

# Java程序设计编程作业-4

## 作业说明

请按照本文档的题目要求和代码框架完成编程作业。代码框架的示例结构如下所示，在 `src` 文件夹目录下，每一个题目对应一个文件夹(package)。在package内部，通常题目要求实现若干个类(class)，分别对应一个java源代码文件（未完整实现）。此外，每个题目有一个包含main函数的公有类 `Test.java`，功能是对题目中待实现的类进行测试。最后，在 `src` 文件夹目录下有一个 `testutil` 包，是测试时所用到的工具，同学无需阅读或更改。

```
1  └─hw1
2      └─README.pdf
3
4      └─src
5          └─problem1
6              └─Class1.java (待实现)
7              └─Class2.java (待实现)
8              └─Test.java
9          └─problem2
10             └─Class1.java (待实现)
11             └─Test.java
12
13             ...
14
15         └─testutil
```

同学们完成作业的建议步骤如下：

1. 按照文档，在代码框架中完成题目所要求的功能类的完整实现。
2. 运行每个题目的 `Test` 类，根据测试结果进行调试。
3. **严格按照以下文件结构和命名提交源代码文件，并打包成 zip 格式上传，文件命名为 学号\_姓名.zip。**

```
1  └─2020123456_张三 (学号_姓名)
2      └─report.pdf/txt/md (实验报告，可选)
3
4      └─src
5          └─problem1 (抽象名，根据实际题名决定，小写)
6              └─Class1.java (抽象名，根据实际类名决定，首字母大写)
7              └─Class2.java
8              └─... (其他java源代码)
9
10         └─problem2
11             └─Class1.java
12             └─... (其他java源代码)
13
14         ...
```

请注意：

- 编程作业为个人作业，请独立完成。如果提交的代码参考了资料或个人，请在report中标注来源，否则视为抄

袭。

- 编程作业采用自动化评判，因此不按照规定格式提交作业可能导致错误。
- 请严格按照代码框架和文档要求对成员变量、类、函数、文件命名，函数参数类型、名称、顺序和函数返回值类型请严格按照文档实现，此外类、成员变量和函数访问权限请按照文档要求指定。**Java严格区分大小写，不符合要求的命名会导致错误。**
- 请不要更改代码框架的文件结构和命名，请不要删除代码框架中java源文件开头的包声明 `package ...;`。
- 除了题目要求外，可以根据需要实现其他类、函数和成员变量。可以提交其他新增的源代码文件，在新增的源代码文件开头必须进行包声明 `package ...;`。
- `Test` 类对应的源代码文件以及 `testutil` 包可以不提交，同时同学们也可以不改写。无论它们是否被提交或改写，评判过程中会对这两类文件重新覆盖。
- 本次实验框架包含一个 `data` 目录，内容为示例测试数据，该目录不需要提交。
- `Test` 类包含了测试程序的一些基本测例。无特殊说明的情况下，作业框架中提供的测例是评判时测例的子集。最终的评判分数为同学所提交代码在全体测例中通过的比例。同学们可以在 `Test` 类中根据需要补充测例，进行测试。**本次作业提供的测例是评判时的约40%。**
- 实验报告是可选的，内容为除了源代码外额外需要说明的内容，例如参考的资料，复杂题目的实现思路等。实验报告不影响本次评判。
- 文档和实验框架的错误、歧义和bug可联系助教 [wujy22@mails.tsinghua.edu.cn](mailto:wujy22@mails.tsinghua.edu.cn)。
- Java API官方在线文档: <https://docs.oracle.com/en/java/javase/19/docs/api/>
- 本次作业要求实现的所有类和方法均为公有的。

## Problem 1: ExceptionTest (45')

在包 `exceptiontest` 中创建一个进行异常处理测试的 `ExceptionTest` 类，以及2个异常类 `XLessThanZeroException`，`XGreaterThanOneException`。该包所要求的接口见以下伪代码。

1. `ExceptionTest` 类是一个公有类，使用默认的无参构造函数。
2. 读取一个合法的数字。

在 `ExceptionTest` 类中实现公有成员函数 `getFirstValidInt`，该函数中需要调用 `Test` 类的 `readInt` 函数（已经实现，无需更改）获取一个合法的整数，然后将其返回。`readInt` 函数的返回值是一个随机 `int` 类型的整数，但是有一定概率抛出一个 `NumberFormatException` 异常。下面是 `readInt` 函数的接口：

```
1 // Test.java
2 /**
3  * 读取一个随机整数，以一定概率抛出异常
4  * @return int类型的整数
5  * @throws NumberFormatException 未读取到合法的整数，抛出异常
6  */
7 public static int readInt() throws NumberFormatException;
```

关于 `readInt` 函数的详细实现请同学们参考 `Test.java`。

同学们需要实现 `ExceptionTest` 类的 `getFirstValidInt` 函数，功能为不断调用 `readInt`，忽略产生异常的操作，直到读到第一个合法的 `int` 类型整数，将其返回。

```

1 // ExceptionTest.java
2 /** 反复调用readInt并处理其抛出的异常，直到得到第一个合法整数
3  * @return 从readInt中得到的合法整数
4  */
5 public int getFirstValidInt();

```

3. 解  $x$  有定义域的方程  $e^x = y$ 。

```

1 // ExceptionTest.java
2 /** 求解方程 $e^x=y$ .定义域为 $0 \leq x \leq 1$ .
3  * @param y
4  * @return x
5  * @throws XLessThanZeroException, 当 $x < 0$ 时抛出该异常
6  * @throws XGreaterThanOneException, 当 $x > 1$ 时抛出该异常
7  */
8 public double getX(double y);

```

其中，异常类 `XLessThanZeroException` 继承于异常基类 `Exception`，包含一个构造函数：

```

1 // XLessThanZeroException.java
2 /**
3  * @param y 方程 $e^x=y$ 中的等式右边.
4  */
5 public XLessThanZeroException(double y);

```

该子类还包含一个 成员函数 `getInfo`:

```

1 // XLessThanZeroException.java
2 /**
3  * @return 字符串:"y is too small", 其中y是变量, 例如"0.2 is too small".
4  */
5 public String getInfo();

```

异常类 `XGreaterThanOneException` 继承于异常基类 `Exception`，使用默认构造函数，并实现成员函数 `getInfo`:

```

1 // XGreaterThanOneException.java
2 /**
3  * @return 字符串:"Too large".
4  */
5 public String getInfo();

```

在 `Test` 类中有一个 `solve` 函数（已经实现，无需更改），调用上述待实现的接口完成了方程的完整求解。该函数用以测试同学实现的接口是否正确。

```

1
2  /**
3  * 求解代数方程 $e^x=y$ ,  $x$ 的定义域 $[0,1]$ 
4  * @param y double 参数
5  * @return String 方程的解,保留3位小数. 若 $x<0$ , 返回"y is too small", 例如"0.2 is too
    small"; 若 $x>0$ ,返回"Too large".
6  */
7  public static String solve(double y);

```

数据范围：

- getX函数:  $y > 0$ .

## Problem 2: DataChooser (20')

在包 `datachooser` 中实现公有的功能类 `DataChooser`，用以处理csv文件。

csv文件是一种表示表格数据的文本文件，通过英文逗号将每一列隔开，并通过换行符将每一行隔开，其中文件的第一行表示列名。`data/data.csv` 是一个示例的csv文件，是一个记录学生信息的表格：

```

1  Name,Gender,Age,Department,Student ID,Province,Score
2  Three Zhang,Male,22,Computer Science,2000123456,Beijing,90
3  Four Li,Female,20,Electronic Engineering,2002023567,Hebei,91
4  Five Wang,Male,21,Software Engineering,2001135798,Henan,88
5  Six Six Six,Male,30,Computer Science,1999999999,Tibet,94
6

```

实现一个DataChooser的方法choose，取出表格中"Department"一列，并输出为一个csv文件。例如示例文件的输出应为：

```

1  Department
2  Computer Science
3  Electronic Engineering
4  Software Engineering
5  Computer Science
6

```

示例输出文件请见 `data/data_out.csv`，供同学们比对。

若输入文件中只有标题行，没有数据，那么输出文件也只需要输出2行：

```

1  Department
2

```

`choose` 函数的接口如下：

```

1  /**
2   * @param inputFile 输入csv文件的绝对路径
3   * @param outputFile 输出csv文件的绝对路径
4   */
5   public static void choose(String inputFile, String outputFile);

```

请注意：

- 文本文件中的换行符在 Windows 下是“\r\n”，而在Linux和MacOS系统下是“\n”。要求实现方法需要对不同系统下的换行符具有通用性，亦即输入csv文件的换行符可能为“\r\n”或“\n”。但输出csv文件的换行符不作要求。
- 输入和输出的csv文件末均有且仅有一个空行。
- 本题目提供了一个示例的csv文件，但实际评测中的csv文件内容、列数、列名、行数都可能示例文件不一致。也就是说，Department不一定是第4列。但是保证Department这一列一定存在。
- 本题不提供评判时使用的Test类，且示例文件 data/data.csv 不是唯一测例，请同学们仿照其他题目的Test类自行编写测试，自行测试编写的Test类不需要提交：
  - 调用DataChooser.choose方法，输入 data/data.csv，输出文件与 data/data\_out.csv 比对。请注意，输出文件应该是6行，最后一行是空行。
  - 编写其他输入文件，进一步测试。

数据范围：

- 保证输入csv文件格式正确，并且包含"Department"一列。
- csv文件的行数 $L$ :  $0 \leq L \leq 50$ 。
- csv文件每行的总字符数不超过100。

## Problem 3: FileSearcher (35')

在包 filesearcher 中实现功能类 FileSearcher。

创建一个search方法，用以在给定的自然语言文本文件中搜索某个关键词出现的总次数。关键词是一个英文单词，例如"go"。文件中和关键词匹配的单词不区分大小写，例如GO、Go、gO、go 都算出现一次。同时关键词可能包含大写或小写字母。但是单词是需要完整匹配的，也就是说 google 这样的单词不算出现了go。除此之外，What's 这样的字符串中算出现了 what，但没有出现 is，同时只有"what"会作为搜索的关键词而"what's"不会。Three-year-old中同时出现了three、year和 old。总结来说，单词指的是连续出现的大/小写字母，且单词的前一个和后一个字符（如果存在的话）都必须不是大/小写字母。

```

1  /**
2   * @param inputFile 输入文本文件的绝对路径
3   * @param keyword 搜索的关键词
4   * @return 关键词出现的次数
5   */
6   public static int search(String inputFile, String keyword);

```

本题目提供了一个示例文件，见 data/dialog.txt，实际评测中有更多测试用例。请同学们尽可能多测试一些单词，并配合文本处理器的查找功能核对答案。

请注意：

- 文本文件中的换行符在 Windows 下是“\r\n”，而在Linux和MacOS系统下是“\n”。要求实现方法需要对不同系统下的换行符具有通用性，亦即输入文件的换行符可能为“\r\n”或“\n”。
- 本题提供示例的Test类，但是使用时需要将inputFile改为 `data/dialog.txt` 在同学系统中的绝对路径。

```
1 String inputFile = "改为data/dialog.txt在您系统中的绝对路径";
```

- 示例测试文件为gbk编码格式，请同学们注意设置相关参数以避免读取错误。