

零基础学 Java



1

类 (Class)

- 用变量表示商品
- 用类来表示商品

用变量表示商品

- 商品有标识，名字，数量，价格着几个属性
- 多个商品怎么办？重复，又是重复！

重新认识类（class）真正的用途

- 是用来描述同一类事物的
- 可以在内部定义任意数量的、不同类型的变量，作为这一类事物的属性。这种属性叫做成员变量（member variable）。
- 有类名，类名必须和文件名一样
- 就好像文件路径+文件名不能重复一样，一个Java程序中相同名字类只能有一个

看例程：定类的语法

2

初探类和对象

- 如何创建类的实例/对象（ Instance/Object ）
- 通过点操作符操作对象的属性

如何创建类的实例/对象 (Instance/Object)

- 从数据类型的角度来看，类就是自己创建了一种新的数据类型。类也叫做“自定义类型”。一个Java程序中不允许类同名。
- 看例程，学习类和对象的使用

通过点操作符操作对象的属性

认识点操作符

- 点操作符是用来访问/操作前面实体的属性的，类似于“的”
- `merchandise.name`可以读作merchandise的name。

3

认识引用类型

- 引用（reference）数据类型
- 引用数据类型和基本数据类型
- Java有一个大大的布告板，放着所有实例

引用（reference）数据类型

- Java 中的数据类型分为基本数据类型和引用数据类型
- 看例程，理解引用

引用数据类型和基本数据类型的相同点

- 都可以用来创建变量，可以赋值和使用其值
- 本身都是一个地址

引用数据类型和基本数据类型的不同点

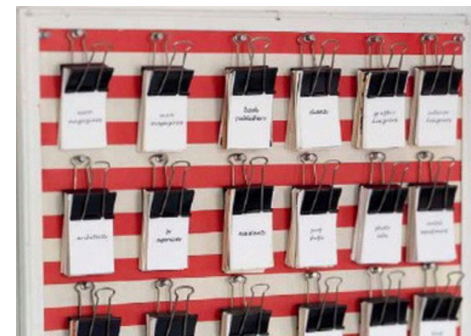
- 基本类型变量的值，就是地址对应的值。引用数据类型的值还是一个地址，需要通过“二级跳”找到实例
- 引用数据类型是Java的一种内部类型，是对所有自定义类型和数组引用的统称，并非特指某种类型

尝试输出一下引用类型的值到控制台，看看是什么

Java有一个的大大的布告板，放着所有实例

```
Merchandise m1 = new Merchandise();
```

- 使用 new 操作符可以创建某个类的一个实例。在 Java 程序运行的时候，所有这些创建出来的实例都被Java放在内存里一个叫做堆（heap）的、类似公告板的地方
- 创建一个实例，就是根据类的定义，点出需要的“纸”，订成一个本子，挂在布告板上。实例本身，可以认为是一个小本子
- 引用里存放的，相当于某个本子所在的布告板的地址



4

类，对象和引用的关系

类，对象和引用的关系

类和对象的关系

- 类是对象的模版，对象是类的一个实例
- 一个 Java 程序中类名相同的类只能有一个，也就是类型不会重名
- 一个类可以有很多对象
- 一个对象只能根据一个类来创建

引用和类以及对象的关系

- 引用必须是、只能是一个类的引用
- 引用只能指向其所属的类型的类的对象
- 相同类型的引用之间可以赋值
- 只能通过指向一个对象的引用，来操作一个对象，比如访问某个成员变量

5

认识数组类型

数组是一种特殊的类

- 数组的类名就是类型带上中括号
- 同一类型的数组，每个数组对象的大小可以不一样。也就是每个数组对象占用的内存可以不一样，这点和类的对象不同。
- 可以用引用指向类型相同大小不同的数组，因为它们属于同一种类型

引用的数组

- 可以把类名当成自定义类型，定义引用的数组，甚至多维数组

6

引用的缺省值——null

引用的缺省值——null

引用也有缺省值——null

- null是引用类型的缺省值
- null代表空，不存在。可以读作空
- 引用类型的数组创建出来，初始值都是空

null带来的问题

- 大名鼎鼎的 NullPointerException (NPE)
- 如果不确定，使用前要先判断引用是不是空

通过null理解引用的“二级跳”

7

像自定义类型一样使用类

像自定义类型一样使用类

类就是一种自定义类型

- 在类定义中可以使用类，创建类的引用
- 在类定义中，甚至可以使用类自己的类创建引用
- 引用类型的缺省值是null。一个类定义中如果有引用，创建出来的实例，其缺省值是null

8

Java中的包和访问修饰符

- 类多太混乱？用 package 管理
- 类使用太繁琐怎么办？用 import
- 属性访问修饰符：public
- 类的全限定名

类多太混乱？用 package 管理

- 为了避免类在一起混乱，可以把类放在文件夹里。这时就需要用 package 语句告诉 Java 这个类在哪个 package 里。package 语句要和源文件的目录完全对应，大小写要一致
- package 读作包。一般来说，类都会在包里，而不会直接放在根目录
- 不同的包里可以有相同名字的种类
- 一个类只能有一个 package 语句，如果有 package 语句，则必须是类的第一行有效代码

类使用太繁琐怎么办？用 import

- 当使用另一个包里的类时候，需要带上包名
- 每次使用都带包名很繁琐，可以在使用的类的上面使用 import 语句，一次性解决问题，就可以直接使用类了。就好像我们之前用过的 Scanner 类
- import 语句可以有多个
- 如果需要import一个包中的很多类，可以使用 * 通配符

属性访问修饰符：public

- 被 public 修饰的属性，可以被任意包中的类访问
- 没有访问修饰符的属性，称作缺省的访问修饰符，可以被本包内的其他类和自己的对象
- 访问修饰符是一种限制或者允许属性访问的修饰符

类的全限定名

- 包名 + 类名 = 类的全限定名。也可以简称为类的全名
- 同一个 Java 程序中全限定名字不可重复

9

打造一个小超市

- 小超市例程
- Java 的世界是一个类和对象的世界

小超市例程

Java 的世界是一个类和对象的世界

Java 就是使用类来描述世界，用类的实例（对象）让世界运转起来

各种操作数据的代码太乱了怎么办？

10

IntelliJ 调试程序初探

- debug : 不让程序不要一闪而过
- 用断点调试程序

debug：不让程序不要一闪而过

设置断点，debug调试模式运行程序

断点（ breakpoint ）：可以让程序在调试模式停在某一行

Frame 视图和 Variable 视图。视图的作用，隐藏和显示

程序调试标准动作

- 查看变量的值，展开实例看内部成员变量的值
- 程序继续执行之 Step Over：执行一行
- 程序继续执行之 Resume：继续执行直到遇到下一个断点或者程序结束（视频中有口误，Step Out是执行到方法结束，我们会在讲到方法的时候讲解这个功能。在这里因为我们只有一个方法，Step Out和Resume是一样的效果）
- 执行任意代码之 Evaluate Expression：在对话框输入代码，直接执行看结果值
- 条件断点：给断点设置条件，只有满足条件时，程序才会在该断点停住

使用快捷键和调试更配哦

11

方法：让 Merchandise 对象有行为

- 如果要描述一个商品怎么办？
- 方法——让可以商品描述自己
- 详解describe方法

如果要描述一个商品怎么办？

- 看例程
- 每次需要描述商品，都要重复这些代码？一个成熟的商品应该学会自己描述自己

方法（Method）——让商品描述自己

方法英文名叫做method，又称作function。看例程，学习方法的语法

方法的调用

- 通过引用的点操作符，可以调用对象的方法
- 方法调用要有括号，即使没有参数

方法可以使用的数据：对象的成员变量（member variable）

12

返回值：让 Merchandise
计算利润

让商品自己计算利润

- 看例程：让商品自己计算利润的方法
- 看例程：让超市找出利润最高的商品的方法
- 看例程：商品的其它方法
- 方法调用调试之 Step Into：进入被调用的方法内部继续调试

13

参数：让 Merchandise 计算多件商品的总价

- 第二件半价哦！
- 参数：告诉商品实例要购买多少个

第二件半价哦！

看例程

这种方式有哪些问题？

- 每个用到这个逻辑的地方，都要重复这段代码
- 如果告诉商品要买多少个，商品帮我算多少钱，就不用重复代码了

参数：告诉商品实例要购买多少个

看MerchandiseV2中的新方法，学习方法参数相关语法

14

参数和返回值是怎么传递的

参数和返回值是怎么传递的

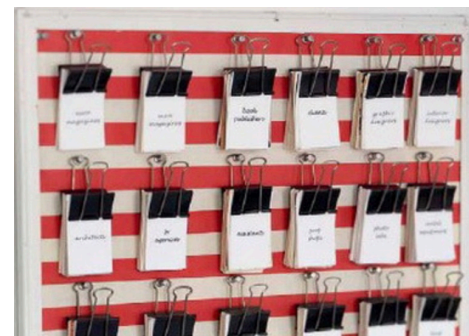
- 参数和方法里的局部变量可以认为是一样的东西。只是在方法调用之前，会用实参给参数的形参赋值
- 发生在代码块里的，就让它留在代码块里。方法执行完毕，参数和方法的局部变量的数据就会被删除回收。就好像演草纸，作用是计算一个值，算好之后，演草纸就可以扔了
- 调用一个有返回值的方法时，就好像访问一个成员变量
- 看代码，学习参数传递
- 看代码，学习返回值的传递

15

分清参数、局部变量和实例的地盘

分清参数、局部变量和实例的地盘

- 局部变量就是我们之前说的变量，是在方法体里创建的变量
- 参数和局部变量都是草稿纸，方法执行完就清除了
- 对象是实体/实例，不是变量。对象创建出来后，被堆在一起，放在类似公告板的地方。方法里创建的对象是不会随着方法结束被清除的。所以对象的地盘不受限制，只要有引用指向一个对象，这个对象的数据就可以通过这个引用来访问
- 看例程：理解局部变量，参数和实例



16

隐藏的 this 自引用

- 参数和局部变量重名会怎样
- 使用this：访问局部变量的完整形态

参数和局部变量重名会怎样

使用this：访问局部变量的完整形态

- 看例程，理解this自引用的用法和意义

17

理解方法：一种特殊的代码块

- 方法是什么
- 方法的特殊之处

方法是什么

- 类中如果不定义方法，只定义成员变量，那么一个类就没有了功能，只是简单的数据的封装。创建一个对象之后，所有对这些数据的操作，都要在每个用到这些数据的地方写代码
- 类通过成员变量和方法描述世界。成员变量是描述一类事物的属性，是数据；方法是描述一类事物的行为和功能，是对数据的操作。比如最开始我们写的 describe 方法
- 方法中的代码可以通过操作一个对象的成员变量，完成一个功能

方法是什么（续）

- 方法是 Java 中代码执行的单元，是代码的载体。所有的代码，都必须属于某一个方法
- 方法就是一串语句，加上数据输入 this 自引用和参数，执行后得到一个返回值。所以使用一个对象调用一个方法，可以叫做调用对象的方法，也可以叫做“在这个对象上调用方法（ invoke a method on an object ）”
- 方法不是对象的一部分，它是类的一部分。每个对象可以给成员变量赋不同的值，但是无法让方法有不同的行为。同理，无论在一个类中定义多少方法，都不会影响创建一个对象所占用的内存

方法的特殊之处

- 和代码块比，方法特殊在：
 - 有名字
 - 有返回值
 - 有参数
 - 有 this 自引用
 - 明确的属于某一个类
 - 可以（也只能）通过对象引用来调用

18

理解方法的调用：代码的一种特殊跳转

理解方法的调用：代码的一种特殊跳转

- 看例程：模拟this自引用传递
- 看例程：对象可以在自己的方法里调用自己类的别的方法，甚至可以调用自己的方法。
- 方法在类中定义的先后，没有任何影响。
- 方法是一种根据名字做跳转的代码流程控制方式。它的特殊之处在于有this自引用，有参数，有返回值。

19

给类和方法加 Java 注释

给类和方法加 Java 注释

- 多行注释
- 给类添加注释 (Javadoc)
- 给方法添加注释 (Javadoc)
- Oracle 官方的 Javadoc 地址

<https://docs.oracle.com/en/java/javase/11/docs/api/index.html>

