



链滴

# 011 继承和多态

作者: [pzs233](#)

原文链接: <https://ld246.com/article/1537286149575>

来源网站: [链滴](#)

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

## 11.2 父类和子类

**继承**：面向对象的编程允许你从已经存在的类中定义新的类。

子类中它的父类中继承可访问的数据域和方法，还可以添加新的数据域和新方法。

**关键字**: `extends`

几个注意的：

- 一个子类通常比它的父类包含更多的信息和方法
- 父类中的私有数据域在该类之外是不可访问的
- 不是所有的“是一种” (is-a) 关系都该用继承来建模
- 继承是用来为“是一种”关系 (is-a) 建模的
- 单一继承：一个 Java 类只能直接继承自一个父类

## 11.3 super 关键字

关键字 `super` 指代父类，可以用于调用父类中的普通方法和构造方法。

构造方法用于构造一个类的实例。不同于属性和普通方法，父类的构造方法不会被子类继承。它们只使用关键字 `super` 从子类的构造方法中调用。

调用父类的构造方法必须出现在子类构造方法的第一行，这是显式调用父类构造方法的唯一方式。

构造方法可以调用重载的构造方法或父类的构造方法。如果它们都没有被显式地调用，编译器就会自动地将 `super()` 作为构造方法的第一条语句。

所以，其实每个子类构造方法的第一条语句，都是隐含地调用 `super()`，如果父类没有这种形式的构造数，那么在编译的时候就会报错。

故，一般情况下，最好为每个类提供一个无参构造方法，以便于对该类进行拓展。

**构造方法链**：在任何情况下，构造一个类的实例时，将会调用沿着继承链的所有父类的构造方法。当构造一个子类的对象时，子类构造方法会在完成自己的任务之前，首先调用它的父类构造方法。如果父类继承自其它类，那么父类构造方法又会在完成自己的任务之前，调用它自己的父类的构造方法。这个过程持续延续到沿着这个继承体系结构的最后一个构造方法被调用为止。

## 11.4 方法重写

**方法重写 (method overriding)**：子类修改从父类中继承的方法的方法实现。

注意：

- 仅当父类的实例方法是可访问时，它才能被覆盖。
- 与实例方法一样，静态方法也能被继承。但是，静态方法不能被覆盖。如果父类中的静态方法在子中被重新定义，那么父类中定义的静态方法将被隐藏。

## 11.5 方法的重写和重载

- 重载 意味着使用同样的名字但是不同的签名来定义多个方法。
- 重写 意味着再来子类中提供一个对方法的新的实现。

注意：

- 方法重写发生在通过继承而相关的不同类中；方法重载可以发生在同一个类中，也可以发生在由于承而相关的不同类中。
- 方法重写具有同样的签名和返回值类型；方法重载具有同样的名字，但是不同的参数列表。

重写标注 (override annotation) : 在重写的子类方法前面放一个 `@Override`。

## 11.6 Object 类及其 toString() 方法

Java 中所有的类都继承自 `java.lang.Object` 类。所以，所有的方法都能使用 `java.lang.Object` 类中方法。

## 11.7 多态

多态意味着父类的变量可以指向子类对象。

总可以将子类的实例传给父类型的参数。

**多态 (polymorphism)** : 使用父类对象的地方都可以使用子类的对象。简单来说，多态意味着父类的变量可以引用子类型的对象。

## 11.8 动态绑定

- 声明类型：一个变量必须被声明为某种类型，这个类型称为它的声明类型。
- 实际类型：被变量引用的对象的实际类。

方法可以在沿着继承链的多个类中实现。JVM 决定运行时调用哪个方法。

**动态绑定**：一个变量调用哪个方法，是由它的实际类型决定的。

## 11.9 对象转换和 instanceof 运算符

对象转换：对象的引用可以类型转换成对另一种对象的引用。

- 隐式转换 (子类型转换成父类型，向上转换)
- 显式转换 (父类型转换成子类型，向下转换)

`a instanceof b` : 判断 a 是否为 b 的子类，是返回 true，不是返回 false。

转换一个对象引用不会创建一个新的对象。

## 11.10 Object 类的 equals 方法

在子类中，使用签名 equals(SomeClassName obj) 重写 equals 方法是一个常见的错误，应该使用 equals(Object obj)。

## 11.11 ArrayList 类

ArrayList 对象可以用于存储一个对象列表。

## 11.12 对于列表有用的方法

- asList
- sort
- max、min
- shuffle

## 11.14 protected 数据和方法

数据和方法的可见性

类成员修饰符 子类内可访问	在同一类内可访问	在不同包可访问	在同一包内可访问	在同一包内可访问
public	Y	Y	Y	Y
protected	Y	Y	Y	-
(default)	Y	Y	-	-
private	Y	-	-	-

修饰符 private 和 protected 只能用于类的成员。

public 和 默认修饰符（也就是没有修饰符）既可以用于类的成员，也可以用于类。

一个没有修饰符的类（即非公共类）是不能被其他包中的类访问的。

## 11.15 防止拓展和重写

一个被 final 修饰的类和方法都不能被拓展。被 final 修饰的数据域是一个常数。

END