15.大数据专业群教师（软件技术方向）1岗位
试讲内容

注意事项：

1.每位考生试讲时间为8分钟；

2.试讲统一采用PPT讲授方式（自备U盘，如因U盘打不开课件，责任自负，U盘不能用考生姓名命名）；

3.试讲的考生在候考室抽签结束后在教案封面填写抽签号提交教案打印件（一式7份）给工作人员。教案不能透露任何个人信息，考生不得穿制服、单位工作服或有明显文字或图案标识的服装参加面试，凡透露个人信息的考生，扣减面试成绩的5%—20%，情节严重的，取消面试成绩。

教学内容：第3章 面向对象程序设计

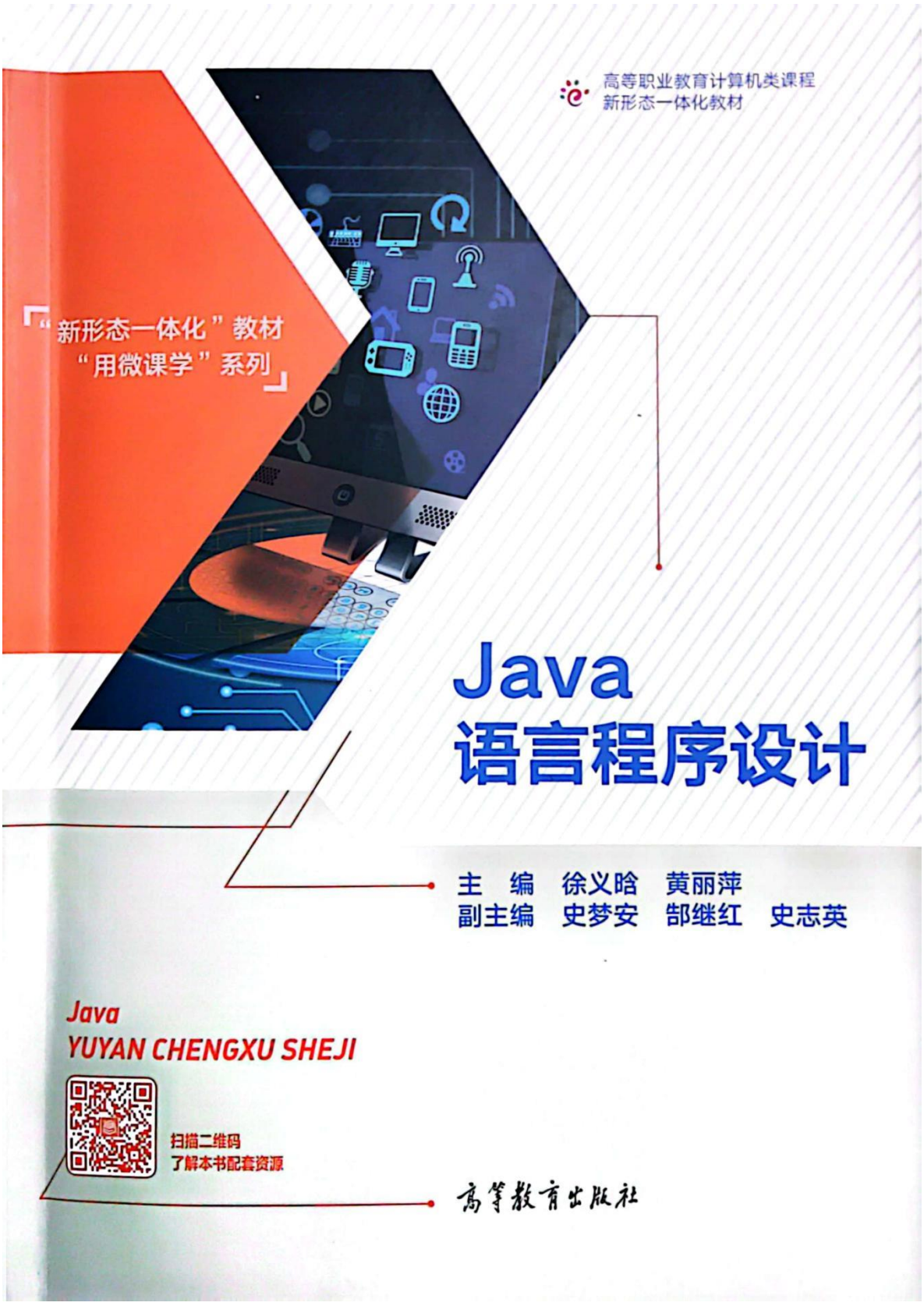
3.1 类和对象

3.2 定义类和创建对象

教学重点：类和对象的概念及创建方法,可自备教具及自备案例。

教材信息：教材名称《Java语言程序设计》，高等教育出版社，2020年11月出版（第2版），徐义晗、黄丽萍主编。

教材封面


高等职业教育计算机类课程
新形态一体化教材
“新形态一体化”教材
“用微课学”系列
Java
语言程序设计
主　编　徐义晗　黄丽萍
副主编　史梦安　郜继红　史志英
Java
YUYAN CHENGXU SHEJI
扫描二维码
了解本书配套资源
高等教育出版社

教学内容：第 3 章　面向对象程序设计

3.1　类和对象

3.2　定义类和创建对象

案例描述——教师和学生的描述

在学校中，主要包括学生和教师两大类主要人员。教师和学生又都属于人。人的基本信息主要包括姓名、性别、年龄、地址和电话，学生还包括学号和班级号，教师还包括工号和学历等信息。怎样使用 Java 语言来描述这些信息并且创建出实际存在的学生和老师对象呢?

知识储备

PPT 3-1：
类和对象的概念

PPT

3.1　类和对象

•3.1.1　对象的概念

微课 3-1：
类和对象的概念

对象（Object）是现实世界中实际存在的某个具体事物。例如：一本《Java 程序设计》的书、一张桌子、一张板凳等。人类在对事物进行描述的时候大多是从两个方面，即从静（特性、特征）和动（用途、行为）来展开。所以，对象包含静态的特征和动态的行为或用途。

在 Java 语言中，在对对象进行描述时，其静态的特征称为属性，动态的行为或用途称为方法。

•3.1.2　类的概念

人类在认识客观世界时，习惯把众多的事物进行归纳、划分或分类。把具有相同特征及行为的一组对象称为一类对象。面向对象思想中，类是同种对象的集合与抽象。例如，家用轿车、公交车、货车等都属于汽车类，并且通过比较可以发现不同的车之间有共同的特点。为了方便地了解和描述这些实际存在的实体，在面向对象思想中引入了类的概念，用于对所有对象提供统一的抽象描述，其内部包括属性和方法两个部分。

•3.1.3　类和对象的关系

对象是具体的一个个实实在在的事物，类是这些具体事物（对象）的原型，是这些具体事物（对象）一般性特征的描述。类与对象的关系如同模具和铸件的关系，类是创建对象的模具，而对象则是由类这个模具制作出来的铸件。对象与类的关系如图 3-1 所示。

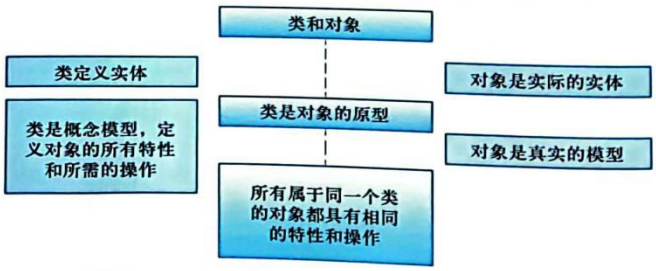


图 3-1
类和对象关系

3.2 定义类和创建对象

Java 中是如何来表述现实世界中具体的事物（对象）以及这些事物的一般特性（类）的呢？在 Java 中，类是面向对象程序设计的基本单位。类定义了某类对象的共有的变量和方法（即一般特性），类的属性是现实对象的特征或状态的数值表示，类的方法是对现实对象进行的某种操作或其对外表现的某种行为。通过类可以创建一个个具体的对象，对象是由一组相关的属性和方法共同组成的一个具体实体。

PPT 3-2：类的定义

PPT

微课 3-2：类的定义

3.2.1 类的声明

在 Java 中类声明的格式如下：

```
[类的修饰字] class 类名称 [extends 父类名称][implements 接口名称列表]
    {
        变量定义及初始化;
        方法定义及方法体;
    }
```

① 类的修饰字：[public] [abstract | final]。

- public 为类的访问控制符。Java 类具有两种访问控制符：public 和 default。public 允许类具有完全开放的可见性，所有其他类都可以访问它，省略 public 则为 default 可见性，即只有位于同一个包中的类可以访问该类。
- abstract 指明该类为一个抽象类，指该类是一个定义不完全的类，需要被继承，才能实例化创建对象。final 表明该类为最终类，不能被继承。

② class 是创建类所使用的关键字。

③ <classname> 是类的名称。

④ <body of class> 包含属性和方法的声明。

⑤ extends 为类继承，superClassName 为父类。如果在类定义时没有指定继承关系，则自己从 Object 类派生该类。

⑥ implements 为实现接口，interfaceNameList 为被实现的一个或多个接口名。

以上说明中，有些内容读者可能一时不理解，没有关系，这里先有个印象，在后续章节中会继续学到。

3.2.2 类的成员

类的成员包括属性（变量）和方法两个部分，定义格式如下。

1．成员变量定义格式

```
[变量修饰字] 变量数据类型 变量名 1，变量名 2[=变量初值]...;
```

变量修饰符可以为 [public | protected | private] [static] [final] [transient][volatile]。

成员变量的类型可以是 Java 中任意的数据类型，包括简单类型、类、接口、数组。

在一个类中的成员变量应该是唯一的。

2．成员方法定义格式

```
[方法修饰字] 返回类型　方法名称(参数 1,参数 2,…) [throws exceptionList]
{
    …(statements;)　//方法体：方法的内容
}
```

① 方法修饰字可以为[public | protected | default | private] [static] [final | abstract] [native] [synchronized]。

② 返回类型可以是任意的 Java 数据类型，当一个方法不需要返回值时，返回类型为 void。

③ 参数的类型可以是简单数据类型，也可以是引用数据类型（数组、类或接口），参数传递方式是值传递。

④ 方法体是对方法的实现。它包括局部变量的声明以及所有合法的 Java 指令。局部变量的作用域只在该方法内部。

说明

Java 定义了 4 种访问级别：public、protected、default 和 private。访问级别用来控制其他类对当前类的成员的访问。

具有 static 声明的成员属于静态成员，该成员属于类本身，不需要实例化就可以访问。

final 声明的变量为常量，final 声明的方法在类继承时不许子类覆盖。

transient 表明类成员变量不应该被序列化，序列化是指把对象按字节流的形式进行存储。

volatile 告诉编译器被 volatile 修饰的变量可以被程序的其他部分改变。

native 方法就是一个 Java 调用非 Java 代码的接口，该方法的实现由非 Java 语言实现，如 C。

synchronized 代表这个方法加锁，保证线程安全。

在例 3-1 中，创建一个立方体 Box 类，在其中定义 3 个变量表示一个立方体的长、宽和高；定义一个方法求立方体的体积；定义一个方法求立方体的表面积。

【例 3-1】

例题 3-1：Box.java

```
public class Box {
    double length;
    double width;
    double height;
    public double getV(){
        return length*width*height;
    }
    public double getArea(){
        return 2*(length*width+length*height+width*height);
    }
}
```

3.2.3 创建对象

PPT 3-3：对象的创建

PPT

以上的程序中定义了一个 Box 类，它只是对 Box 这一类东西的一个抽象的描述，需要通过它来产生一个有具体的长、宽、高大小的 Box。

要创建新的对象，需要使用 new 关键字和想要它创建对象的类名，如：

Box box1=new Box( ); 等号左边以类名 Box 作为变量类型定义了一个变量 box1，来指向等号右边通过 new 关键字创建的一个 Box 类的实例对象，变量 box1 就是对象的引用。注意，在 new 语句的类名后一定要跟这一对括号( )，Box( )被称为构造方法，在后面将重点讲解。

微课 3-3：对象的创建

对象中的属性和方法可使用圆点符号来访问，对象在圆点左边，而属性或方法在圆点右边，如 box1.length = 100.6; box1.getV ( );。

通过修改上面的例子来利用 Box 类创建对象 box1、box2，同时为了能运行测试结果，在 main( )方法中创建对象。

【例 3-2】

例题 3-2：Box.java

```
public class Box {
    double length;
    double width;
    double height;
    public double getV(){
        return length*width*height;
    }
    public double getArea(){
        return 2*(length*width+length*height+width*height);
    }
    public static void main(String args[]){
        Box box1=new Box();
        box1.length=200;
        box1.width=200;
        box1.height=200;
        System.out.println("第 1 个箱子的体积为:"+box1.getV()+",
            表面积为:"+box1.getArea());
        Box box2=new Box();
        box2.length=100;
        box2.width=100;
        box2.height=100;
        System.out.println("第 2 个箱子的体积为:"+box1.getV()+",
            表面积为:"+box1.getArea());
    }
}
```

程序运行结果如图 3-2 所示。

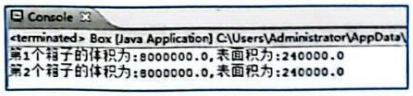


图 3-2
运行结果

PPT 3-4:
构造方法

PPT

3.2.4　构造方法

在上面的例子中用到 Box box1 = new Box( );语句来创建一个对象，new 可以理解为创建一个对象的关键字，通过使用 new 关键字为对象分配内存，初始化实例变量，并调用构造方法。那么 Box( )是什么意思呢？它在形式上和调用方法的形式相同。这个 Box( )就是一个特殊的方法叫构造方法。那为什么在程序中没有看到这个方法的定义呢，那是因为在没有定义构造方法的时候，系统会自己创建一个默认的构造方法。

为了让读者加深对构造方法的理解，来看下面的例子，在上面 Box 中添加一个方法。

微课 3-4:
构造方法

```
public Box(){
    System.out.println("来到构造方法");
}
```

程序的运行结果如图 3-3 所示：

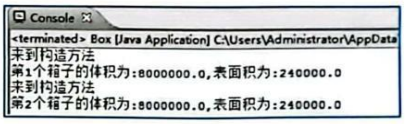


图 3-3
运行结果

通过运行结果读者会发现，在 main( ) 方法中并没有调用 Box( )方法，但它却被自动调用了，而且每创建一个 Box 对象，这个方法都会被自动调用一次，这就是“构造方法”。关于这个 Box( )方法，有以下几点不同于一般方法的特征。

① 它具有与类相同的名称。

② 它不含返回值。

③ 它不能在方法中用 return 语句返回一个值。

在一个类中，具有上述特征的方法就是“构造方法”。构造方法在程序设计中非常有用，它可以为类的成员变量进行初始化工作，当一个类的实例对象刚产生时，这个类的构造方法就会被自动调用，可以在这个方法中加入要完成初始化工作的代码，如为其中的变量赋初始值。

在构造方法里不含返回值的概念是不同于“void”的，对于“public void Person( )”这样的写法就不再是构造方法，而变成了普通方法，很多人都会犯这样的错误，在定义构造方法时加了“void”，结果这个方法就不再被自动调用了。

构造方法可以分为两类，一类是当程序没有定义构造方法时，系统自己生成的默认的构造方法，这个默认构造方法没有参数，在其方法体中也没有任何代码，即什么也不做，但是会对类成员变量进行默认的初始化。

类成员变量默认的初始化的值见表 3-1。