

2020 年夏季 Java 小学期

Homework3

问答题

【约定】

- 本题要求将答案写到答题卡.xlsx 的作答区域中。
 - 问题保证答题卡.xlsx 文档没有使用任何宏。
- 提交的答卷格式为.xlsx。如果不支持此格式，可以将文件转为答题卡.xls 或答题卡.csv。
 - 如果文件被保存为.csv 格式，请确保编码被设置为 UTF-8。但不建议在.csv 格式下作答，因为会丢失选择题的数据验证。
- 提交结果为半自动批阅，请不要私自修改非作答区域的内容，否则可能出现答案无法读取。
 - 可以修改行宽、列宽、字体、字号，但不要增加 sheet 或改变答案的行号、列号，也不可以合并单元格。
 - 建议直接使用默认字体，而不要使用自定义字体，自定义字体可能使用特殊编码导致内容无法读取。
- 回答多选题时请从左到右按照编号顺序填写选项，一个区域一个选项，多余区域保留空白选项。

【第一题】补充信息

在 Java 中，对数据通信的操作进行了封装，并提供了四类基本的用来进行输入输出的对象。分别是：

- 字节输入流：__ (1) __
- 字节输出流：__ (2) __
- 字符输入流：__ (3) __，
- 字符输出流：__ (4) __，

选项：

- A. java.io.OutputStream
- B. java.io.InputStream
- C. java.io.PrintStream
- D. java.io.Writer
- E. java.io.Reader
- F. java.lang.System
- G. java.lang.String
- H. java.util.Scanner

【第二题】选择

(1) 下列哪些类型的实例是 Java 中的可抛出对象？

- A. java.lang.Error
 - B. java.lang.ClassNotFoundException
 - C. java.lang.Exception
 - D. java.io.IOException
 - E. java.lang.Void
- (2) 下列哪些类型的实例在 Java 中可以不声明抛出，也不强制捕获？
- A. java.io.EOFException
 - B. java.lang.AssertionError
 - C. java.lang.ArrayIndexOutOfBoundsException
 - D. java.lang.NullPointerException
 - E. java.lang.Throwable
- (3) 下列哪些方法可以读入文本文件 “input.txt” 的内容？
- ```
import java.io.*;
import java.util.*;
```
- A. new Scanner("input.txt").next()
  - B. new FileInputStream(new File("input.txt")).read()
  - C. new FileReader("input.txt").read()
  - D. new Scanner(new File("input.txt")).next()
  - E. new File("input.txt").list()
- (4) 下列哪些方法可以把一个字节/字符写入文本文件 “output.txt” ？
- ```
import java.io.*;
import java.util.*;
```
- A. new FileWriter("output.txt").write(0)
 - B. new PrintStream("output.txt").write(0)
 - C. new FileOutputStream("output.txt").write(0)
 - D. System.out.append("output.txt")
 - E. System.out.print("output.txt")

编程题

【约定】

时间限制：5 s。

空间限制：1 GB。（VM 参数：-Xmx1GB）

程序入口类为 Main。

【第一题】文本统计

问题

输入一个文件的文件名，文件的编码格式，然后用指定编码读入文件。同时统计读入的文件中每个字符的出现次数（不包含文件中的换行）。

输入

若干行，每行包括一个文件名 file，一个编码 encoding。

输出

答案输出到文件 statistics.txt，编码为 UTF-8。

第一行输出读入的字符总数。

之后每行输出一个出现的字符，并加上字符出现的次数。行尾需要换行。

输出字符时按次数降序排序，次数相同时，按 char 类型的升序排序（即 Unicode 编码顺序）。

例子

输入：

```
hello-gbk.txt GBK
hello-utf8.txt UTF-8
```

文件 hello_gbk.txt (GBK):

```
你好，世界！
```

文件 hello_utf8.txt (UTF-8):

```
你好，世界！
```

输出 statistics.txt (UTF-8):

```
12
世 2
你 2
好 2
界 2
! 2
, 2
```

规模

文件数量不超过 10 个。

文件名不含空格，长度不超过 50。

文件名只包含 ASCII 字符，并且没有特殊含义。

提示

文件读入：

若用这种方式构造 Scanner：`Scanner in=new Scanner("fileio.in")`，则方法 `in.next()` 只会返回字符串 "fileio.in"。

正确方式请参考 Scanner 的 API 构造方法：`Scanner(File)`

文件输出：

在 Java 8 的 API 中，用流输出文件的快捷构造方法为：

`new PrintStream("fileio.out")` 及 `new PrintStream("fileio.out", "US-ASCII")`

或：

`new PrintWriter("fileio.out")` 及 `new PrintWriter("fileio.out", "US-ASCII")`

使用快捷方法不支持对文件追加输出，追加输出需要如下嵌套：

`PrintStream ps=new PrintStream(new FileOutputStream(new File("fileio.out"), true));`

提交文件

src/Main.java

【第二题】文件查找

问题

编写程序，列出 input/test/case 目录下的文件名称中包含某一字符串的所有文件，要求按照 ASCII 码表编号由小到大的顺序显示当前目录下的文件名，搜索时大小写不敏感。不需要考虑子目录下的文件，结果中不能包含目录。

数据使用标准输入输出读写。

例子

input/test/case 中有以下文件：

Abc
abc
babc
qqyy
zab

输入：

ab

输出：

Abc
abc
babc
zab

规模

文件数量不超过 1000 个。

文件名长度不超过 200。

文件名只包含 ASCII 字符，并且没有特殊含义。

提交文件

src/Main.java

【第三题】网安系统

问题

有一个网络系统近期遭到非法代码的攻击，为了保卫网络的安全，网络安全员请你帮他写一个网安系统。

在本题中，你需要在类 NetSecure 中实现一个网安系统，并且实现网安异常 NetException。网络系统会调用 NetSecure 的如下方法来检查一个网络数据包是否合法：

public static void check(String packet) **throws** NetException

如果发现了不合法的数据包，则你需要抛出异常，并设置异常信息，保证异常的打印结果与题中一致。

一个数据包是不合法的，当且仅当这个数据包的名字结尾是“.exe”（不区分大小写）。

输入

一行一个字符串，为数据包的名称。

输出

所有被捕获的异常，一行一个。

例子

输入：

helloworld.cpp

helloworld.exe

this_is_not_virus.eXe

输出：

NetException: network attack: helloworld.exe

NetException: network attack: this_is_not_virus.eXe

规模

数据包的内容只包含所有可打印 ASCII 字符。(字符编码 32~126)

数据包的长度为 0~20。

数据包的个数不超过 1000 个。

附加代码

src/NetException_template.java

src/NetSecure_template.java

src/Main.java

提交文件

src/NetException.java

src/NetSecure.java

【第四题】虚拟计算机

问题

假设有一台计算机，计算机有 N 个地址编号 $0 \sim N-1$ ，每个地址是一个 32 位有符号整数，地址的初始状态为 null，访问方式为 $\text{Mem}[i]$ ，你需要模拟以下指令：

1. “= a b num” 向量赋值指令，将数字 num 的值写入 $\text{Mem}[i]$ ， $i \in [a, b)$ 。
2. “+ a b c” 加法指令，将 $\text{Mem}[a] + \text{Mem}[b]$ 的结果写入 $\text{Mem}[c]$ 。
3. “- a b c” 减法指令，将 $\text{Mem}[a] - \text{Mem}[b]$ 的结果写入 $\text{Mem}[c]$ 。
4. “* a b c” 乘法指令，将 $\text{Mem}[a] * \text{Mem}[b]$ 的结果写入 $\text{Mem}[c]$ 。
5. “/ a b c” 除法指令，将 $\text{Mem}[a] / \text{Mem}[b]$ 的结果写入 $\text{Mem}[c]$ 。
6. “? a” 输出 $\text{Mem}[a]$ 的值，如果 $\text{Mem}[a]$ 没有被初始化，输出“null”。

如果指令中出现了算术溢出，使用默认方式截取计算结果。比如 $(1122334455) - (-2000000000) = (-1172632841)$ 。

你的程序需要负责处理异常：

- 如果某条数学运算指令读取了没有赋值过的内存，则捕获 `NullPointerException`，捕获

异常后跳过当前指令，并输出一行“Null Number”，然后继续处理后面的指令。

- 如果某条指令出现了被零除的操作，则捕获 ArithmeticException，捕获异常后跳过当前指令，并输出一行“Divided By Zero”，然后继续处理后面的指令。
- 如果某条指令访问了超出内存范围的地址，则捕获 ArrayIndexOutOfBoundsException，捕获异常后跳过当前指令，并输出一行“Illegal Address”，然后继续处理后面的指令。
- 对于向量赋值“= a b n”，本题不要求提前检查地址范围，而是模拟对 Mem[i], i=a, a+1, ..., b-1 进行依次赋值，若运行中出现异常，则处理异常后跳过当前指令，但已完成的赋值操作需要保留。
- 对于某操作 Mem[a] op Mem[b] -> Mem[c]，若 Mem[a] 与 Mem[b] 都存在异常，先抛出 Mem[a] 上的异常，异常包括空值内存与非法地址。

输入

第一行包含一个数字 N，为地址空间大小。

之后若干行，每行一条指令，指令如题目所述。

输出

每次遇到异常后，输出一行，为异常信息。

每次成功执行完“? a”指令，输出一行，包括一个有符号整数 Mem[a]。

例子

输入：

```
10
? 0
/ 0 0 0
? 10
= 0 10000 -2
? 0
/ 0 0 0
? 0
/ 0 9 0
? 0
/ 0 0 0
```

输出：

```
null
Null Number
Illegal Address
Illegal Address
-2
1
0
Divided By Zero
```

解释

执行过程为：

1. ? 0: 打印 Mem[0], 输出了“null”。
2. / 0 0 0: 令 Mem[0]=Mem[0]/Mem[0], 抛出 NullPointerException。
3. ? 10: 打印 Mem[10], 抛出 ArrayIndexOutOfBoundsException。
4. = 0 10000 -2: 令 Mem[0], Mem[1], ..., Mem[9999]=-2, 在写入 Mem[10]的时抛出 ArrayIndexOutOfBoundsException。
5. ? 0: 打印 Mem[0], 输出了“-2”。
6. / 0 0 0: 令 Mem[0]=Mem[0]/Mem[0]。
7. ? 0: 打印 Mem[0], 输出了“1”。
8. / 0 9 0: 令 Mem[0]=Mem[0]/Mem[9]。
9. ? 0: 打印 Mem[0], 输出了“0”。
10. / 0 0 0: 令 Mem[0]=Mem[0]/Mem[0], 抛出 ArithmeticException。

规模

数字 N 的范围是[0, 10000], 指令条数不超过 1 万条, 涉及的有效内存访问次数不超过 50 万次。

对于全部指令, a、b、c 和 num 会是任何合法的 32 位有符号整数。

提示

- 可以使用数组 Integer[]来模拟含有 null 状态的内存。
- 为了优先处理左操作数的异常, 可以依次取出操作数, 如:
 - `int operandA=Mem[a], operandB=Mem[b];`
 - 由于 Java 语言的自动装箱与拆箱机制, 会直接将 Integer 转为 int 类型并检查空引用。

提交文件

src/Main.java

【提交方法】

程序入口类名称为 Main, 包名为默认, 并放入相应的文件夹内。将解题过程写入实验报告中。关于程序的其他说明写入 readme.txt 中。并将以上文件打包为 zip 文件。提交至网络学堂。

文件打包如下:

zip 归档文件 (命名为: 学号_姓名_homework3.zip)

-- 答题卡.xlsx(.xls,.csv)

-- 实验报告.pdf (解题过程)

-- programming-1 (编程题 1)

-- src (目录内为提交的程序代码)

-- readme.txt (可选, 建议写)

-- programming-2 (编程题 2)

-- src (目录内为提交的程序代码)

-- readme.txt (可选, 建议写)

.....

【注意事项】

- 1、 代码要求自己写。代码提交后会进行查重，如果被认定为抄袭，课程将会被记为 0 分。
- 2、 按时提交。如有迟交，作业成绩在原有作业成绩基础上每迟交一天（不到一天按一天计）扣 10%。
- 3、 题目给分结果由最终测试决定。
不要相信简单的附加代码与样例结果，最终测试时，附加代码，输入、输出文件都会变得更严格。