# 内部类

匿名内部类

没有名字的一个内部类，叫匿名内部类

内部类主要作用

可以把数据封装到一个类中处理

内部类可以访问外部类的资源，内部类融入到外部类中，可以看做是同一个类中的资源

匿名内部类 拥有内部类的功能，同时，也方便编程。

不需要对接口实现很多类，只要用一次的类，那么就应该用匿名内部类

Instanceof 判断一个对象它是不是 后面定义的类型

静态的，不需要实例化就能访问资源，资源共享

非静态。成员属性，成员方法，需要实例化对象才能访问

静态的不能访问非静态的，

非静态的，可以访问静态的资源

# 垃圾

内存中的对象，如果不适用，那么就有可能成为垃圾

GC 垃圾回收机制

面向对象 oop

封装 修饰符的使用

继承 哪些能继承 ，方法的重写 引用 转型

多态 必须是继承的，方法要重写

接口 属性 公开的 静态 常量 方法只有定义，但是1.8，default 可以实现了

抽象类 abstract 方法 可以有实现的方法，必须是继承类去实现它

枚举 默认继承Enum类 项目就是类中的静态成员 ，如果遇到其他构造函数，重新定义，作用 避免数据的传递错误。定义什么类型的数据，传递的时候就固定

内部类 静态 ，非静态的 怎么去调用 内部调用外部调用

new Outer().Inner()

outer.new.Inner()

匿名内部类 父类，接口

New 接口名(){

}

判断对象是否相等

== 判断的是两个对象内存地址

Equals 判断的也是两个对象内存地址是否相等

但是可以重写

# 异常处理

程序一旦遇到错误，就会被终止

有些错误是可以容忍的，只要不是“原则性”的错误

有些错误不会让程序终止，继续程序，异常，可以进行处理

异常的技术不难理解，只要是会在特点场合使用异常处理

对一些可能发生的错误，但是又不确定是什么情况，或者是什么数据的时候等等，可以使用异常处理，简化程序。

Try 出现的异常要能被捕获 ，不能被捕获则，等于没有异常处理

Throws 抛出异常

运行时异常，编译型异常

# 常用类

# 文件流