# 第六章上机报告

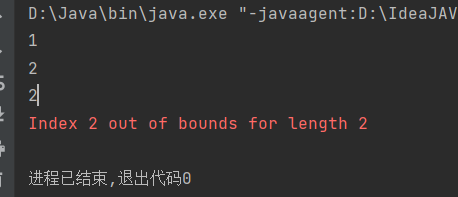
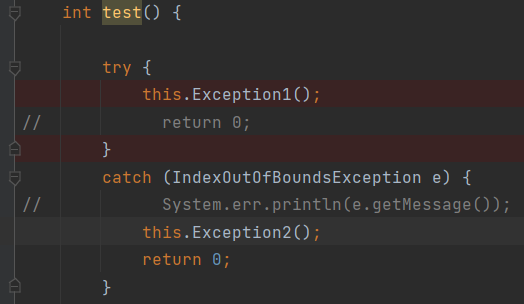
#### 1（a）

1. 在try中抛出 IndexOutOfBound异常的时间，在try语句块中被捕获，然后再与try后的catch语句块中的异常类型进行匹配，如果匹配到就会执行当前catch语句块中的Java语句，如果没有匹配的异常类型，就会出现语法错误。需要注意到的是，可以匹配到抛出异常类型的父类。

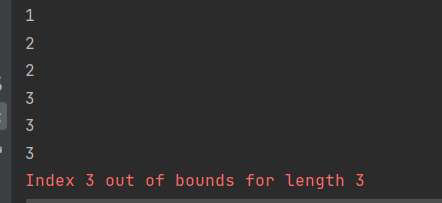




1. 在catch中抛出IndexOutOfBound异常，如果在try中没有匹配到当前catch语句类型的异常，就你不会执行当前的catch语句。如果匹配到之后按照语句块的顺序执行语句，先执行完finall。在当前语句块中抛出异常后依照方法的调用关系继续向上抛出，直到被处理，到main（）还未被处理的时间报错。
2. 在final中抛出的异常给调用方，没有被处理的话报错。
3. 如果在try、catch中都有异常，则编译器将catch中的异常抛出给调用方将其处理（一直向上抛出）。未被处理将报错。



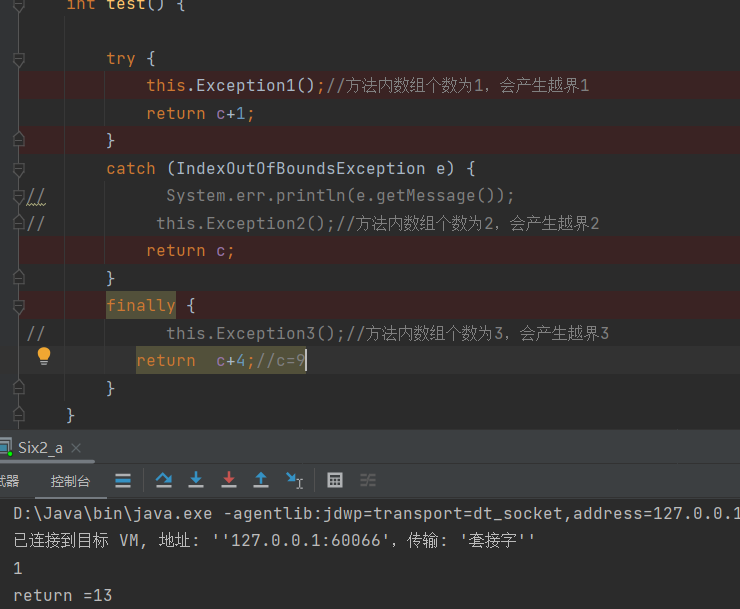
1. 如果在try、catch、final中都有异常，将final的异常抛出，讲其他两个抑制！



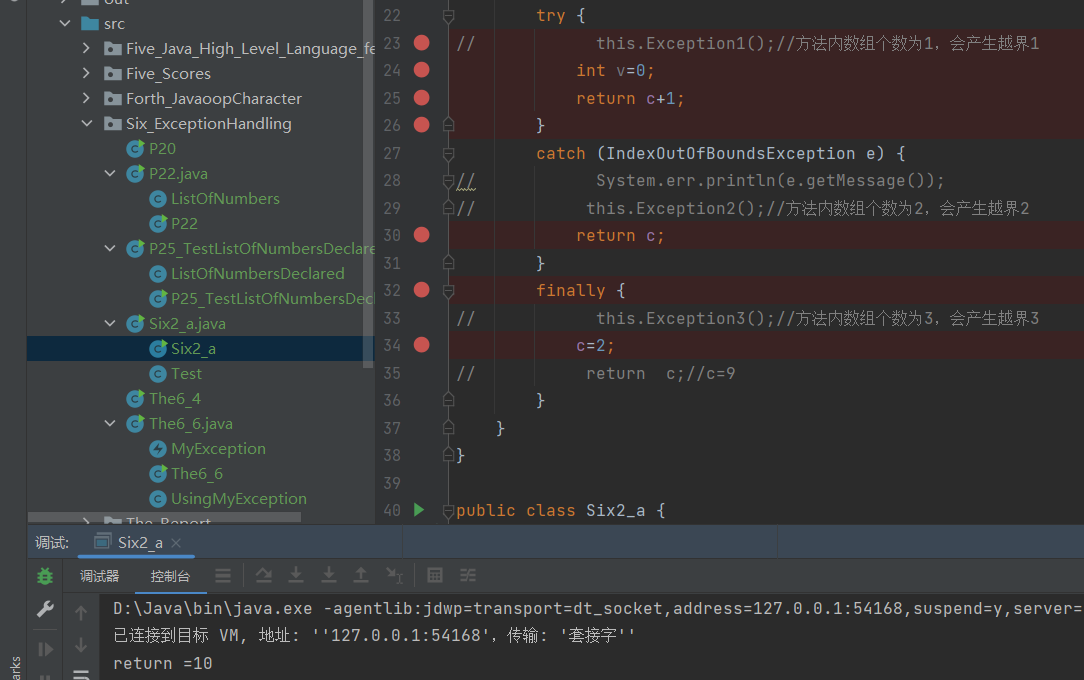
1. 经过测试后，综上：在try、catch、finally中同时有多个异常，则按照上述排列顺序最后的产生异常的抛出，将其他异常抑制！

#### 1（b）

1. 在这三个中都有使用return语句返回数值的时间，顺序是和上述结果相同的即同时在三个中都有返回值得时间在try抛出异常后不在执行后面的语句，后面的不在有异常的时间按照catch、finally的次序返回最滞后的一个。



1. 在try中捕获到异之后天专职catch中，catch返回将其已经绑定c=9给返回，然后再执行c=2执行后再一次返回到catch中此时的c已经变成了2，但是由于在catch中已经将其绑定，所以最后返回的数值还是9;无论是在try中还是在catch中返回一个整数在finally中修改的时间会改变其值，但是在前两个返回的时间已经绑定了数值，即使在finally系应该后又跳转值相应的return语句，返回的数值仍然是刚开始绑定的。



#### 1（b）

1. 免检异常及必检异常在不进行捕获的时间有不相同的结果：在免检异常中编译会通过，会将程序执行完毕，在遇到异常的时间报错；而必检异常中在编译的时间就会报错，编译不通过。
2. 免检异常和必检异常在进行捕获的时间基本上是相同的结果：对于必检异常必须进行捕获处理或者将其继续抛出。对于免检异常捕获处理得到正确的程序结果。

#### 2（a）

创建DateException异常类，创建整数类型Date，在Test中测试样例。

1. DateException类中，有成员变量reason，还有返回reason的方法。
2. 在Date类的数据中拥有getInstance的静态方法来创建一个Date的实例，可能会抛出异常（不是一个正确的数字）。

