# Java三大特性

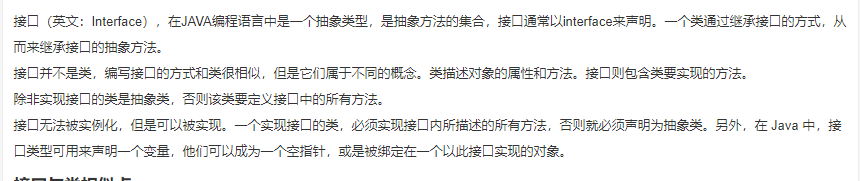
封装，继承，多态。

封装，继承，多态，抽象。

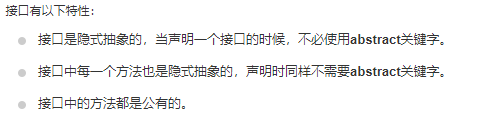
# 接口

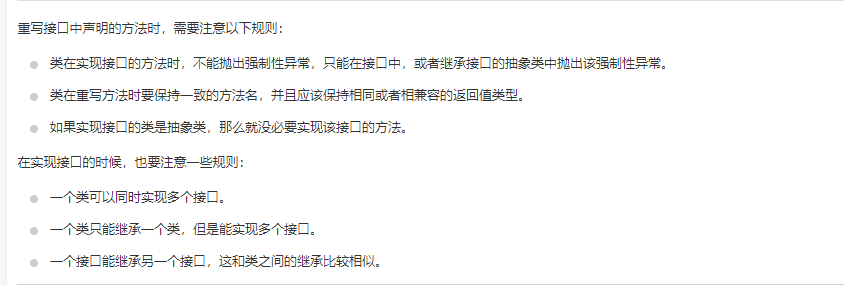
实现接口的类除非是抽象类，否则要实现接口中所有的方法。

接口与接口之间是继承关系，并且接口可以实现多继承。类不可以多继承。

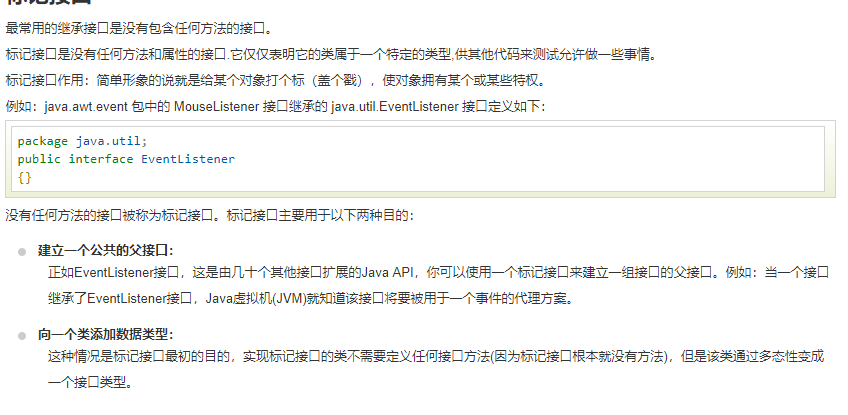








## 标记接口



# 抽象类

所有的对象都是通过类描述的，但是反过来，并不是所有的类都是用来描述对象的。

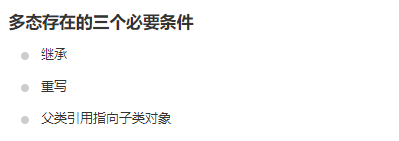
抽象类除了不能实例化对象外，类的其他功能都存在。成员变量，成员方法，构造方法的访问方式和普通类一样。

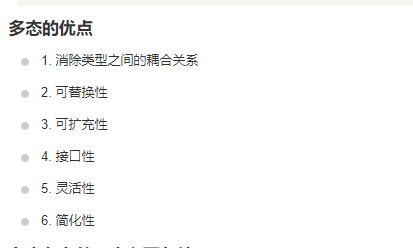
子类要实现所有的父类抽象方法，否则子类也是抽象类。

# 静态

静态方法可以通过类名.方法名直接调用。

# 多态





当使用多态调用方法时，首先检查父类中是否有该方法，如果没有，则编译错误。如果有，则再去调用子类的同名方法。

多态分为编译时多态和运行时多态，编译时多态是静态的，指的是方法的重载。运行时多态是动态的。

# Java8新特性

## Lambda表达式

## 函数式编程

## 默认方法

## Stream

## Optional

## 新的日期时间API

Local和Zoned

### Local

简化了日期时间的处理，没有时区的问题。

LocalDate、LocalTime、LocalDateTime

### Zoned

通过定制的时区处理日期时间

# 基本数据类型

基本数据类型没有父类，基本数据类型的包装类的父类是java.lang.Number.

String的父类是Object。

数组的父类是Object

Number的父类是Object

# Native

Java 调用一个非java代码的接口，比如有些功能是c实现的，java调用的时候就可以使用这个关键字进行调用。

Native可以和其他关键字连用，但是abstract除外。native暗示这些方法是有实现体的，只不过实现体不是java，但是abstract是没有实现体的。

|  |
| --- |
| public class HelloWorld {  public native void displayHelloWorld();  static {  System.loadLibrary("hello");  }  public static void main(String[] args) {  new HelloWorld().displayHelloWorld();  }  } |

声明native方法：如果你想将一个方法做为一个本地方法的话，那么你就必须声明改方法为native的，并且不能实现。其中方法的参数和返回值在后面讲述。  
Load 动态库：System.loadLibrary("hello");加载动态库（我们可以这样理解：我们的方法displayHelloWorld()没 有实现，但是我们在下面就直接使用了，所以必须在使用之前对它进行初始化）这里一般是以static块进行加载的。同时需要注意的是 System.loadLibrary();的参数“hello”是动态库的名字。