

一，异常：在java中，将程序执行中发生的不正常的情况称为异常（开发中的语法错误和逻辑错误不

是异常）

分类：error：java虚拟机无法解决的严重问题，如系统内部错误，资源耗尽，一般不编写针对性的代

码进行处理

exception：其他因编写错误或偶然的外在因素导致的一般性问题，可以使用针对性的代码进

行处理（空指针异常，试图访问不存在的文件，网络连接中断等）

java.lang.Throwable

Error:错误，程序中不进行处理

Exception:异常，要求在编写程序的时候，就要考虑异常的处理

处理方法：①遇到错误就终止程序的运行

②由程序员在写程序的时候，就考虑到错误的检测、错误信息的提示，以及错误的处理

捕获错误最理想的是在编译期间，但有的错误只有在运行的时候才会发生，比如除零异

常和数组下标越界

. java文件-----编译----->. class(字节码文件)----运行----->执行结果

编译时异常：在编译的时候出现的异常（执行javac.exe命令时，出现异常）

运行时异常：在运行期间出现的异常（执行java.exe命令时出现的异常）

①ArrayIndexOutOfBoundsException数组下标越界

②ArithmeticException算术异常

③ClassCastException类型转换异常

④NullPointerException空指针异常

当一个程序执行过程中，如果出现异常，那么异常之后的代码不在执行

```

18 public class TestException {
19     //编译时异常
20     @Test
21     public void test6(){
22         FileInputStream fis = new FileInputStream(new File("hello.txt"));
23         int b;
24         while((b = fis.read()) != -1){
25             System.out.println((char)b);
26         }
27         fis.close();
28     }
}

```

二，异常处理：java提供的是异常的抓抛模型

1，“抛”：当我们执行代码时，一旦出现异常，就会在异常的代码块处生成一个对应的异常类型的对

象，并将此对象抛出。（自动抛出/手动抛出 `throw new Exception`）

①一旦抛出此异常的对象，那么程序就会终止执行

②此异常类的对象抛给方法的调用者

2，“抓”：抓住上一步抛出来的异常类的对象，采用异常处理的方式进行抓

java提供了两种方式用来处理一个异常类的对象

①try{ 可能出现异常的代码 }catch(Exception e1){ 处理方式 }catch(Exception e2){ 处理方式 }
 1 } finally{ 一定要执行的代码 }

注：① （`e.printStackTrace`打印错误信息 `e.getMessage()`得到错误原因）try内声明的变量，类似

于局部变量，出了try{} 语句，就不能被调用

②finally是可选的

③catch语句内是对异常对象的处理

④可以有多个catch语句，try中抛出的异常类对象从上往下去匹配catch中异常类的类型，一旦满

足就执行catch中的代码，执行完，就跳出其后的多个catch语句

⑤如果异常处理了，其后的代码将继续执行

⑥若catch多个异常类型是并列关系，谁上谁下都可以

若catch中多个异常类型是包含关系，须将子类放在父类的上面，进行处理，否则会报错

⑦finally中存放的是一定会执行的代码，不管try中、catch中是否有异常未被处理，以及是否有

return语句，都会执行

⑧try catch是可以相互嵌套的

三，对于运行时异常，可以不显示的进行处理

对于编译时异常，必须显示的进行处理