讨论下面内容：

1. 从方法中返回值
2. This关键字
3. 类和实例化成员
4. Access 控制

# **Returning a Value from a Method**

遇到下面三种情况，一个方法返回到调用的地方：

1. 运行完所有代码
2. Return
3. 抛出异常

你在方法声明中声明了方法的返回类型，在方法主题中，你使用return语句返回值。

任意声明了void的方法都不返回值，它并不需要包含一个return声明。在这种情况下，一个return语句可以用来

### 返回一个类或者接口：

当一个方法使用类名字作为返回类型，比如，返回对象的类型必须类对象本身，要么是子类。

不同返回类型可以重载。

## 使用this关键字

使用thisfiled的最常见的原因是，一个field被构造器参数shadow。

用构造器使用this，在狗在其中，你也可以使用this关键字来调用同一类中其他构造器。

### 控制转移到类成员

访问级别修饰器决定了其他类能够使用一个特别的类或者调用一个特别的方法。有两个级别的控制

1. 最上级：public，或者说package private
2. 成员级别：public，private，包括或者package private

类修饰器：如果public，那么任何部分都可以使用这个类。如果没有声明类是public，那么就是package private也就是只能在package内部使用。

Member：你可以使用public修饰器或者使用和类一样的修饰器。对于成员，有两个额外的修饰器：。Private修饰器声明了成员只能在类内部使用，protect修饰器声明了成员只能在自己的包或者另外一个包的子类中访问。

第一列声明了类是哦后能够获取成员。

第二列声明了类

第三列声明了在同样的包内部，其他类能够获取成员

第四列声明了外部包的子类能否获取成员

第五列声明了是否所有类能够获取成员

访问级别从两方面影响着你，首先，从其他源中使用子列，访问级别决定了这些类中哪些成员你的类可以使用。第二，当你写类的时候，你需要决定，每个成员变量和方法的访问级别是什么样的。

设置访问级别的建议：

1. 使用最严谨的访问级别，也就是，一般都设置为private，除非你有别的原因。
2. 除了常数，避免用public。Public域会将你连接到一个特殊的是实现，并限制你修改你的代码。

### 理解类成员：

初始化域，正如你所看到你，你可以提供域的初始值。

当初始化值可见，实例化可以放在一行时，这个样鞋没问题。然而， 这种实例化的方式会限制简单些。如果实例化需要多层逻辑，那么简单的复制不客气。实例变量可以用构造器实现，其中可以使用逻辑。为了提供类成员同样的能力。Java编程语言提供静态实例化块。

我们不需要从类定义开始来声明。只需要确保在使用前已经被声明并初始化了即可。

### 静态初始化块：

一个静态初始化块就是被{}里面的内容

一个类可以有任意熟练的静态初始化块。运行系统保证初始化按照顺序被调用。

你可以写一个静态私有方法。

### 初始化实例成员：

一般情况，你会在构造器中初始化实例变量。有两个其他方式来初始化实例变量。

初始化块：

Java编译器将初始化块放进所有构造器，因此这个方法可以用来共享代码。

Final方法：

Final方法不能被子类重写。