高等学校应用型本科“十二五”规划教材

机电 MECHATRONICS
计算机 COMPUTER
电子 ELECTRONICS

- 融合软件工程思想与数据库编程
- 像程序员一样学习与思考

Java 面向对象程序设计与系统开发

主　编　姚骏屏　何桂兰
副主编　陈俊玲　陈素琼　任姚鹏

西安电子科技大学出版社
http://www.xduph.com

高等学校应用型本科“十二五”规划教材

像程序员一样学习与思考

Java 面向对象程序设计与系统开发

主　编　姚骏屏　何桂兰
副主编　陈俟伶　陈素琼　任姚鹏

西安电子科技大学出版社

内容简介

本书为已经具备了 Java 或者 C 语言语法基础的软件开发入门者编写，力求从面向对象软件的开发能力、单元测试及集成测试能力、基本设计文档与 UML 设计图的阅读和理解能力、编程规范的养成等多方面对读者进行综合培训。

全书分为三大部分，共 14 章：第一部分(1～5 章)以任务驱动让学生逐步形成面向对象开发的基本理念并能阅读 UML 类图，了解面向对象开发的特点，形成一定的面向对象设计思想，能熟练使用 Java 软件开发中的常用类；第二部分(6～13 章)以一个数据库应用项目为引导，使学生在完成项目的过程中学习到软件分层结构、JDBC 数据库应用开发、JUnit 单元测试工具、集成测试原理、UML 用例图及活动图和时序图等核心内容，掌握多线程的使用以及 GUI 图形用户界面的开发等关键技术；第三部分(14 章)为一个技能拓展案例。

本书重视软件工程规范的逐步养成，不断出现编码规范提示和标准 UML 图例，使学生逐渐习惯软件工程中的通用描述方式。

本书是 Java 软件开发系列教材之一，可作为应用型本科院校以及高职高专院校的教材，也可作为 Java 软件开发人员的参考书。

图书在版编目(CIP)数据

Java 面向对象程序设计与系统开发/姚骏屏，何桂兰主编.
—西安：西安电子科技大学出版社，2015.2
高等学校应用型本科“十二五”规划教材
ISBN 978-7-5606-3649-8

Ⅰ. ① J… Ⅱ. ① 姚… ② 何… Ⅲ. ① JAVA 语言—程序设计—高等学校—教材
Ⅳ. ① TP312

中国版本图书馆 CIP 数据核字(2015)第 023675 号

策　　划　李惠萍
责任编辑　李惠萍
出版发行　西安电子科技大学出版社（西安市太白南路 2 号）
电　　话　(029)88242885　88201467　　邮　　编　710071
网　　址　www.xduph.com　　电子邮箱　xdupfxb001@163.com
经　　销　新华书店
印刷单位　陕西大江印务有限公司
版　　次　2015 年 2 月第 1 版　2015 年 2 月第 1 次印刷
开　　本　787 毫米×1092 毫米　1/16　印张 18
字　　数　423 千字
印　　数　1～3000 册
定　　价　32.00 元
ISBN 978 – 7 – 5606 – 3649 – 8 / TP
XDUP 3941001-1

西安电子科技大学出版社
高等学校应用型本科“十二五”规划教材
编审专家委员会名单

主　任：鲍吉龙(宁波工程学院副院长、教授)

副主任：彭　军(重庆科技学院电气与信息工程学院院长、教授)

张国云(湖南理工学院信息与通信工程学院院长、教授)

刘黎明(南阳理工学院软件学院院长、教授)

庞兴华(南阳理工学院机械与汽车工程学院副院长、教授)

计算机大组

组　长：刘黎明(兼)

成　员：(按姓氏笔画排列)

刘克成(南阳理工学院计算机学院院长、教授)

毕如田(山西农业大学资源环境学院副院长、教授)

李富忠(山西农业大学软件学院院长、教授)

向　毅(重庆科技学院电气与信息工程学院院长助理、教授)

张晓民(南阳理工学院软件学院副院长、副教授)

何明星(西华大学计算机与软件工程学院院长、教授)

范剑波(宁波工程学院理学院副院长、教授)

赵润林(山西运城学院计算机科学与技术系副主任、副教授)

雷　亮(重庆科技学院电气与信息工程学院计算机系主任、副教授)

黑新宏(西安理工大学计算机学院副院长、教授)

前　言

1995 年 Sun 公司推出 Java 语言之后，全世界的目光都被这个神奇的语言所吸引，在跨平台开发及互联网应用开发等领域扮演了越来越重要的角色，被公认为功能最强大、最有前途的编程语言之一。

根据教材开发团队多年软件项目开发经验和教研经验，将 Java 软件开发工程师的核心职业能力由低到高划分为四个层次：基本编码与调试能力、面向对象分析及 C/S 软件开发能力、B/S 网站开发能力、主流框架应用开发能力。本书面向第二个层次的能力锻炼，适合具有一定 Java 或者 C 语言语法基础的入门读者。

本书基于“教、学、做一体”的教学模式，根据任务需要对知识点（涉及编程语言、软件工程、数据结构、数据库、软件测试等学科知识）进行整合与精简，体现了理论与实践一体化的教学思想。

全书分为三大部分，共 14 章：第一部分（1～5 章）以任务驱动让学生逐步形成面向对象开发的基本理念并能阅读 UML 类图，了解面向对象开发的特点，形成一定的面向对象设计思想，能熟练使用 Java 软件开发中的常用类；第二部分（6～13 章）以一个数据库应用项目为引导，使学生在完成项目的过程中学习到软件分层结构、JDBC 数据库应用开发、JUnit 单元测试工具、集成测试原理、UML 用例图及活动图和时序图等核心内容，掌握多线程的使用、GUI 图形用户界面的开发等关键技术；第三部分(14 章)作为技能拓展模块，以一个小型图书管理系统为例，进一步帮助学生理解 Java 面向对象程序设计的思想及其应用。

本书重视软件工程规范的逐步养成，与以往的同类型教材不同的是，本书不仅培养学生的程序编制能力，更加注重培养学生的软件开发工程应用能力，其中不断出现编码规范提示和标准 UML 图例，使学生逐渐习惯软件工程中的通用描述方式。书中还安排了单元测试章节，让学生学会常用的 JUnit 测试基本应用，并在项目结束环节讲解了集成测试的相关理论。同时，在细节处理上，本书也针对学生的认知特点进行了特别处理，对关键代码进行了加粗描述，使学生能够更快、更集中精力关注到所学知识内容，找到问题的关键点。

另外，在巩固与提高与单元测试中，为了加强学生的自主学习能力，本书有意识地增加了少量超纲题，书末所附综合测试题为由历年计算机二级考试 Java 部分真题整合而成。

本书是 Java 软件开发系列教材之一，可作为应用型本科院校以及高职高专院校的教材，也可作为 Java 软件开发人员的参考书。

本书由姚骏屏、何桂兰主编，陈俟伶、陈素琼、任姚鹏为副主编。姚骏屏编写 5、6、7、8、10、11 章并负责全书统稿与修订工作；何桂兰完成了全书框架设计，1、2、3、9 章编写，单元测试与综合测试的编制，并完成全书审读与修订工作；陈俟伶主持了全书前期项目引导部分的编写工作；陈素琼编写第 4 章并参与全书修订工作；任姚鹏参与了 9、14 章的编写和全书的校正工作。此外，谭登超编写第 12 章，张正龙参与编写第 1、2 章和第 10 章时序图的设计，张红实完成项目引导部分的代码编制，谢东亮参与编写第 12 章，向守超完成各章节习题的编制，李冀明为软件详细设计提供了技术支持，在此一并致以诚挚的谢意。

由于作者水平有限，疏漏和错误之处在所难免，欢迎广大读者提出宝贵意见。

作　者

2014 年 12 月

目　录

第 1 章　面向对象开发基本概念

在软件开发中有结构化和面向对象两种软件设计和开发方法。在结构化的软件应用系统开发中，一个软件由两个互不相关的部分组成：一个部分描述产品的所有行为；另一个部分描述在产品行为中需要操作的数据。在这样的软件组织结构中，数据和行为被隔离开，而在面向对象的软件应用系统开发中，行为和数据被统一在对象模型中，一个对象同时包括了用于描述该对象所需要的数据，以及这个对象可以进行的操作。这些操作会根据需要访问或者修改描述对象用到的数据。在这样的结构中，一个对象是一个很完善的个体，是数据和操作的统一体，通过调用对象不同的操作就可以完成产品的行为。

Java 语言是一种简单的、跨平台的、纯面向对象(OO，Object Oriented)的、分布式的、解释的、健壮的、安全的、结构的、中立的、可移植的、性能优异的、多线程的、动态的语言。面向对象的精髓在于考虑问题是从现实世界中人类的思维习惯出发，将客观世界看成一个个事物及其关系的组合。本章和接下来的两章将从实际问题出发来阐述面向对象的思维方法，以使读者认识面向对象方法中的基本概念并习惯这种思维模式。

类与对象是面向对象程序设计中最基本且最重要的两个概念，有必要仔细理解和彻底掌握，它们将贯穿全书并将逐步深化。

1.1　类

学习目标

(1) 理解类的概念；

(2) 用面向对象的思维对生活中的事物进行归类；

(3) 能够使用 Java 语言定义类。

任务描述

用 Java 语言描述一个矩形类，其有长和宽两种属性，并且能够计算每一个矩形的面积和周长。

任务分解——理论阐述

1.1.1　什么是类

要解决什么是类的问题，我们必须要理解抽象的概念。抽象是人类解决复杂问题的一种基本方法。把众多的事物进行归纳、分类是人们在认识客观世界时经常采用的思维方法，“物以类聚，人以群分”就是分类的意思，分类所依据的原则是在这个类中的事物满足某

些相同的标准，而这些标准其实就是抽象出来的。所谓抽象，就是忽略事物的非本质特征，从而找出这些事物的共性，并把具有共性的事物划为一类，因此类是一个抽象的概念，它不是一个实物，只是标准(抽象)的总和。比如，你的自行车只是现实世界中许多自行车中的一辆，请说出全世界自行车有什么标准能将其划为一类，你能将自行车和飞机划为一类么？能将自行车和房子划为一类么？

标准是抽象出来的，如果我们抽象出这样一个标准：自行车有两个轮子、有把手、可以骑动……所以有这样标准的物体就被我们人为地归为了一类——自行车类。我们可以按能载人这个标准将自行车和飞机归为一类——交通工具类，我们也可以按是固体这个标准将自行车和房子等归为一类——固体类，只要我们抽象得出标准，就可分出不同的类。

类就是对事物的抽象和归纳，是具有相同标准的事物的集合与抽象。如自行车类，它就是许多真实存在的自行车的抽象。它为属于该类的全部事物提供了统一的抽象描述。

1.1.2 类的基本结构

Java 类中包括属性和方法两个主要部分。

1. 属性

属性是用来描述对象静态特征的一组数据，一般用名词描述。如自行车类中的轮子、把手、重量、颜色等，就是自行车类的属性。

2. 方法

方法是用来描述对象动态特征的一组操作，一般用动词描述。如自行车类中的车的起动、刹车和加速等，就是自行车类的方法。

1.1.3 类的定义

1. 抽象出类

要用 Java 来描述一个类，首先得将该类的属性和方法在头脑中清楚地抽象出来。如我们定义一个人的类，人有眼睛、鼻子、手、姓名、性别、年龄等属性，有能吃饭、能走路、会说话、能思考等行为，在面向对象设计过程中我们用类图来描述一个类，图 1-1 是人类的类图。

图 1-1 人类的类图

【思考】 请画出学生类的类图和教师类的类图。

是不是感觉到在软件开发中，你就是一个个类的创造者，是你创造了一个个的规则进行了分类，在内存中的世界里你就是造物主。

2. 类的创建

Java 中类定义语句的形式为：

```
[类修饰符] class  类名 {  // 类头的定义
    /* 类体的定义*/
    成员变量声明;
```

```
    成员方法定义;
}
```

其中：class 是 Java 语言中定义类时必须使用的关键字，以告诉 Java 这是一个类，用固定的小写格式；“类名”是为这个类取的名，应书写为 Java 语言合法的标识符，由编程人员命名；大括号{ }中是定义类体的地方，包含该类中的数据成员(属性)和成员方法(方法)。在 Java 语言中也允许定义没有任何成员的空类或只有属性或方法二者存一的类。

编码规范要求：

按照类名的书写规范，类名不允许用中文，其首字母应大写，若类名由多个单词组成，则每个单词的首字母均要大写。如：class　TestStudents { …}

前面我们分析了一个人的类，下面将说明在 Java 语言中如何定义一个人的类，其描述形式为：

```
class　人类 {　　// 定义一个人类
    /*人类的属性*/
    眼睛;
    鼻子;
    手;
    姓名;
    年龄;
    …
    /*人类的成员方法定义*/
    吃饭();
    走路();
    说话();
    …
}
```

【例 1-1】　定义一个圆形类，其有一个半径属性，一个求周长的方法。

解：

1) 定义类头

类名用圆形类的英文 round 命名，根据类名编法规范要求，首字母必须大写，如下所示：

```
class Round{ }
```

2) 定义成员变量

我们将在类中定义的方法和变量统称为类的成员。定义在类中但在类方法之外的变量称为成员变量。Java 中完整的语法形式如下：

[修饰符]　成员变量数据类型　成员变量名 [= 初值];

其中：成员变量数据类型可以是任意的 Java 类型；成员变量名必须是合法的 Java 标识符，但是在同一个类中成员变量名不能同名；[修饰符]和[=初值]为可选项。

编码规范提示：

成员变量名中，第一个单词的第一个字母小写，其后的单词第一个字母大写，而且单词之间没有任何分隔符。单词的选用要求能体现该属性的特定含义，成员变量名在一个类中还要保证唯一性。

去掉可选项[修饰符]和[= 初值]就是最简单的成员变量声明。

上面定义了类头，类体是空的，就需要添加类的成员属性表示圆形的状态。代码如下：

```
class Round{
        // 定义成员属性
        double radius;    // 半径
    }
```

从这个例子可以看到，在类中进行成员变量的声明与一般变量的声明形式完全相同。成员变量的类型可以是基本数据类型，也可以是自定义类型。

3) 定义成员方法

定义在类中的方法称为成员方法。在 Java 类体中，方法的定义描述了方法的处理过程及其所需要的参数，并用一个方法名来标示该过程。方法的定义中要声明方法的返回类型、方法名、访问权限以及入口参数。Java 中成员方法的定义形式如下：

```
[修饰符]  方法返回值类型  方法名([形式参数列表]) {
        // 方法体
}
```

其中，方法名返回值类型说明方法执行以后的返回值数据类型，如果某个方法的返回值类型是整形，就定义为 int，如果方法没有返回值，就定义为 void。如果方法的返回类型不为 void,方法体中必须用 return 语句来返回一个方法声明时指定的返回类型。方法名编码规范要求同成员变量一样。

```
class Round {
    // 定义成员属性
    double radius;    // 半径
    // 定义成员方法
    double perimeter(){   // 求圆形的周长
        return 2*3.14*radius;
    }
    double area(){   // 求圆形的面积
        return 3.14*radius*radius;
    }
}
```

任务分解——操作方案

(1) 从功能需求中分析并抽象出类，做出类图；

(2) 按照类图将对应的类的结构构建出来；

(3) 实现出类中的方法。

任务实施

1. 分析抽象出类

类的内部包括属性和方法两种成员。按照属性是名词，而方法是动词的原则，我们不难从本任务中分析抽象出有一个矩形类 Rectangle，它包含两个表示长方形属性的成员变量：长 length 和宽 width，以及两个表示长方形行为的成员方法：perimeter()和 area()方法，分别用于计算长方形的周长和面积。

类图(Class Diagram)是用来描述类内在结构的图，是面向对象系统建模中最常用的图。在 UML 中，类用矩形来表示，并且该矩形被划分为三个部分：名称部分、属性部分和方法部分。其中，名称部分存放类的名称，中间的部分存放类的属性名、属性的类型及值信息，底部的部分存放类的操作、操作的参数表和返回类型信息，如图 1-2 所示。

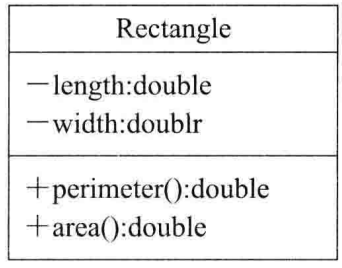

图 1-2　矩形类图

2. 按照类图编制出类

```
class Rectangle{
        // 定义成员变量
         double length;     // 长
         double width;      // 宽
        // 定义成员方法
         double perimeter(){     // 求长方形的周长
         }
         double area(){     // 求长方形的面积
         }
}
```

3. 实现类中的方法

```
double perimeter(){     // 求长方形的周长
         return 2*(length + width);
 }
double area(){     // 求长方形的面积
         return length*width;
 }
```

4. 完整源代码

```
class Rectangle{
        // 定义成员变量
        double length;     // 长
        double width;      // 宽
```

```
    // 定义成员方法
    double perimeter(){     // 求长方形的周长
        return 2*(length + width);
    }
double area(){    // 求长方形的面积
        return length*width;
    }
}
```

【思考】 试分析“学生”这个类，然后用 Java 语言来定义一个学生类 Student，其有姓名、年龄、性别、身高、专业等属性，以及其有回答提问、考试等行为。

1.2 对　　象

学习目标

(1) 明确什么是对象；
(2) 理解对象和类的关系；
(3) 会用 Java 语言创建对象；
(4) 能够访问对象中的成员变量和成员方法。

任务描述

利用 1.1 节设计的矩形类，创建出一个长和宽分别为 20 和 10 的具体的矩形，并计算该矩形的长、宽、周长和面积。

任务分解——理论阐述

1.2.1 什么是对象

面向对象思维认为，客观世界由一个个具体的对象(Object)组成，任何客观的事物和实体都是对象，复杂的对象可以由多个简单的对象组成。所谓对象，是现实世界中的一个实际存在的事物，它可以是有形的，如狗、自行车、房屋等，也可以是无形的，如国家、生产计划等。面向对象思想认为，万物皆为对象，而所有对象均是能划入某些类的，正如我们前一节所述只要你能抽象出标准就能给这些对象分类，那么这些对象就是它所属类的一个具体事物。

在面向对象软件设计中，对象就是用来描述客观事物的一个实体，它将现实中的事物变成了软件世界中的具体东西，而它们的创造者就是你——程序员，当你的软件世界中有了这些东西以后，你就可以让这些东西来完成你作为造物主所交予它们的任务。在你所创造的软件世界中，对象就是构成该世界的一个基本单位，其由一组属性和对这组属性进行操作的一组方法所组成。

1.2.2 对象与类的关系

对象由属性(Attribute)和行为(Action)两部分组成。属性用来描述对象的静态特征，行为用来描述对象的动态特征。

类是具有相同属性和行为的一组对象的总称，它为属于该类的全部对象提供了统一的抽象描述，其内部包括属性和行为两个主要部分，类是对具体对象集合的再抽象。如“人”这个类它是从一个个具体的人的集合中抽象出共同的静态和动态特征形成的一个分类标准，它不是一个具体的事物；而具体的某个人如张三是人这个类的一个实例，是人类的一个对象。因此类与对象是抽象和具体的关系。

类与对象的关系如同一个模具与用这个模具铸造出来的铸件之间的关系。类给出了属于该类的全部对象的抽象定义(标准)，包含了该类对象的所有属性和方法，它是对象的“模具”。而对象则是符合这种定义的一个实体。所以，一个对象又称作类的一个实例(Instance)，可以由一个类制造出多个实例。当然也可以拿设计图纸和产品来比喻类和对象的关系。当你设计了自行车类这个图纸后，你可以按这个图纸创建任意多个自行车对象。当创建了一个类的实例后，系统将为这个对象实例分配内存，此时它就确确实实地存在于你的软件世界中了。类是一类事物共性的反映，而对象是一类事物中的一个，是个性的反映。每个对象都有与其他对象不完全一样的特性。例如，你和我的自行车虽然都是自行车，但二者的颜色、质量等就不可能完全一致。

特别提示：

类是抽象的，而对象是具体的，在我们创造的软件世界中我们一般只让具体的对象去完成实际的任务，正如同我们可以让人类的一个对象(某个具体人)去挑水，而不会让人类去挑水。

1.2.3　对象的生命周期

对象和现实生活中许多事物一样，也有它的产生、发展和消亡的阶段，对象一旦完成了它的任务，也就实现了创建它的目的，它将被释放，对象所占用的资源将被回收以提供给其他对象使用，所以一个对象的生命周期包括创建，使用、释放。

1.2.4　对象的创建

一个类就是一个新的数据类型，与 Java 语言提供的几种基本数据类型一样，该类型可以用来声明、定义该类型的变量。如 int i，就定义了一个变量名为 i 的 int 变量，而 Rectangle rectangle 就定义了一个变量名为 rectangle 的 Rectangle 变量，这个 Rectangle 是我们自己设计的一个类。

创建类的变量称为类的实例化，类的变量也称为类的对象、类的实例。要实际创建一个 Rectangle 对象，使用下面这样一条语句：

```
//创建一个变量名为 rectangle 的 Rectangle 对象
Rectangle  rectangle= new  Rectangle();
```

执行上面这条语句后，rectangle 将成为 Rectangle 的一个实例。它在内存空间中存在了，它具有“物理的”真实性。

由此可见，创建类的对象需用 new 关键字，它的一般形式如下：

```
        类名 对象名;                                  // 声明对象
        对象名 = new  类名([参数列表]);               // 创建对象
或者    类名 对象名 = new  类名([参数列表]);          // 定义类的对象
```

其中："类名"指出了这个对象属于哪个类；"对象名"是给这个对象取一个区别于其他对象的变量名；类名后面的圆括号指定了类的构造方法(将在下一章详细学习)，也就是在构建这个对象的时候必须自动执行的一个方法；new 运算符是 Java 关键字，专门用于调用构造方法来产生一个实体对象。

编码规范提示：

对象变量名的书写规范与一般变量名或方法名的书写一样，以小写字母开头，若对象名是由多个单词组成，则从第二个单词开始每个单词的首字母均要大写。如：Rectangle rectDemo = new Rectangle();

1.2.5 访问对象成员

每次创建一个类的实例时，就创建了一个对象，这个对象包含由类定义的属性和方法。使用点(.)运算符来访问(使用)该对象中的属性和方法，例如，要给 rectangle 的 width 变量赋值为 10，可以使用下面的语句：

```
Rectangle rectangle = new Rectangle();      // 定义类的对象 rect
rect.width = 10;        // 访问对象 rect 的数据成员 width 并赋值 10
```

该语句告诉编译器将包含在 rectangle 对象内的 width 属性赋值为 10。当然也可以如此访问对象中的方法。

访问对象成员的一般形式为：

对象名.成员变量名

对象名.成员方法名(参数列表)

【例 1-2】 利用例 1-1 定义的圆形类 Round，计算半径为 10 的一个具体圆的周长。

```
public static void main (String args[]){
    Round round = new Round();      // 声明并实例化圆形对象 r
    round.radius = 10;        // 访问成员变量并赋值
    double p = round. perimeter();    // 调用成员方法求周长
    System.out.println("半径" + round.radius + "的圆的周长是:" + p);
}
```

结果：

```
半径 10.0 的圆的周长是:62.8
```

同一个类的对象虽然都有相同的属性和方法，但其属性和方法对于每个对象都是独立的。这意味着如果有两个 Rectangle 对象，则每个都有自己的 width 和 length 的副本。其表现是改变一个对象的实例变量是不会影响另一个实例变量的。正如张三和李四都是人类的具体对象，但对张三做的事是不会让李四发生任何变化的。

【例 1-3】 利用例 1-1 中定义的圆形类 Round，生成半径为 20、10 的两个对象，并计算每个对象的周长。

```
public static void main(String args[]){
    Round r1=new Round ();    // 声明并实例化 Round 对象 r1
    Round r2=new Round ();    // 声明并实例化 Round 对象 r2
    // 给每个对象的半径属性赋值
```