try-with-resource语句

1. **与Python中的with语句实现的功能差不多，都是省去了close方法。**
2. 语法：

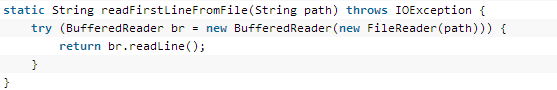
**try(需要关闭的资源的创建){**

**可能出现异常的语句。**

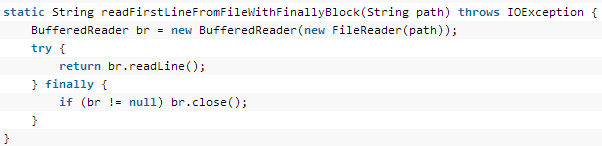
**}**

**编译器最后自动执行：判段对象是否为空，若不是空，则执行close方法。**

编绎器自动在try-with-resources后面增加了判断对象是否为null，如果不为null，则调用close()函数的的字节码。



**等价于：**



1. 拓展：

只有实现了java.lang.AutoCloseable接口，或者java.io.Closable（实际上继随自java.lang.AutoCloseable）接口的对象，才会自动调用其close()函数。

有点不同的是java.io.Closable要求一实现者保证close函数可以被重复调用。而AutoCloseable的close()函数则不要求是幂等的。

使用了try-with-resources语句之后，有可能会出现两个异常，一个是try块里的异常，一个是调用close函数里抛出的异常。

当然，平时我们写代码时，没有关注到。一般都是再抛出close函数里的异常，前面的异常被丢弃了。

如果在调用close函数时出现异常，那么前面的异常就被称为Suppressed Exceptions，因此Throwable还有个addSuppressed函数可以把它们保存起来，当用户捕捉到close里抛出的异常时，就可以调用Throwable.getSuppressed函数来取出close之前的异常了。