情共鸣。在分析 Java 继承与方法重写时，充分提问，让学生深刻理解 Java 概念。同时，将教师的科研成果、研发的系统演示给学生，让学生在轻松的状态学习 Java 语言，从而激发学生的学习兴趣、释放出巨大的学习潜能。

❖ 教学进程设计

本堂课的教学进程如图 8.48 所示。

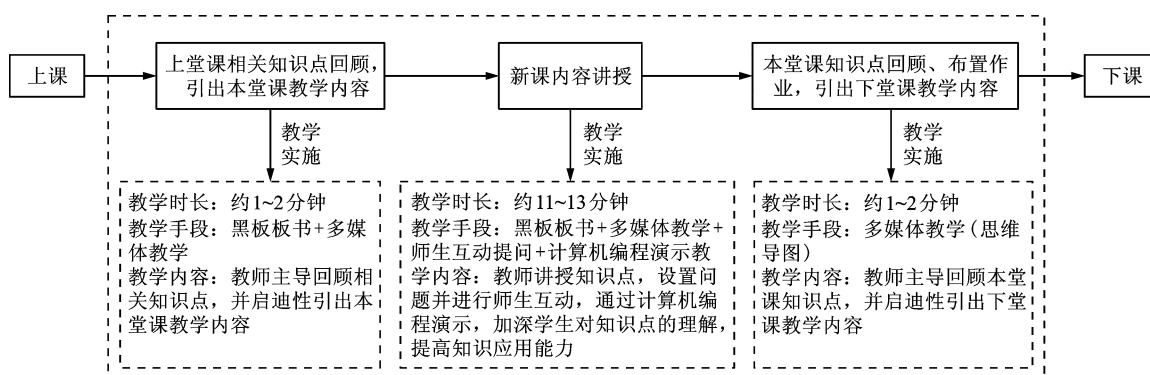


图 8.48 教学进程图

▶ 【课程思政】

本堂课我们可以通过以下几点，让学生更好地结合实际去了解学习内容：

(1) 通过讲解 Java 类的继承是面向对象的一块基石，因为它允许我们分层次地创建类，引导学生从



不同的事物中寻找相同，力图挖掘事物现象本质，本着实事求是的理念与态度，在潜移默化中影响学生的认知角度，从而形成辩证的思想。

(2) 通过讲解 Java 类的继承将本质传递，减少不必要的重复，提升效率，将复杂予以简化，引导学生透过现象看本质，充分利用不同事物之间的联系，使结果更加精简化，培养学生的独立思考、分析、归纳和总结能力，同时锻炼学生实际动手编程能力。

(3) Java 访问控制是用来限制对类、方法、属性的访问，在 Java 语言中，变量的作用域不仅取决于变量声明的位置，还取决于变量访问控制符的修饰。通过讲解不同的访问控制符会对对象采取不同的保护措施，使学生充分认识到数据的重要性，并且培养学生运用正确的方式对隐私数据的保护。

(4) 通过讲解 Java 方法覆盖能够让我们减少重复，充分利用已有的资源，高效使用“复用”意识，培养学生积极思考、开动脑筋的习惯，提升学生思维能力。通过学习 Java 方法覆盖，使学生充分认识到事物的优缺点和异同，通过自己的独立思考将别人的东西转化为自己的知识。

▶【课前探究】

本堂课作为概述课，课前探究使学生对 Java 语言产生兴趣非常重要，教师设计的课前探究内容包括：

- (1) 学生探究 C 语言结构体与 Java 类的异同，带领学生深刻理解类的概念。
- (2) 学生尝试探究主流面向对象语言，了解面向对象语言中 Java 继承的发展历史。



◆ 温习 Java 面向对象概念，展开分组讨论

由教师主导，组织学生复习面向对象的基础知识点，如：类、对象、属性、方法等，在学生复习之后组织学生分组探讨面向对象编程。

学生五人一组，分成九组进行讨论主流程序语言内容，讨论内容以教师要求的讨论内容为主。学生需要回想课前探究的知识点，为本堂课学习做好铺垫。

由教师主导，教师分析学生课前探究的留言：

(1) 在编写相近的多个类时，经常每个类都要编写相同的属性和方法，如何简便操作？

(2) 在 Java 编程中是不是所有的属性和方法都向外公开？

教师点评学生留言具有较好实践探索精神，通过导入扶贫案例回答学生留言。

◆ Java 面向对象编程类的复用

教师引入继承知识。

面向对象最重要的特点之一，就是实现对类的复用。类的复用可通过两种方法实现(见图 8.49)。

【设计意图】

- (1) 通过课前复习巩固学生面向对象的知识。
- (2) 了解学生课前探究的成效，对面向对象思想的理解，为开展 Java 继承教学内容的讲解做铺垫。



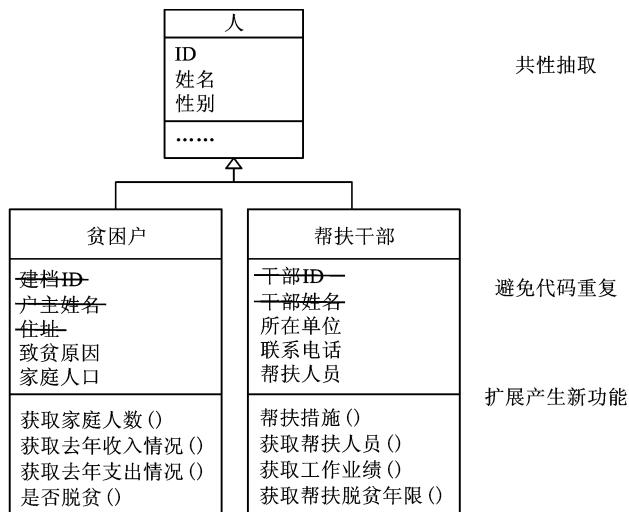
- 一种方法是将一个类当作另一个类的属性(聚合关系)。
- 另一种方法是一个类继承另一个类，实现继承的类称为子类，而被继承的类称为父类，也叫超类或基类。

类的复用，避免代码重复，大大提高工作效率



图 8.49 类的复用示例图

教师解释：类与类之间会有某种关联，像学生类与教师类，而继承是关联的一种。教师类与学生类都包括了人类的属性与行为，所以在开发过程中，教师类和学生类可以继承人类，从而减少定义共同属性和行为的时间，这便是继承的基本思想(见图 8.50)。



当类满足“is of”的关系时，可“继承”一个现有的类，“继承”该类的方法和属性。

图 8.50 继承的基本思想

(1) 教师讲授继承的基本语法。

```
<modifier> class <name> [extends <superclass> ]
{
    <declaration> *
}
```

强调在 Java 中，一个类只能继承一个父类，这种方式叫作单继承(见图 8.51)。这一点和 C++不一样。

【设计意图】

(1) 通过讲解面向对象编程是不断发展的，适应时代需求、不断创新的产物，激发学生的创新意识，让学生了解创新可以创造出影响和改变世界的产品。

(2) 通过讲解面向对象编程语言继承是共性抽取，减少不必要的重复，提升效率，将复杂予以简化。

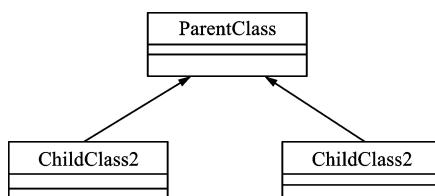


图 8.51 单继承

虽然一个类只可以继承一个父类，但是一个父类却可以被多个子类所继承。

从子类的角度看，它是“继承(extends)”父类；而从父类的角度看，它是“派生(derive)”子类。它们指的都是同一个动作，只是角度不同而已。

(2) 进行代码展示(见图 8.52)。

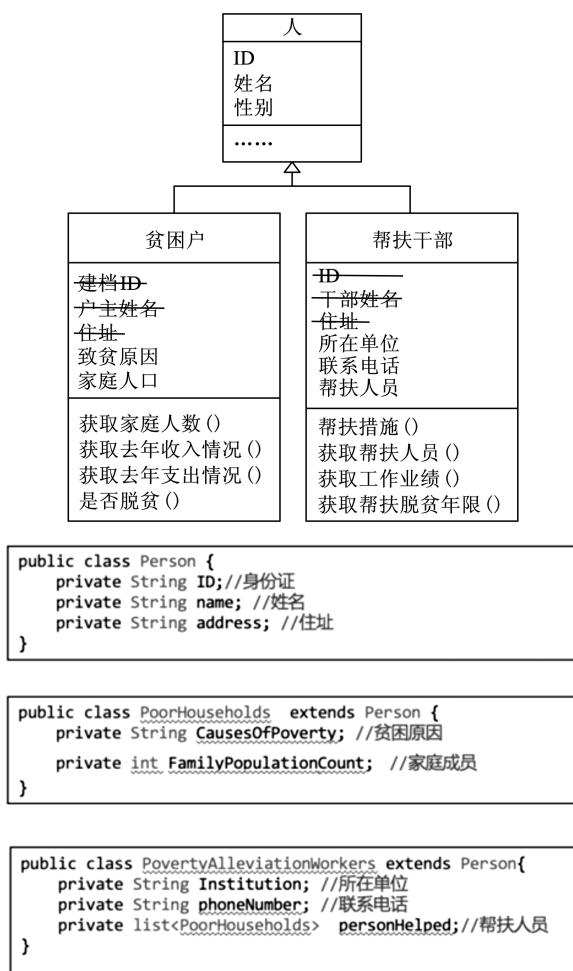


图 8.52 代码展示

① Person 父类。

(3) 引导学生透过现象看本质，充分利用不同事物之间的联系，使结果更加精简化，培养学生的独立思考、分析、归纳和总结能力，同时锻炼学生实际动手编程能力；并为讲解 Java 继承做铺垫。



```
public class Person {  
    private String ID; //身份证  
    private String name; //姓名  
    private String address; //住址  
    public Person(String ID, String name, String address) {  
        this.ID = ID;  
        this.name = name;  
        this.address = address;  
    }  
    //扶贫方法  
    public void PA(){  
        System.out.println(name+"正在扶贫");  
    }  
}
```

(4) 加深学生对继承地理解，力图挖掘事物现象本质，本着实事求是的理念与态度。

② PovertyAlleviationWorkers 类。

```
public class PovertyAlleviationWorkers extends Person{  
    private String phoneNumber;  
    private String personHelped;  
    public PovertyAlleviationWorkers (String ID, String name,  
String address, String phoneNumber, String personHelped) {  
        //向父类传递数值  
        super(ID, name, address);  
        this.phoneNumber = phoneNumber;  
        this.personHelped = personHelped;  
    }  
    public static void main(String[] args) {  
        PovertyAlleviationWorkers paw = new  
PovertyAlleviationWorkers("3565565", "李四", "陕西", "13445566", "张  
三");  
        System.out.println(paw.getID());  
        System.out.println(paw.getName());  
        System.out.println(paw.getAddress());  
        System.out.println(paw.phoneNumber);  
        System.out.println(paw.personHelped);  
        paw.PA();  
    }  
}
```

(5) 继续加深 Java 类的继承的概念，使学生充分理解 Java 继承，提升效率，培养学生的独立思考、分析、归纳和总结能力，同时锻炼学生实际动手编程能力。

③ 运行结果如下。



3565565
李四
陕西
13445566
张三
李四正在扶贫

(3) 教师导入其他相关案例(见图 8.53)。

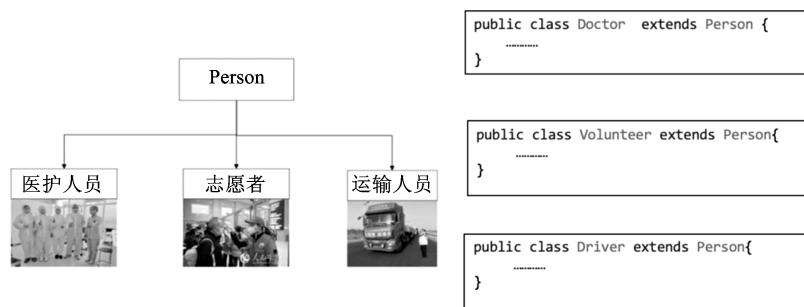


图 8.53 案例展示图

(4) 教师解释代码作用，询问学生是否有疑惑。

若学生提出疑惑，教师对疑惑的地方再次进行解释。

教师讲述教学内容：在课堂导入结构体与类的对比中提到 Java 类可以互相嵌套，这个嵌套关系便是 Java 中的聚合概念。

聚合：它是两个类之间的一种关系，类似于关联，但它是一种定向关联，严格意义上的单向关联，表示一个 has-a 关系。简单来讲就是将一个类当作另一个类的属性(见图 8.54)。

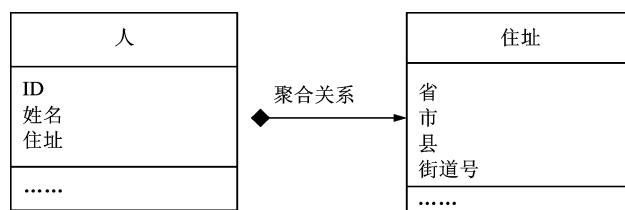


图 8.54 聚合关系

代码展示。

① 构建 Address 类。



```
class Address {  
    String province; //省  
    String city; //市  
    String village; //村  
    int streetNum; //街道号  
    public Address (String province, String city, String village, int  
streetNum) {  
        this.province=province;  
        this.city=city;  
        this.village=village;  
        this.streetNum=streetNum;  
    }  
    @Override  
    //将地址连接起来  
    public String toString() {  
        return "Address:"+province+city+village+streetNum;  
    }  
}
```

②构建 PoorHouseholds 类，以 Address 类作为属性。

```
public class PoorHouseholds {  
    private String ID; //身份证  
    private String name; //姓名  
    private Address address;  
    //住址 Address 类作为 PoorHouseholds 类的元素  
    private String CausesOfPoverty; //贫困原因  
    //家庭成员  
    private int FamilyPopulation;  
}
```

③在 PoorHouseholds 类主程序中实现赋值。

```
public static void main(String[] args) {  
    Address ad=new Address("湖南", "湖市", "南村", 13);  
    PoorHouseholds poorHouseholds = new PoorHouseholds ("4433223", "张  
三", ad, "患病", 4);  
    System.out.println(poorHouseholds.ID);  
    System.out.println(poorHouseholds.name);  
    System.out.println(poorHouseholds.address.toString());  
    System.out.println(poorHouseholds.CausesOfPoverty);  
    System.out.println(poorHouseholds.FamilyPopulation);  
}
```

④运行结果如下。



```

4433223
张三
Address: 湖南湖市南村 13
患病
4

```

Java 继承与聚合概念的区别及联系：继承与聚合相似之处都是对数据的复用，在一定程度上减少程序上的开发。但继承与聚合在语法和概念上又有本质的区别，继承表示的是一种父子关系，通过关键字 extends 明确表示；聚合表示的是一种拥有关系，即 has-a 的关系，一个类作为另一个类的属性(见图 8.55)。

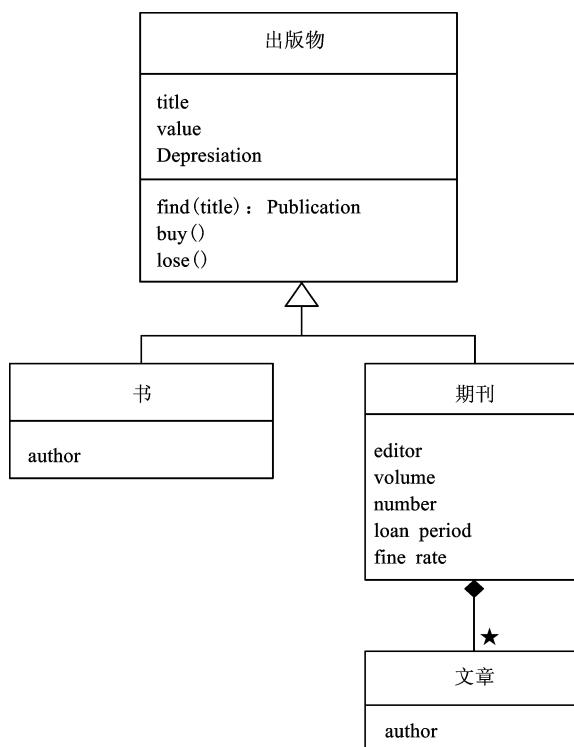


图 8.55 “继承”与“聚合”相结合

◆ Java 访问控制

(1) 教师讲述教学内容。

在 Java 语言中也有类似的变量，但是变量的作用范围并不只是由变量的位置决定，还取决于变量的修饰符。

修饰符的作用是用来限制对类、方法、属性的访问。

Java 有四个修饰符：private、default、protected 和 public。

Java 修饰符的作用范围如表 8.3 所示。

【设计意图】

(1) 使学生深入理解 Java 修饰符的概念。



表 8.3 修饰符的作用范围

修饰符	同一个类中	同一个包中	不同包中	全局
private	Yes			
default	Yes	Yes		
protected	Yes	Yes	Yes	
public	Yes	Yes	Yes	Yes

(2) 让学生能够理解修饰符在 Java 程序中的作用，探讨 Java 修饰符的实际用途，并培养学生对隐私数据的保护意识。

注意事项：

- default、public 也可以用来修饰一个类，但是 protected 不能修饰一个类。
- 访问控制的限制程度从高到低如图 8.56 所示。注意，Default 不是一个修饰符，它只是用来表示一种不加任何修饰符时的状态。

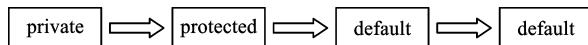


图 8.56 访问控制符限制状态

(2) 教师展示修饰符的作用。

- ① 在 Person 父类中，因为 ID、name、address 设置为 private，所以 PovertyAlleviationWorkers 类需要通过 get 方法来获取。

```

//Person 类
public class Person {
    private String ID; //身份证
    private String name; //姓名
    private String address; //住址
}
//PovertyAlleviationWorkers 类
public class PovertyAlleviationWorkers extends Person{
    public static void main(String[] args) {
        System.out.println(paw.getID());
        System.out.println(paw.getName());
        System.out.println(paw.getAddress());
    }
}
  
```

- ② 将 name、address 的修饰符设为 protected、public。

```

//Person 类
public class Person {
    //身份证
    private String ID;
    //姓名
    protected String name;
    //住址
    public String address;
}
  
```



```
//PovertyAlleviationWorkers 类
public class PovertyAlleviationWorkers extends Person{
    public static void main(String[] args) {
        System.out.println(paw.getID());
        System.out.println(paw.getName());
        System.out.println(paw.getAddress());
    }
}
```

③程序展示。可以看到除了 ID 以外其他属性都可以访问(见图 8.57)。

```
System.out.println(paw.getID());
System.out.println(paw.ID);
System.out.println(paw.getName());
System.out.println(paw.name);
System.out.println(paw.getAddress());
System.out.println(paw.address);
```

图 8.57 Java 访问控制程序

◆ 延伸阅读

教师建议学生在课后进一步去了解：

- 在子类中实现方法名相同、参数名不同的方法属于什么。
- 了解更多类与类之间的关系。

聚合关系：<https://beginnersbook.com/2013/05/aggregation/>

【设计意图】

延伸阅读主要培养学生科研性学习习惯。

◆ 课堂小结

教师对本堂课教学内容进行小结，并引出下堂课内容——Java 修饰符。

学生认真听讲，并积极思考。

【设计意图】

对本堂课内容进行小结，梳理知识脉络。

◆ 布置作业、实践以及下一堂课的课前探究内容

教师布置课后作业与巩固练习题，准备下课。

【具体教学内容】

(1) 布置作业。

请学生谈谈 Java 聚合与继承的异同。

(2) 布置实践。

请学生编写一个具有继承、聚合的 Java 程序。

(3) 布置下一堂课课前探究内容。

学生探究 Java abstract 修饰符的使用规则。



►【板书设计】

类的复用板书设计如图 8.58 所示。



图 8.58 类的复用板书设计

►【形成性评价】

本堂课形成性评价主要采用课堂提问和选择题方式来分析学生是否主动参与课堂活动，做出学生对学习内容的及时评价。

(1) 教学重点考察。

下面哪些访问控制符能在不同的子类中进行访问()。

- A. private B. default C. protected D. public

(2) 教学难点考察。

在 Java 中，下列关于覆盖正确的说法是()。

A. 当对父类的方法进行覆盖时，子类中的方法必须和父类中对应的方法具有相同的方法名称、输入参数和返回值

B. 子类中的覆盖方法不能使用比父类中被覆盖方法更严格的访问权限

C. 方法覆盖不局限于子类和父类之间

D. 父类的静态方法不能被子类覆盖为非静态方法

(3) 思政考察点。

①学生有没有形成注重隐私的意识？

②学生有没有形成在 Java 语言中减少重复、合理使用的“复用”意识？

③针对现在不断发展的技术，激发学生的创新精神。

④学生有没有提高做事的效率，将复杂予以简化？



教 学 反 思

Java 编程概述是“Java 语言程序设计”课程中的面向对象特征教学内容，该内容教学的主要目的是让学生了解 Java 继承的概念、Java 访问空字符，激发学生对 Java 语言学习的浓厚兴趣。通过对学生的认知水平、能力状态、学习需要等分析，学生已经了解 Java 面向对象基本思想，掌握 Java 的基本语法，但是对于面向对象的用法并没有涉及。学生在掌握 Java 的基本语法，熟悉了面向对象的基础知识后，对面向对象在程序中的应用会产生较高的兴趣。基于上述内容，在本堂课内容的讲授中进行了如下设计：

(1) 首先回顾 Java 面向对象知识，对比 C 语言结构体和 Java 类，加深学生对类的知识的印象。一方面激发学生的兴趣，使其尽快进入课堂角色，另一方面为本堂课的内容讲授奠定基础。

(2) 讲解 Java 继承与聚合的知识后，用图片展示两者的差别，使学生深入理解类、继承、聚合的概念。同时，在知识讲述过程中联系时政，理论与实践相结合，一方面了解 Java 知识，另一方面了解国家时事，体会个人在国家中的作用。在讲解 Java 访问控制符时，体会数据在程序中的运行，思考如何通过限制对数据的访问来提高程序的安全，培养学生对数据隐私的保护意识。通过讲解 Java 继承知识，让学生能合理改变子类方法，以减少程序的开发，在编程中合理使用“复用”主义，提高编程效率。

(3) 作为 Java 语言程序知识讲解课，引导学生对本堂课的知识点进行更进一步的探讨，增加 Java 知识面的深度与广度，培养研究性学习习惯以增强创新意识。对本堂课内容进行小结，梳理知识脉络，帮助学生认识到 Java 语言特点是顺应时代的产物，强调学生争当时代弄潮儿的重要性，争取全面发展，为国家的科技事业作出贡献。同时，承前启后，引出下一堂课内容——Java 修饰符；布置课后习题，提高学生的知识点的运用能力。

另外，根据教学过程的形成性评价教学内容重点、难点目标以及思政目标进行达成度分析，根据统计结果，对今后教学做改进。针对本届学生，对今后教学做改进：若大部分学生对 Java 类、继承、聚合仍有疑惑，在今后的教学中还会继续结合本堂课的内容和代码继续讲解知识点，让同学更好地掌握。针对下一届学生，对本堂课教学做改进：本堂课将继续结合国家最新的政策向学生讲解知识，让学生在学习的过程中同时了解到国家的实时政策，让学生达到最好的状态。