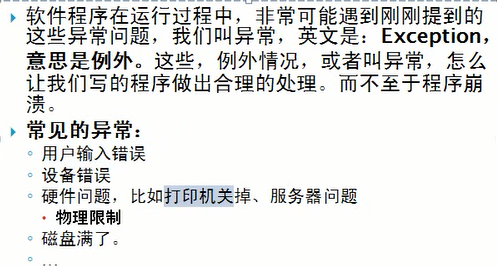
1.异常



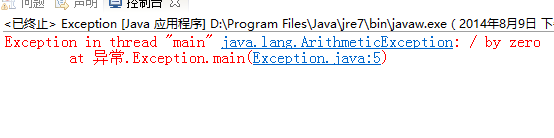
例如 ：

**public** **static** **void** main(String args[]){

**int** i = 1/0;

}

这就会一个异常



2》异常的分类error 不用管的

Exception 是必须管的 这里主要是看它

1》》》自动捕获的异常 Thread.sleep（）时间停顿几秒

**try** {

Thread.*sleep*(10000);

// checked的异常\*(自动捕获的异常)

} **catch** (InterruptedException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

}

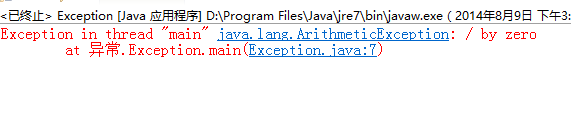
System.*out*.println("zhangyujin");

2》》》必须手动捕获的异常

例如

int a = 3/0;

执行程序之后出现的结果是



此时程序不可以执行了，我们必须手动处理了

这里我是使用了一个if条件来讨论的一下，这就是一个手动处理异常，

**int** b = 1;

**if**(b == 0){

**int** a = 3/b;

}

**else**{

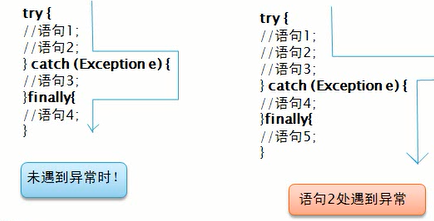
System.*out*.print(" ");

System.*out*.println("被除数不可以是 0 ");

}

**3.异常的处理方法**

**1》捕获异常**



解释上面的图形：第一个：---》当我们的语句try语句中1,2没有异常的时候，直接进行了finally后面的语句4.

第二个图----》当我们的try中有一个语句出现异常的时候 会跳过后面点语句，catch捕获异常，执行语句进行处理，最后执行finally后面的最终的语句，不会再去执行try中没有执行的语句，意思就是直接跳过了；

2.至少带有一个catch或者finally ，一个try可以对应多个catch

catch ：可以有多个

finally 1.: 我们有些东西不管是不是发生异常，都需要程序去执行，所以我们把它写到finally后面

2.一般不用return语句

2》真正的自己写处理异常的方法

1》例如下面的语句就会出现异常，因为不知道该文件是存在的还是不存在的。

FileReader read = **new** FileReader("F:/新建文件夹/Xin.java");

2》现在我们先使用try将它包围起来并且写入catch

**try** {

FileReader read = **new** FileReader("F:/新建文件夹/Xin.java");

} **catch** () {

}

3》此时我们鼠标移动到 FileReader 类上面的时候会抛出东西 如下图



显示的抛出是 [FileNotFoundException](eclipse-javadoc:%E2%98%82=%E5%BC%82%E5%B8%B8/D:%5C/Program%20Files%5C/Java%5C/jre7%5C/lib%5C/rt.jar%3Cjava.io(FileReader.class%E2%98%83FileReader~FileReader~Ljava.lang.String;%E2%98%82FileNotFoundException) 所以此时我们需要将捕捉到的写入catch()中，一般情况下的变量的名字是 e所以是

**try** {

FileReader read = **new** FileReader("F:/新建文件夹/Xin.java");

} **catch** ( FileNotFoundException e) {

}

4》现在我们 需要对异常进行处理

e.printStackTrace();

5》此时我们还想 往下面写，

**try** {

FileReader read = **new** FileReader("F:/新建文件夹/Xin.java");

**char** a = (**char**)read.read(); //这个是 int类型 不能够强制转化

这个事读取一个单独的字符

} **catch** ( FileNotFoundException e) {

e.printStackTrace(); **///这个就是对错误进行处理**

}

**catch**(IOException e){

e.printStackTrace();

}

6》这里的两个 catch不可以调换（上面的是子类，下面是父类）越是父类，越放下面（）

7》程序完成之后，要对资源 进行关闭

使用finally

}**finally**{

read.close();

}

8》这样的话只是对 try中的进行关闭，现在我们需要对整个程序进行关闭

所以需要将变量read放到try的外面

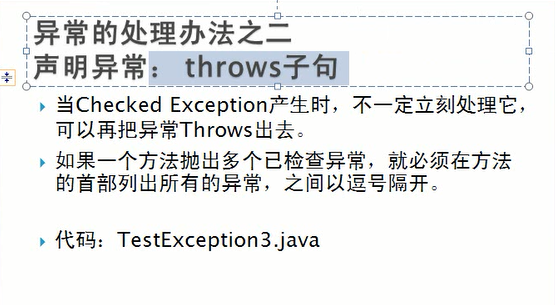
FileReader read = **null**;

**try** {

read = **new** FileReader("F:/新建文件夹/Xin.java");

9》但是后面的read.close();出现了异常，所以还要进行处理（同我们之前的东西进行异常处理）

2.异常的处理方法二 throws 抛出



2.现在进行一个例子

**public** **static** **void** main(String args[]){

}

**public** String OpenFile(){

FileReader a = **new** FileReader("F:/新建文件夹");

**return** "";

}

1》鼠标移动到FileReader 上面的时候出现了两种解决方案，我们现在需要的是Throw异常 选择他，就会在方法的前面进行抛出

**public** String OpenFile() **throws** FileNotFoundException{

FileReader a = **new** FileReader("F:/新建文件夹");

**return** "";

}

2》抛出处理之后，谁调用它，谁就会进行处理，但是如果调用的时候也可以抛出，但是不会建议用的，有时候我们测试的时候是可以的

3》抛出多个异常的时候，我们可以讲异常放到前一个异常的后面，用逗号隔开

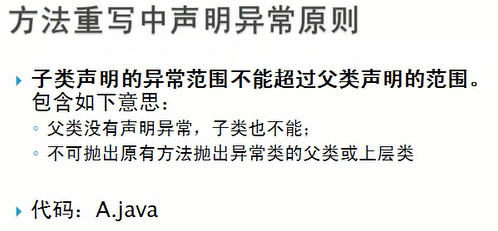
**public** String OpenFile() **throws** FileNotFoundException, IOException {

FileReader a = **new** FileReader("F:/新建文件夹");

**char** b = (**char**)a.read();

**return** "";

**3.方法重写中的异常声明原则**



**public** **class** FangFachongxieyichang {

**public** **void** yichang() **throws** FileNotFoundException{

}

}

**class** a **extends** FangFachongxieyichang{

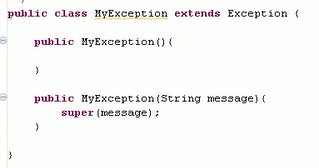
**public** **void** yichang() **throws** FileNotFoundException,IOException{

} //这个是错误的 不可以抛出父类 声明异常的类的 上层类 用一句土话理解 **不可以基因突变不可以超过父类异常**

}

**4.异常处理方法之 手动抛出异常 throw 用的不多了解即可**

5.自定义的异常



6.