



链滴

## 009 对象和类

作者: [pzs233](#)

原文链接: <https://ld246.com/article/1537283768832>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## 9.1 为对象定义类

类为对象定义属性和行为。

对象 (object) 代表现实世界装可以明确表示的一个实体。

每一个对象都有自己独特的标识、状态和行为。

- 一个对象的状态 (state, 也称为特征 (property) 或属性 (attribute)) 是由具有当前值的数据来表示的。
- 一个对象的行为 (behavior, 也称为动作 (action)) 是由方法定义的。

类是一个模板、蓝本或者说是合约, 用来定义对象的数据域是什么以及方法时做什么的。

一个对象是类的一个实例。可以从一个类中创建多个实例。

创建实例的过程称为实例化 (instantiation)。对象 (object) 和实例 (instance) 经常时可以互换。

## 9.4 使用构造方法构造对象

构造方法在使用 new 操作符创建对象的时候被调用。

- 构造方法必须具备和所在类相同的名字
- 构造方法没有返回值类型, 甚至连 void 也没有
- 构造方法是在创建一个对象使用 new 操作符时调用的。构造方法的作用就是初始化对象。

## 9.5 通过引用变量访问对象

### 基本类型变量和引用类型变量的区别

每个变量都代表一个储存值的内存位置。

- 对基本类型变量来说, 对应内存所存储的值是基本类型值。
- 对引用类型变量来说, 对应内存所存储的值是一个引用, 是对象的存储地址。

如果你认为不再需要某个对象时, 可以显式地给该对象的引用变量附 null 值。如果该对象没有被任何引用变量所引用, Java 虚拟机将自动回收它所占的空间。

## 9.6 使用 Java 库中的类

### Date 类

```
java.util.Date
+Date()
+Date(elapseTime: long)
```

```
+toStrin(): String
+getTime(): long
+setTime(elapseeTime: long): void
```

## Random 类

```
java.util.Random
+Random()
+Random(seed: long)
+nextInt(): int
+nextInt(n: int): int
+nextLont(): long
+nextDouble(): double
+nextFloat(): float
+nextBoolean(): boolean
```

## 9.7 静态变量、常量和方法

要声明一个静态变量或定义一个静态方法，就要在这个变量或方法的声明中加上修饰符 `static`。

- 调用静态方法： `className.staticMethod`
- 调用静态变量： `className.staticVariable`

## 9.8 可见性修饰符

- 被 `public` 修饰的类、方法、数据域 可被任何其他类访问
- 没有可见性修饰符的类、方法、数据域 可被同一个包内的任何类访问

`private` 只能修饰类成员。

## 9.9 数据域封装

**数据域封装 (data field cncapsulation)**: 为了避免对数据与的直接修改，可使用 `private` 修饰符将据域声明为私有。

为了能够访问私有数据域，可以提供一个 `get` 方法（访问器，accessor）返回数据域的值，一个 `set` 法（修改器，mutator）给数据域设置新值。

## 9.10 向方法传递对象参数

给方法传递一个对象，是将对象的引用传递给方法。

## 9.11 对象数组

对象数组实际上是引用变量的数组。

## 9.12 不可变对象和类

可以定义不可变类来产生不可变对象。不可变对象的内容不能被改变。

不可变类 必须满足:

- 所有数据域都是私有的
- 没有修改器方法
- 没有一个返回指向可变数据域的引用的访问器方法

## 9.13 变量的作用域

实例变量和静态变量的作用域是整个类，无论变量是在哪里声明的。

## 9.14 this 引用

关键字 this 引用对象自身。

- 使用 this 引用隐藏数据域
- 使用 this 调用构造方法

END