接口的定义及使用

# 引言

1. 接口的基本定义以及使用形式；
2. 与接口有关的设计模式的初步认识；
3. 接口与抽象类的区别。

# 具体内容

接口与抽象类相比，接口的使用几率是最高的，所有的设计几乎都是围绕接口进行的，但是要想把接口彻底明白，需要很长一段时间。

## 接口的基本概念

接口是一种特殊的类，但是在接口里面的组成与类不同，比类的组成部分简单，主要由抽象方法和全局常量所组成。而接口使用interface关键字来定义。

当一个接口定义完成之后，需要遵循如下的步骤进行接口的使用：

* 接口一定要定义子类，子类利用implements关键字来实现接口，一个子类可以同时实现多个接口。
  + 秒杀抽象类的单继承局限，一个抽象类只能够被一个子类所继承；
* 接口的子类（如果不是抽象类）那么必须覆写接口中的全部抽象方法；
* 接口的对象利用子类对象的向上转型进行实例化操作。

注意：关于接口的组成描述

接口里面在定义的时候就已经明确的给出了开发要求：抽象方法和全局常量，所以以下两种接口的定义从本质上讲是完全一样的。

如果在定义接口方法的时候没有使用public，那么本质上也不是default权限，而默认就是public。

现在一定要知道，抽象类可以实现接口，但是反过来，接口可不能继承抽象类，但是一个接口却可以使用extends关键字继承多个父接口。

虽然接口本身只能够有抽象方法和全局常量，但是内部的结构是不受限制的，所以接口内部可以继续定义内部类、内部抽象类、内部接口。如果一个内部接口上使用了static定义，那么这个内部接口就属于外部接口。

总之对于接口的使用可以发现有如下几点：

* 接口避免了单继承的局限，一个子类可以实现多个接口；
* 接口中的权限统一为public，方法都是抽象方法；
* 所有的内部类结构都不受到定义语法的限制，static定义的内部接口就是外部接口。

实际开发中接口的三个使用原则：

* 制定操作标准；
* 表示一种能力；
* 将服务器端远程方法视图提供给客户端。

## 接口的实际应用——标准

现实生活中对于接口这个名词应该不陌生。

## 抽象类与接口的区别

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

# 总结

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |