JAVA程序设计

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

目 录 CONTENTS

1 学习指南

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

学习指南

本章首先介绍java面向对象程序设计的基本概念，这些概念主要包括类的定义，以及类的基本特性，包括封装、继承、多态；然后通过引入抽象类和接口技术实现了类的多态体现，通过使用包对多个类进行结构组织；最后介绍了常用的几种内部类。要求读者通过本章的学习，彻底掌握类的基本构建过程，了解类的组织结构，掌握类的特性和面向对象的编程思想。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

2 难点重点

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

难点重点

- 面向对象技术的基本概念
- 抽象与接口
- 类的定义
- 包
- set访问器和get访问器
- 访问控制符号的使用
- 继承性
- 关键字static的用法
- 多态性
- 内部类
- 关键字final的用法

3 知识内容

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

1. 面向对象技术的基本概念

1.1 面向过程与面向对象

面向过程与面向对象是计算机软件开发领域最基本的两种处理问题方法。因其对处理实际问题时的思考方式不同，所以是常被用来作为计算机软件设计的两种代表思路。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

1. 面向对象技术的基本概念

1.1 面向过程与面向对象

（1）面向过程：

面向过程其实就是按照事情发展的先后顺序进行编程。

举个例子，假设有一个需求，要求在计算机内用编程方法模拟一个人在房间内的行走过程，首先需要建立的过程模型如下：程序开始 - 抬起左脚 - 向前迈 - 放下左脚 - 抬起右脚 - 向前迈 - 超过左脚所在位置 - 放下右脚 - 继续走吗？ - 是 - 转到抬起左脚处 - 否 - 结束，以上为行走过程的细致描述。

1. 面向对象技术的基本概念

1.1 面向过程与面向对象

（2）面向对象：

面向对象是事先描述对象。还是上面的例子。如果采用面向对象的方法来处理，则将行走过程中涉及到的每个对象都单独加以描述。第一个对象是人，对于人来说设定好他的基本参数：腿长、迈距作为属性设定好；脚抬起来了（发生了一个事件），就执行向前迈的动作（定义为方法），这样定义好了之后人就有了实实在在的结构和数据。第二个对象是房间，对于房间来说设定好长、宽、高三个属性，然后再设定人在其中行走的起点和终点的位置座标值，那么房间结构的建立和数据的填充也完成了。

1. 面向对象技术的基本概念

1.2 面向对象软件开发方法

面向对象软件开发方法是一种把面向对象的思想应用于软件开发过程中，指导开发活动的系统方法，简称OO (Object-Oriented)方法，是建立在"对象"概念基础上的方法学。对象是由数据和容许的操作组成的封装体，与客观实体有直接对应关系，一个对象类定义了具有相似性质的一组对象。而每继承性是对具有层次关系的类的属性和操作进行共享的一种方式。所谓面向对象就是基于对象概念，以对象为中心，以类和继承为构造机制，来认识、理解、刻画客观世界和设计、构建相应的软件系统。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

1. 面向对象技术的基本概念

面向对象的基本概念有：

① 对象

对象是要研究的任何事物。世界上所有事物都可以被看作对象，它不仅能表示有形的实体，也能表示无形的（抽象的）规则、计划或事件。

② 类

类是对象的模板。即类是对一组有相同数据和相同操作的对象的定义，一个类所包含的方法和数据描述一组对象的共同行为和属性。类是在对象之上的抽象，对象则是类的具体化，是类的实例。

1. 面向对象技术的基本概念

面向对象的基本概念有：

③ 继承

继承性（Inheritance）是指，在某种情况下，一个类会有"子类"。子类比原本的类（称为父类）要更加具体化，子类会继承父类的属性和行为，同时也可包含它们自己的属性和行为，这意味着程序员只需要将相同的代码写一次。

④ 封装性

封装是面向对象方法的一个重要原则。它有两个含义：一是指把对象的属性和行为看成为一个密不可分的整体，将这两者"封装"在一个不可分割的独立单位（即对象）中。另一层含义是指"信息隐蔽"，把不需要让外界知道的信息隐藏起来。

1. 面向对象技术的基本概念

面向对象的基本概念有：

⑤ 多态

多态（Polymorphism）是指由继承而产生的相关的不同的类，其对象对同一消息会做出不同的响应。多态描述的是同一个行为方法可以根据执行行为方法的对象的不同而产生不同的行为方式。JAVA通过方法重载、成员覆盖和接口等概念来实现多态。

1. 面向对象技术的基本概念

1.3 面向对象分析

面向对象就是主张从客观世界固有的事物出发来构造系统，提倡用人类在现实生活中常用的思维方法来认识、理解和描述客观事物，强调最终建立的系统能够映射问题域，也就是说，系统中的对象以及对象之间的关系能够如实地反映问题域中固有事物及其关系。面向对象的方法恰好可以使得程序设计按照人们通常的思维方式来建立问题域的模型，设计出尽可能自然地表现求解方法的软件。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

2. 类的定义

2.1 类的定义

Java程序的基本单位是类，建立类之后，就可用它来建立许多你需要的对象。Java把每一个可执行的成分都变成类。

2. 类的定义

2.2 类的定义格式

一个类的定义包含两部分内容：类的声明和类体。一个完整类的定义形式如下：

```
[访问权限] [final] [static] class 类名 [extends 父类名]
[implements 接口名]
{
        定义成员变量
        定义成员方法
}
```

2. 类的定义

2.2 类的定义格式

说明：

在类的声明部分，关键定class后跟类名称和类体。类名称用标识符表示，类体用一对大括号括起来。类体中包含类的成员变量和成员方法，也可以仅包含其中一种。对类的成员可以设定访问权限，以限定其他对象对它的访问，访问权限可以有private、protected、public和包访问权限。上述定义中用[ ]括起来的部分为可选项，可以不写出来。

2. 类的定义

2.2 类的定义格式

[例4-1] 定义一个 CubeBox类。它包括四个成员变量，String类型的盒子颜色和double类型的盒子宽度、高度及深度。它包括两个成员方法：计算盒子体积的方法getVolume(),返回一个double类型的计算结果；显示盒子信息的方法showBoxMessage()，无返回值。

程序源码：

CubeBox.java

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

2. 类的定义

2.2 类的定义格式

[例4-2] 类Circle定义了一个类的属性和行为，成员变量x、y和radius表示圆的坐标和半径。方法drawCircle ( )和eraseCircle ( )表示画圆和擦圆。其中与类名相同的两个方法为构造方法，可以用来创建Circle类的对象。

程序源码：

Circle.java

2. 类的定义

2.2 类的定义格式

成员变量表示一个类的属性，其定义格式如下：

[访问权限] [final] [static] 类型 变量名

成员变量的类型和名字是必须要定义的，其它均为可选项,可以省略。

2. 类的定义

2.2 类的定义格式

上述定义说明如下：

（1）类型 用于说明成员变量的类型，可以是基本的数据类型，也可以是其它复合类型。

（2）访问权限表示对成员变量的访问控制，有四种类型：public protected package private。

（3）static：说明该变量为类变量，默认时为普通的在对象中使用的变量。

（4）final: 表示值不能被改变的量，它用来表示常量。

2. 类的定义

2.2 类的定义格式

[例4-3] 实例定义类中的成员变量并为其赋值。

程序源码：

BianLiang_test.java

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

2. 类的定义

2.2 类的定义格式

方法表示类所具有的功能或行为，它是一段用来完成某些操作的程序片段。方法就是语言基础中的函数。方法的定义包括两个部分：方法的声明和方法体。

```
[访问权限] [final] [static] [abstract] 返回类型 方法名（{参数列表}）
{
   局部变量的声明;
   合法的JAVA表达式语句;
[return 返回值] ;
}
```

2. 类的定义

2.2 类的定义格式

说明：

访问权限与成员变量的类似，也有四个访问级别：public protected package private。默认时为package。

final方法不能被子类重新定义，abstract方法为抽象方法，抽象方法只有方法声明，没有方法体，在后面章节中会对final方法和abstract方法做详细介绍。

[例4-4] 定义一个求面积的程序，可以直接运行，本例用到了类中方法的定义。

程序源码：

SquareDouble.java

2. 类的定义

2.2 类的定义格式

JAVA语言允许在一个类中定义几个同名的方法，但要求这些方法具有不同的参数集合，即方法参数的个数、类型和次序要不同。这种做法称为方法重载。当调用一个重载的方法时，JAVA编译器可根据方法参数的个数、类型、次序的不同，来调用正确的方法。

[例4-5] 定义两个同名方法square( )，区别在于方法参数不同，一个是整型，另一个为双精度类型。

程序源码：

Method_overload.java

2. 类的定义

2.2 类的定义格式

[例4-6] 定义五个重载的方法area用于计算不同类型数据的面积值。虽然名字相同，但是5个area函数的参数类型不同，因此可以通过一个函数完成多种类型数值的计算。

程序源码：

Area.java

2. 类的定义

2.3 对象的定义与使用

（1）对象的定义

定义了用户自定义的类之后，就可以用这个类去定义类对象了。面向对象的编程就是生成多个对象，这些对象通过方法传递来进行数据交流，最终完成复杂的任务。

对象的生成通常包括声明对象变量、实例化对象和初始化对象三个步骤。

格式如下：

类名 对象名 = new 类名([参数列表])；

举例如下，我们用一个Child类去创建对象：

```
Child chi = new Child("小明",21,"1987-1-3");
```

2. 类的定义

2.3 对象的定义与使用

（2）对象的具体使用方法

创建好对象以后，就可以使用对象，让其完成一些功能。

例如从对象获取一些信息、改变对象的属性、让对象完成某些操作等。这些功能可以通过调用访问对象的成员变量或调用对象的方法来实现。

2. 类的定义

2.3 对象的定义与使用

[例4-7] 定义一个CubeBox类，构造函数完成赋值运算，getVolume()函数获得长宽高的乘积值，showBoxMessage()函数输出各个变量的值。

程序源码：

CubeBox.java

2. 类的定义

2.3 对象的定义与使用

构造方法在类中是一种特殊的方法，专门用来创建对象，并完成对象的初始化工作，这就是构造的功能。构造方法的特点如下：

（1）构造方法的名称与所在类的名称相同。

（2）构造方法没有返回值，在方法声明部分不能写返回类型，也不能写void。

（3）构造方法只能用new运算符调用，用户不能直接调用构造方法。

（4）每个类中至少有一个构造方法。

（5）定义类时如果没有定义构造方法，运行时系统会为该类自动定义默认的构造方法，称为默认构造方法。默认构造方法没有任何参数，并且方法体为空，它不做任何事情。

2. 类的定义

2.3 对象的定义与使用

[例4-8] 构造函数的具体使用。在类Gouzao中，有两个重载的构造方法。

程序源码：

Gouzao.java

[例4-9] Child类中在构造方法使用this关键字完成重载。

程序源码：

Child.java

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

3. set访问器和get访问器

用类创建对象后，对属性值的获取和设置非常频繁，这就需要采用一些方法来统一属性的存取和设置过程。为了使程序规范，易于理解，创建了两种被称为构造器的结构：一种是访问器另一种是set访问器，两种访问器其实就是类中的getter方法和setter方法，分别用于获取属性值和设置属性值。

3. set访问器和get访问器

[例4-10] get和set构造器的具体用法。在本例中类Person内部有四个set构造器和四个get构造器。

程序源码：

Person.java

4. 继承性

继承性是子类自动共享父类属性变量和方法的机制，这是类之间的一种关系。在定义和实现一个类的时候，可以在一个已经存在的类的基础之上来进行。在软件开发中，类的继承性使所建立的软件具有开放性、可扩充性，它简化了对象、类的创建工作量，增加了代码的可重用性。

4. 继承性

4.1子类的创建

类继承也称为类派生，指一个类可以继承其它类的成员，包括成员变量和成员方法。被继承的类称为父类或超类，继承后产生的类称为派生类或子类。在JAVA语言中都是用自己编写的类去继承已有的类。所有类都是通过直接或间接地继承java.lang.Object类得到的。在继承的规模性方面，JAVA类只允许单继承，即只允许每个类有一个父类，不允许有多个父类，但一个类可以有多个子类。类继承并不改变成员的访问权限，父类中的成员为公有的public、私有的private或被保护的protected，其子类成员访问权限仍为公有的、私有的或被保护的。

4. 继承性

4.1子类的创建

类继承用关键字extends实现，格式为：

```
[修饰符] subClassName extends superClassName
{
  类体;
}
```

subClassName为子类名，superClassName为父类名，extends表示继承。在类的定义过程中，如果没有使用extends关键字，则默认该类继承于Object类。子类继承父类，但父类中有些成员变量和方法子类不能继承。

4. 继承性

4.1子类的创建

子类继承父类的规则归纳如下：

(1)子类能够继承父类中public和protected的成员；

(2)子类不能继承父类中的private私有成员；

(3)子类能够继承父类中没有访问控制符的成员，只要子类和父类在同一个包中；

(4)子类不能继承父类中的构造方法；

4. 继承性

4.1子类的创建

[例4-11] 继承的实现实例

程序源码：

B.java（父类）和A.java（子类）

4. 继承性

4.1子类的创建

[例4-12] 定义小学生类StuChild继承小孩类Child，包含了继承中的添加、隐藏、重写功能。

程序源码：

Child.java（父类）和StuChild.java（子类）

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

4. 继承性

4.2 this和super关键字

创建子类对象时，使用子类的构造函数对其初始化，不但要对自身的成员变量赋初值，还要对继承的父类的成员变量赋初值。为了区分开父与子之间的关系，规定用this关键字表示当前类的构造函数名，用super关键字表示父类的构造函数名。对于super关键字来说，用于引用父类，如果子类隐藏了父类的成员变量或重写了父类的方法，有时还需要使用父类中被隐藏的成员变量或被重写的方法，这就需要借助super关键字来实现对父类成员的访问。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

4. 继承性

4.2 this和super关键字

[例4-13] 建立类A和类B，其中B是A的子类。在主类中对B进行初始化时同时可以完成对父类A中成员的初始化工作。

程序源码：

B.java（子类）和A.java（父类）

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

4. 继承性

4.2 this和super关键字

① this

this的用法在java中大体可以分为3种：

- 普通的直接引用，this相当于是指向当前对象本身。
- 形参与成员名字重名，用this来区分。
- 引用本类的构造函数。

4. 继承性

4.2 this和super关键字

① this

[例4-14] 建立类Person，在GetAge(int age)方法中，age是GetAge成员方法的形参，this.age是Person类的成员变量。

程序源码：

Person.java

4. 继承性

4.2 this和super关键字

② Super

super也有三种用法：

- 普通的直接引用
- 子类中的成员变量或方法与父类中的成员变量或方法同名时可以使用super引用父类的成员。

4. 继承性

4.2 this和super关键字

② Super

[例4-15] 建立父类Country和其子类City，在子类City中使用super.value()调用父类的value()方法，使用super.name调用父类的name属性。

程序源码：

Country.java和City.java

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

4. 继承性

4.2 this和super关键字

② Super

super也有三种用法：

◆ 引用构造函数

super（参数）：调用父类中的某一个构造函数。

this（参数）：调用本类中另一种形式的构造函数。

4. 继承性

4.2 this和super关键字

② Super

[例4-16]构造父类Man和子类Chinese， 用super和this分别调用父类的构造方法和本类中其他形式的构造方法。

程序源码：

Man.java和Chinese.java

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

4. 继承性

4.3继承与组合的区别

- 什么是继承

  A继承B，说明A是B的一种，并且B的所有行为对A都有意义

- 什么是组合

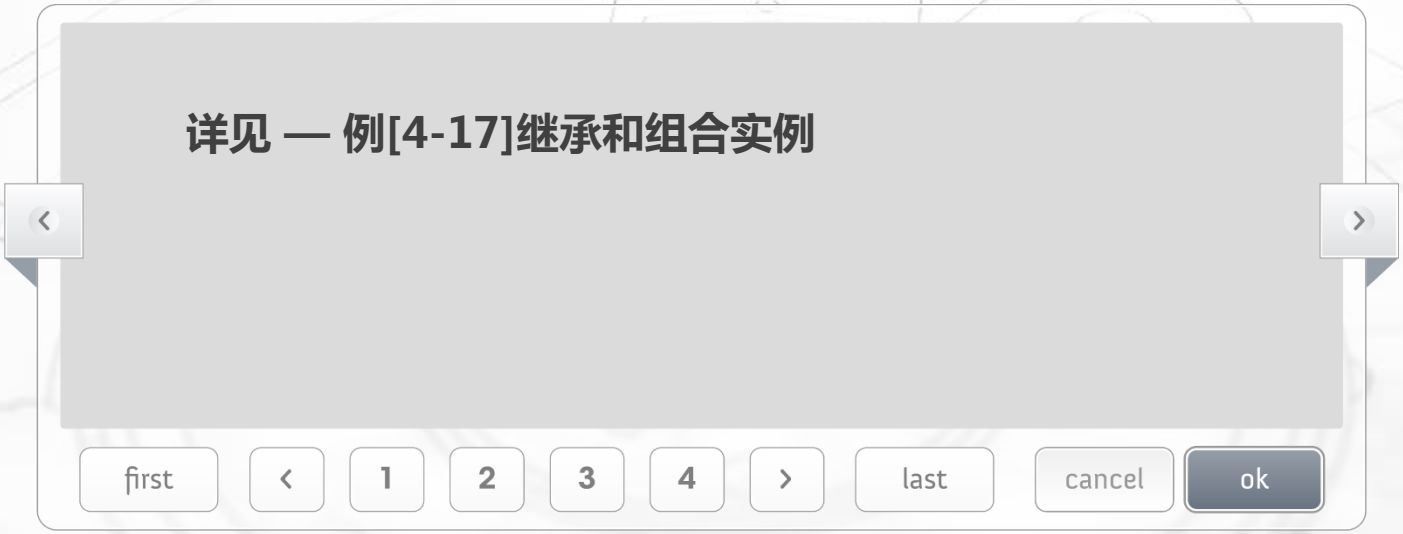
  若在逻辑上A是B的”一部分”(a part of)，则不允许B从A派生，而是要用A和其它东西组合出B。

- 继承的优点和缺点
- 组合的优点和缺点
- 两者的选择
- 法则

4. 继承性

4.3继承与组合的区别

详见 — 例[4-17]继承和组合实例



4. 继承性

4.4 多态性

面向对象编程有三大特性：封装、继承、多态。

封装隐藏了类的内部实现机制，可以在不影响使用的情况下改变类的内部结构，同时也保护了数据。对外界而已它的内部细节是隐藏的，暴露给外界的只是它的访问方法。

继承是为了重用父类代码。两个类若存在IS-A的关系就可以使用继承，同时继承也为实现多态做了铺垫。

所谓多态就是指程序中定义的引用变量所指向的具体类型和通过该引用变量发出的方法调用在编程时并不确定，而是在程序运行期间才确定，

5. 多态性

5.1 方法的覆盖

基于继承的实现机制主要表现在父类和继承该父类的一个或多个子类对某些方法的重写，多个子类对同一方法的重写可以表现出不同的行为。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

5. 多态性

5.1 方法的覆盖

[例4-18]方法的覆盖实例。

程序源码：

Wine.java、JNC.java、JGJ.java

5. 多态性

5.2 变量的隐藏

父类和子类拥有相同名字的属性或者方法时，父类的同名的属性或者方法形式上不见了，实际是还是存在的。隐藏是对于静态方法和成员变量而言的。

（1）当发生隐藏的时候，声明类型是什么类，就调用对应类的属性或者方法，而不会发生动态绑定。

（2）属性只能被隐藏，不能被覆盖。

（3）变量可以交叉隐藏：子类实例变量/静态变量可以隐藏父类的实例/静态变量

5. 多态性

5.2 变量的隐藏

[例4-19]

程序源码：

Shape.java、Circle .java

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

6. 关键字final的用法

关键字final用于修饰类、方法和变量。用final修饰的类不能被继承，即final类没有子类。用final修饰的方法不能被改写，即子类的方法名不能与父类的final方法同名。用final修饰的变量定义时必须同时初始化且在程序中值不能改变，即final变量是常量。

6. 关键字final的用法

[例4-20] final在程序中的具体用法。类A是最终类但它有一个父类是B。

程序源码：

A.java、B.java

7. 抽象与接口

7.1 抽象与接口

抽象类与接口是java语言中对抽象概念进行定义的两种机制，正是由于他们的存在才赋予java强大的面向对象的能力。他们两者之间对抽象概念的支持有很大的相似，甚至可以互换，但是也有区别。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

7. 抽象与接口

7.2 抽象方法与抽象类

abstract修饰符可以修饰类和方法。用abstract修饰的方法称为抽象方法。抽象方法只有方法的返回值类型、方法的名称和方法的参数，没有具体的方法执行体。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

7. 抽象与接口

7.2 抽象方法与抽象类

用abstract修饰的类称为抽象的类，在使用抽象类时需要注意几点：

(1)抽象类不能被实例化，实例化的工作应该交由它的子类来完成，它只需要有一个引用即可。

(2)抽象方法必须由子类来进行重写。

(3)只要包含一个抽象方法的抽象类，该方法必须要定义成抽象类，不管是否还包含有其他方法。

7. 抽象与接口

7.2 抽象方法与抽象类

用abstract修饰的类称为抽象的类，在使用抽象类时需要注意几点：

(4)抽象类中可以包含具体的方法，当然也可以不包含抽象方法。

(5)子类中的抽象方法不能与父类的抽象方法同名。

(6)abstract不能与final并列修饰同一个类。

(7)abstract 不能与private、static、final或native并列修饰同一个方法。

7. 抽象与接口

7.2 抽象方法与抽象类

[例4-21]定义抽象类shape，然后分别定义类circle和类square继承类shape，实现其中的抽象方法。

程序源码：

shape.java、circle.java 、square.java

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

7. 抽象与接口

7.2 抽象方法与抽象类

[例4-22]一个类继承了抽象类之后，要将其中的抽象方法实现。

程序源码：

A.java、B.java

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

7. 抽象与接口

7.3 接口的定义

接口是方法的声明和常量的集合。接口是类似于类的一种结构，也可以看作是一种完全没有实现的类，接口的定义格式与类的定义格式相似：

```
[修饰符] interface 接口名称 [extends 接口列表]
{
  方法说明和静态常量；
}
```

7. 抽象与接口

7.4接口的实现

接口把方法的定义和类的层次关系区分开。通过它可以在运行时动态的定位所调用的方法。同时接口中可以实现“多重继承”，且一个类可以实现多个接口，这正是JAVA中多继承机制实现的方面。用一个类实现接口的关键字为implements。

7. 抽象与接口

7.4接口的实现

[例4-23]定义接口的结构。类AirPlane继承了前面创建过的类CubeBox，并且实现了接口flyer。

程序源码：

flyer.java、AirPlane.java

7. 抽象与接口

7.4接口的实现

[例4-24] 创建了接口A，并用类B去实现它，最后在主类中定义类B的对象b，通过b.showf()来查看接口中未实现的showf()方法的执行结果。

程序源码：

A.java、B.java

7. 抽象与接口

7.5 接口示例

（1）接口的继承与组合

[例4-25] 具体实现接口的继承与组合。

程序源码：

A.java、B.java、C.java、D.java、E.java

7. 抽象与接口

7.5 接口示例

（2）接口的多态

[例4-26]在接口的实现过程中完成了方法的重写，实现了多态。

程序源码：

A.java、B.java、C.java、D.java、E.java

first ‹ 1 2 3 4 › last cancel ok

8. 包

包是一组相关的类或接口的集合，它提供了对于类的访问权限包装的一种功能。包体现了JAVA面向对象编程特性中的封装机制。JAVA语言的开发人员常将完成相关功能的一组类及接口放在一个包内。

8. 包

8.1 包的定义

创建一个包非常简单，只需要在JAVA源文件最开始的语句之前，包含一条package语句。package后面可以出现嵌套的包结构，表示包含关系，定义格式如下：

package pk1[.pk2[.pk3．．．]]

8. 包

8.1 包的定义

举例：

```
package vecle;
class car
{
 // . . .
}
class truck
{
 // . . .
}
```

8. 包

8.2 包的引用

包中的所有类只有public类能够被外部的类访问，要想从包外使用包中public类，有三种实现方法。

- 使用全名引用包中的public类
- 引进包中的类
- 引进整个包

8. 包

8.2 包的引用

例4-27]包的具体使用

程序源码：

A.java、B.java、C.java

first ‹ 1 2 3 4 › last cancel ok

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

9. 访问控制符号的使用

Java语言采用访问控制修饰符来控制类及类的方法和变量的访问权限，从而向使用者暴露接口，但隐藏实现细节。

访问控制分为四种级别：

（1）public

（2）protected

（3）default

（4）private

10. 关键字static的用法

10.1 static变量

static变量也称作静态变量，静态变量和非静态变量的区别是：静态变量被所有的对象所共享，在内存中只有一个副本，它当且仅当在类初次加载时会被初始化。而非静态变量是对象所拥有的，在创建对象的时候被初始化，存在多个副本，各个对象拥有的副本互不影响。

10. 关键字static的用法

10.1 static变量

[例4-28] static成员变量实例。

程序源码：

StaticExp1.java

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

10. 关键字static的用法

10.2 static方法

static方法一般称作静态方法，由于静态方法不依赖于任何对象就可以进行访问，因此对于静态方法来说，是没有this的，因为它不依附于任何对象。并且由于这个特性，在静态方法中不能访问类的非静态成员变量和非静态成员方法，因为非静态成员方法/变量都是必须依赖具体的对象才能够被调用。

10. 关键字static的用法

10.2 static方法

[例4-29] static成员方法实例。

程序源码：

StaticExp2.java

11. 内部类

内部类是定义在其它类内部的类，主要作用是将一些可能有相关联功能的类组合放在一起。内部类与外部类中的成员变量和方法一样都属于外部类的成员。它可以随意直接访问外部类的所有变量和方法。可以将内部类分为：成员内部类、局部内部类、静态内部类、匿名内部类四类。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

11. 内部类

内部类定义的位置和格式如下：

```
public class OutClass{     //定义外部类；
  //外部类的成员变量和方法的定义；
class InnerClass{      //定义内部类；
//内部类的成员变量和方法的定义；
  }
}
```

11. 内部类

11.1成员内部类

最基本的内部类为成员内部类。将内部类作为外部类的一个成员而存在。

11. 内部类

11.1成员内部类

[例4-30] 通过在类OutClass内部建立类InnerClass来实内部和外部之间的调用。

程序源码：

OutClass.java

11. 内部类

11.2 静态内部类

内部类也可以定义为static类型。一个static内部类不依存于某个具体的外部类。所以创建一个static内部类的对象时，不再需要一个外部类对象。与方法内的内部类一样，static内部类不能直接引用外部类中的变量和方法，只能通过一个对象来使用它们。

11. 内部类

11.2 静态内部类

[例4-31] 在外部类内建立一个静态内部类，通过外部类名来创建内部类对象。

程序源码：

Outer4.java、InnerMain.java

first ‹ 1 2 3 4 › last cancel ok

11. 内部类

11.3 局部内部类

在外部类的方法中可以定义内部类。这个内部类可以访问外部类的成员变量但不可以访问所在方法内部的局部变量，除非是用final修饰符修饰的局部量。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

11. 内部类

11.3 局部内部类

[例4-32] 在外部类的方法内定义内部类，并用内部类使用方法中的final成员。

程序源码：

Outer3.java

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

11. 内部类

11.4 匿名内部类

JAVA的匿名内部类的语法规则看上去有些古怪，当你需要创建一个类的对象而且不用它的名字时，可以用匿名内部类。使用内部类可以使代码看上去简洁清楚。它的语法规则是这样的：直接用new关键字加上类名( )的形式即可。例：

```
new interfacename(){      }; 或 new
superclassname(){      };
```

11. 内部类

11.4 匿名内部类

有一点需要注意的是，匿名类由于没有名字，所以它没有构造函数，如果你想要初始化它的成员变量，有下面几种方法：

（1）如果是在一个方法的匿名内部类，可以利用这个方法传进你想要的参数，不过记住，这些参数必须被声明为final。

（2）将匿名内部类改造成有名字的局部内部类，这样它就可以拥有构造函数了。

（3）在这个匿名内部类中使用初始化代码块。

11. 内部类

11.4 匿名内部类

[例4-33] 在类的继承过程中实现匿名内部类的创建和使用。

程序源码：

Mainniming .java

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

4 本章小结

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

本章小结

本章主要介绍类和对象的基本概念、以及封装、继承和多态三大特性，通过抽象类和接口实现类的多态，通过包组织类的结构，加深读者对面向对象的程序设计思想的理解。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社


THANK YOU
中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社