

实验内容

- 计划课时：1.5*45
- 异常的使用
- Exception、RuntimeException
- throw, throws 关键字
- finally 关键字
- 自定义异常类
- 使用 Eclipse 进行调试
- 单元测试-junit4 的使用

题目 1：使用异常机制处理程序错误(PTA)

1.1 为如下代码加入适当的异常处理，使得该程序当输入的字符串不为整数字符串的时候，可以重新输入

```
public static void main(String[] args) {  
    int[] x = new int[3];  
    Scanner inputScan = new Scanner(System.in);  
    for(int i = 0; i < x.length; i++){  
        System.out.println("Please input the "+i+" integer:");  
        String inputInt = inputScan.nextLine();  
        x[i] = Integer.parseInt(inputInt);  
    }  
    System.out.println(Arrays.toString(x));  
}
```

1.2 为以前的“图形程序设计”实验中的“A+B=?”的GUI程序加入异常处理代码，使得该GUI程序更健壮。当文本框中输入的文本不为数值格式，也能进行处理，如弹出对话框提示或者焦点失去时将文本框置为0.0。

题目 2：常见异常及异常的基本用法(PTA

7.1)

a.编写代码产生下列几个常见的异常，并截图说明什么情况下会产生下面几个异常？这几个异常属于什么类型的异常？

- ArrayIndexOutOfBoundsException、NullPointerException、ClassCastException

b.在 Eclipse 中关联 java 类库的源代码

b.1 查看 `Integer.parseInt("123")` 方法的源代码：使用 `ctrl+鼠标单击`。这时候会提示没有相应的源代码。按照提示按 `F3`，将 `jdk` 目录下的 `src.zip` 关联起来，这时候就可查看到相关的源代码了。回答：当传入该方法的参数为 `null` 时会抛出什么异常？该异常是什么类型的异常？找到并复制相关源代码。

b.2 查看代码 `new Random().nextInt(10)` 的 `nextInt` 源代码，回答：什么情况下会抛出异常？该异常是什么类型的异常？需要使用 `try-catch` 进行处理吗？复制相关源代码。

c.改正下列代码，让其可正常运行。注意：里面有多个方法均可能抛出异常：

```
byte[] content = null;
FileInputStream fis = new FileInputStream("testfis.txt");
int bytesAvailabe = fis.available(); // 获得该文件的字节数
if(bytesAvailabe>0){
    content = new byte[bytesAvailabe]; // 创建可容纳文件大小的数组
    fis.read(content); // 将文件内容读入数组
}
System.out.println(Arrays.toString(content)); // 打印数组内容
```

提示：注意在 `finally` 块中对资源(`FileInputStream`) 进行释放。

在 `workspace` 目录相应的项目目录下，新建文件 `testfis.txt` 文件，里面有三行，分别为：

```
123
abc
ABC
```

运行上述代码，分析为什么是这样的输出？

上述代码包含多种类型异常，因此应使用多个 `catch` 块捕获异常。也可以只是用一个 `catch` 块，如 `catch(Exception e)`，试分析为什么？并分析，和前面的处理方式相比，使用 `catch(Exception e)` 有什么优缺点？

进阶：使用 Java7 中的 `try-with-resources` 来改写上述代码实现自动关闭资源？能否使用多重捕获改写上述代码？为什么？

题目 3: 读取文件并组装对象(使用 `try..catch` 让程序更健壮) (基本考核)

- “身份证号.txt”文件中包含若干用户的信息，为其创建对象 `User`，拥有属性 `name(String)`, `id(String)`, `gender(String)`, `age(int)`, `address(String)`。观察“身份证号.txt”，会发现个别行的数据有缺失。直接运行 `ReadFileUsingScanner.java`，会发生异常，是什么异常？捕获异常时，将错误的信息按照“**行号:出错原因: 该行内容**”格式输出。如何修改该程序？
- 试编写一个程序，将文件中的信息读出，组装成 `User` 对象，然后放入一个 `List`。注意：这里需要使用 `try..catch` 处理程序中可能出现的异常。当发生错误时，提示用户第几行发生错误？发生了什么错误？该行的内容是什么？以帮助用户定位错误。
- 对 `List` 中的 `Student` 按照年龄进行排序输出。
- 文件读取参考代码：`ReadFileUsingScanner.java`

题目 4: throw 与 throws 与编写异常(PTA 7.3)

1.创建 Student 类

■ 属性:

private String name

private int score

■ 方法:

各个属性的 setter/getter 方法

public int addScore(int score)

public void setName(String name) //为该方法声明异常 throws Exception

■ 在 addScore 的时候可能抛出异常, 比如加分后分数超过 100 或者小于 0。

■ 在 setName 的时候也可能抛出异常, 比如设置 name 为空字符串, 姓名首字母为数字。

提示: 使用 IllegalArgumentException。

注意: 测试的时候, 产生一些异常, 希望调用者通过查看异常得知错误发生的原因。该异常中一定要告诉使用者到底发生了什么错误。

■ **选做:** 编写 IllegalScoreException, 复制 addScore 方法为 addScore1 方法, 该方法抛出自定义的异常。通过该异常希望可以告诉调用者错误发生的原因。

■ **选做:** 以前编写的求开根号程序, 应该在哪里抛出异常呢? 抛出什么异常?

思考: 自定义的异常可以继承自 Exception 或者 RuntimeException。如果希望代码中一定要用 try...catch 捕获, 该异常应该继承哪个类。如果不需要 try..catch 捕获, 该异常需要继承哪个类? 你认为这里的自定义异常应该继承 Exception 还是 RuntimeException, 为什么?

参考资料:

1. 题目 2 中 代码 new Random().nextInt(10)的 nextInt 源代码,
2. ppt 中编写自己的异常章节。

题目 5: 使用 Eclipse 进行调试(基本加分)

参考资料: 调试相关视频

1. 使用 ReadFileUsingScanner.java 读取 “身份证号.txt”文件时, 会产生异常, 尝试加入异常处理机制。捕获异常时, 将错误的信息按照“**行号:出错原因: 该行内容**”格式输出。进一步, 读取 “蓝山兰” 这行的时候并不会出错, 尝试使用调试技巧, 找出原因所在。

2. 改写 “基础代码” 目录中 TestDebug.java 代码, 查看当 i=5 时数组中值的排列。

拓展: 假设某个循环可能有上万次, 想在第 9861 次, 断点才起作用, 怎么设置? (提示: 搜索条件断点、HitCount)。条件断点: 右键点击断点, 选择 breakpoint properties。

3. 在 sort 方法中的 swap 加断点, 使用 Step Into 进入 swap 函数, 观察进行交换后数组中的值, 使用 Drop To Frame 功能可以反复调试 swap 方法内的代码。

4. 条件断点。

注意：使用 Eclipse 的 debug 功能跟踪改编后的代码运行。学会使用 F5,F6,F7 等快捷键，详见："Eclipse 调试常用快捷键.txt"

选做：3.使用 Eclipse 调试技巧，调试 MessageBoard.zip 中的系统直至可以正常运行。(特别加分)

选做：题目 6：使用 JUnit5 进行单元测试

- 1.我们现在需要测试自己编写的冒泡排序和系统类库的 Arrays.sort()的性能。
- 2.编写一个方法 `public static int[] getRandomArray(int amount, int range)`，随机生成 amount=10000000 个随机数的整型数组，随机数的范围 range 从 0 到 100000000。
- 3.为代码新建 JUnit Test Case
4. JUnit Test Case 大概需要哪几个方法呢？按怎样的步骤进行测试呢？
- 5.记录每个步骤运行所花费的时间。

参见：“基础代码”目录中 BigDataSortTest.java 源文件。

参考资料：

[junit5 入门系列教程](<https://blog.csdn.net/ryo1060732496/article/details/80812102>)