



# Java程序设计

---

## 第6章 接口



# 第 6 章 接口

---

- 6.1 接口简介
- 6.2 定义接口
- 6.3 接口的继承
- 6.4 接口的实现
- 6.5 接口与抽象类
- 6.6 接口的UML图
- 6.7 接口回调
- 6.8 接口与多态
- 6.9 接口参数
- 6.10 面向接口编程



## 6.1 接口简介

Java中的接口是一个特殊的抽象类，接口中的所有方法都没有方法体。比如，定义一个人类，人类可以为老师，可以为学生，所以人这个类就可以定义成抽象类，还可以定义几个抽象的方法，比如讲课，看书等，这样就行成了一个接口。如果你想要一个老师，那么就可以实现人类这个接口，同样可以实现人类接口中的方法，当然，也可以存在老师特有的方法。就像USB接口一样，只需把USB接到接口上，就能实现你想要的功能。

## 6.2 定义接口

语法格式如下:

```
[修饰符] interface 接口名 [extends 父接口名列表]{  
    [public] [static] [final] 常量;  
    [public] [abstract] 方法;  
}
```

修饰符: 可选, 用于指定接口的访问权限, 可选值为public。如果省略则使用默认的访问权限。

接口名: 必选, 用于指定接口的名称, 接口名必须是合法的Java标识符。一般情况下, 要求首字母大写。

extends 父接口名列表: 可选参数, 用于指定要定义的接口继承于哪个父接口。当使用extends关键字时, 父接口名为必选参数。

方法: 接口中的方法只有定义而没有被实现。





## 6.3 接口的继承

---

接口是可以被继承的。但是接口的继承与类的继承不太一样，接口可以实现多继承，也就是说接口可以有多个直接父接口。和类的继承相似，当子类继承父类接口时，子类会获得父类接口中定义的所有抽象方法、常量属性等。



## 6.4 接口的实现

接口可以被类实现也可以被其他接口继承。在类中实现接口可以使用关键字 `implements`。

语法格式为：

```
[修饰符] class <类名> [extends 父类名] [implements 接口列表]{  
}
```

修饰符：可选，用于指定类的访问权限，可选值为 `public`、`final` 和 `abstract`。

类名：必选，用于指定类的名称，类名必须是合法的Java标识符。一般情况下，要求首字母大写。

`extends` 父类名：可选参数，用于指定要定义类继承于哪个父类。当使用 `extends` 关键字时，父类名为必选参数。

`implements` 接口列表：可选参数，用于指定该类实现哪些接口。当使用 `implements` 关键字时，接口列表为必选参数。当接口列表中存在多个接口名时，各个接口名之间使用逗号分隔。

信息科学与工程学院



## 6.5 接口与抽象类

---

接口与抽象类的共同点如下：

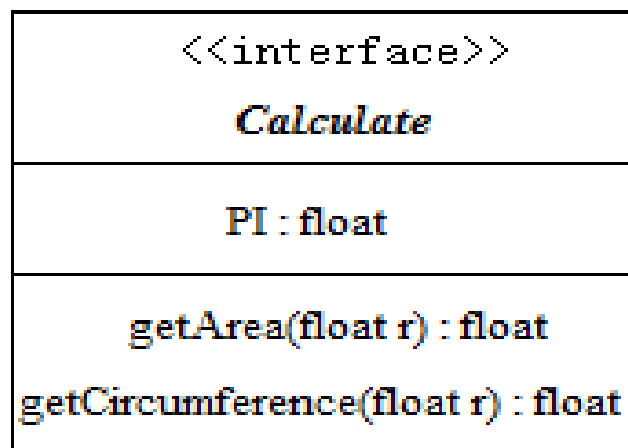
- (1) 接口与抽象类都不能被实例化，能被其他类实现和继承
- (2) 接口和抽象类中都可以包含抽象方法，实现接口或继承抽象类的普通子类都必须实现这些抽象方法。

# 接口与抽象类的用法差别如下:

- (1) 接口中只能包含抽象方法, 不能包含普通方法; 抽象类中可以包含普通方法。
- (2) 接口中不能定义静态方法; 抽象类中可以定义静态方法。
- (3) 接口中只能定义静态常量属性, 不能定义普通属性; 抽象类里可以定义静态常量属性, 也可以定义普通属性。
- (4) 接口不能包含构造器; 抽象类可以包含构造器, 抽象类里的构造器为了让其子类调用并完成初始化操作。
- (5) 接口中不能包含初始化块, 但抽象类可以包含初始化块。
- (6) 一个类最多只能有一个直接父类, 包括抽象类; 但是一个类可以实现多个接口。



## 6.6 接口的UML图



顶部第1层是名字层，接口的名字必须是斜体字形，而且需要用<<interface>>修饰，并且该修饰和名字分布在两行。

中部第2层是常量层，列出接口中的常量及类型，格式是“常量名字：类型”。

底部第3层是方法层，也称操作层，列出接口中的方法及返回类型，格式是“方法名字（参数列表）：类型”。





## 6.7 接口回调

接口也是Java中的一种数据类型，使用接口声明的变量称做接口变量。接口变量属于引用型变量，接口变量中可以存放实现该接口的类的实例的引用，即存放对象的引用。例如，假设Peo是一个接口，可以使用Peo声明一个变量：

```
Peo pe;
```

Java中的接口回调指的是：把实现某一接口的类所创建的对象引用赋值给该接口声明的接口变量，那么该接口变量就可以调用被类实现的接口方法。实际上，当接口变量调用被类实现的接口方法时，就是通知相应的对象调用这个方法。



## 6.8 接口与多态

---

由接口实现的多态就是指不同的类在实现同一个接口的时候可能具有不同的表现方式。

【例】使用Dog类和Cat类都实现了接口Animals接口。



## 6.9 接口参数

---

如果一个方法的参数是接口类型的参数，我们就可以将任何实现该接口的类的实例的引用传递给该接口参数，那么接口参数就可以回调类实现的接口方法。

【例】 实现接口的回调参数。



## 6.10 面向接口编程

面向接口编程是对多态特性的一种体现，面向接口编程是使用接口来约束类的行为，并为类和类之间的通讯建立实施标准。使用面向接口编程增加了程序的可维护性和可扩展性。可维护性体现在：当子类的功能修改时，只要接口不发生改变，系统其他代码就不需要改动。可扩展性体现在：当增加一个子类时，测试类和其他代码都不需要改动，如果子类增加其他功能，只需要子类实现其他接口即可。使用接口可以实现程序设计的“开-闭原则”，即对扩展开放，对修改关闭。如图所示，当多个类实现接口，接口变量variable所在的类不需要做任何修改，就可以回调类重写的接口方法。



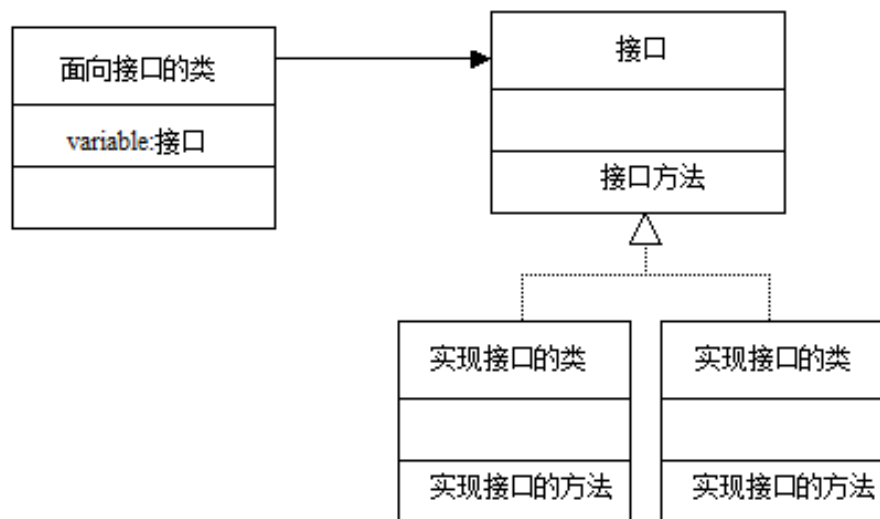


图6-7 实现接口的UML图