

重载与重写的区别

作者: [zhangshaocui](#)

原文链接: <https://ld246.com/article/1508665262867>

来源网站: [链滴](#)

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

1 重载(Overloading)

(1) 方法重载是让类以统一的方式处理不同类型数据的一种手段。多个同名函数同时

存在，具有不同的参数个数/类型。重载 Overloading 是一个类中多态性的一种表现。

(2) Java 的方法重载，就是在类中可以创建多个方法，它们具有相同的名字，但具

有不同的参数和不同的定义。调用方法时通过传递给它们的不同参数个数和参数类型来决定具体使用哪个方法，这就是多态性。

(3) 重载的时候，方法名要一样，但是参数类型和数不一样，返回值类型可以相同

也可以不相同。无法以返回型别作为重载函数的区分标准。

2 重写 (Overriding)

(1) 父类与子类之间的多态性，对父类的函数进行重新定义。如果在类中定义某

方法与其父类有相同的名称和参数，我们说该方法被重写 (Overriding)。在 Java 中，子类可继承父类中的方法，而不需要重新编写相同的方法。但有时子类并不想原封不动地继承父类的方法，而是想作一定的修改，这就需要采用方法的重写。方法重写又称方法覆盖。

(2) 若子类中的方法与父类中的某一方法具有相同的方法名、返回类型和参数表，

则新方法将覆盖原有的方法。如需父类中原有的方法，可使用 super 关键字，该关键字引用了当前类的父类。

(3) 子类函数的访问修饰权限不能少于父类的；

重写方法的规则如下：

ol

li 参数列表：必须与被重写方法的参数列表完全匹配。

li 返回类型：必须与超类中被重写的方法中声明的返回类型或子类型完全相同

li 访问级别：一定不能比被重写方法强，可以比被重写方法的弱。

li 非检查异常：重写方法可以抛出任何非检查的异常，无论被重写方法是否声明了该异常。

li 检查异常：重写方法一定不能抛出新的检查异常，或比被重写方法声明的检查异常更广的检查异常

li 不能重写标志为 final,static 的方法

ol

****重载方法的规则： ****

ol

li 参数列表：被重载的方法必须改变参数列表。

li 返回类型：可以改变返回类型。

li 修饰符：可以改变修饰符

li 异常：可以声明新的或者更广泛的异常。

ol

```
class Animal {  
    public void eat()
```

```
}
```

```
}
```

```
</pre>
```

以下列出对于 Animal 的 eat 方法各种重写重载实例，根据以上列出的规则，判断其是否合法（下的方法都属于 Animal 的子类 Horse 的方法）：

ol

li private void eat(){} 不能通过编译，非法重写，访问修饰符限制性变强；不属于重载，因为参数列表没有发生改变

li public void eat() throw RuntimeException{} 能通过编译，属于重写，重写方法可以抛出任何

检查异常。

public void eat() throw IOException{} 不能通过编译，非法重写，重写方法一定不能抛出新的查异常，或比被重写方法声明的检查异常更广的检查异常；不属于重载，参数列表必须发生变化才属重载

public void eat(String food){} 能通过编译，不是重写；合法重载，因为参数列表发生改变。

public String eat(){ } 不能通过编译，不是重写，因为其返回类型；不是重载，因为参数列表没发生改变。

public String eat(int n){} 可以通过编译，不是重写；是重载，首先参数列表必须发生变化，返类型可以发生改变。

<p>对父类被重写的方法做一个变动： </p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> class Animal {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">     public Animal e
</span> </span> <span class="highlight-line"> <span class="highlight-cl">         t() throws IOException{
</span> </span> <span class="highlight-line"> <span class="highlight-cl">             return null;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">         }
</span> </span> <span class="highlight-line"> <span class="highlight-cl">     }
</span> </span> </code> </pre>
```

<ol start="7">

public Animal eat(){return null;} 可以通过编译，合法重写，可以不抛出异常，只要不抛出新异常或更广泛的异常就可以。

public Animal eat() throws FileNotFoundException{return null;} 可以通过编译，是重写，属子类的非检查异常。

public Animal eat() throws Exception{} 不可以通过编译，非法重写，抛出了更广泛的异常。

public Horse eat() {} 可以通过编译，是重写，因为返回类型可以是被重写的返回类型的子类。

