

JAVA: 作业3

刘泓尊 2018011446 计84 liu-hz18@mails.tsinghua.edu.cn

1.文本统计

读入文件采用如下方式：

```
BufferedReader br = new BufferedReader(new InputStreamReader(new
FileInputStream(filename), encoding));
line = br.readLine();
```

实现了最基本的 Counter 功能，维护了一个 `TreeMap<Character, Integer>` 对象，用于统计对应字符出现的个数。

Counter 的更新过程为：

```
if (wordmap.containsKey(word)) {
    wordmap.put(word, wordmap.get(word) + 1);
} else {
    wordmap.put(word, 1);
}
```

上述更新过程涉及1次查询，1次get，1次set。因此单步操作时间复杂度 $O(3\log n)$

之后完成排序功能，使用 `ArrayList<>` 来实现排序，并自定义了 `Comparator`

```
mList = new ArrayList< Entry<Character, Integer> >(wordmap.entrySet());
Collections.sort(mList, new Comparator< Entry<Character, Integer> > () {
    public int compare(Entry<Character, Integer> left, Entry<Character, Integer>
right) {
        return right.getValue().compareTo(left.getValue());
    }
});
```

最后使用 `PrintStream ps = new PrintStream(filename, encoding);` 进行输出。

总时间复杂度 $O(n\log n)$

2.文件查找

使用 `File` 对象执行对文件夹的操作，使用 `listFiles()` 获得文件夹下所有的文件，如果该文件**不是文件夹**，则将其名字添加到 `ArrayList<String>` 中，最后排序并输出。

检索并筛选文件的代码如下：

```

File dir = new File(fileDir);
if (dir.isDirectory()) {
    File[] files = dir.listFiles();
    for (File aFileName: files) {
        if (aFileName.isFile() &&
aFileName.getName().toLowerCase().indexOf(substring) >= 0) {
            fileList.add(aFileName.getName());
        }
    }
}
}

```

3.网安系统

继承 `Exception` 实现 `NetException` 类，并重载 `toString()` 方法来便于输出信息。

```

public NetException(final String info) {
    mInfo = info;
}
public String toString() {
    return "NetException: network attack: " + this.mInfo;
}

```

在 `NetSecure` 类中，使用 `packet.lastIndexOf(".exe")` 方法判断是否包含 `.exe`，如果包含则取出后缀名，判断是否等于 `.exe`，如果是，就抛出异常。

```

int index = packet.lastIndexOf(".exe");
if (index >= 0) {
    String suffix = packet.substring(index, packet.length()).toLowerCase();
    if (suffix.equals(".exe")) {
        throw new NetException(packet);
    }
}
}

```

4.虚拟计算机

使用抛出异常的形式来中断执行过程中的不合法情况。主要框架如下：

```

VirtualComputer vc = new VirtualComputer(n);
while( cin.hasNextLine() ) {
    try {
        vc.operate(cin.nextLine());
    } catch (NullPointerException e) {
        System.out.println("Null Number");
    } catch (ArrayIndexOutOfBoundsException e) {
        System.out.println("Illegal Address");
    } catch (ArithmeticException e) {
        System.out.println("Divided By Zero");
    }
}
}

```

算数运算以 `add` 为例，使用先访存再计算再存储的方式：

```
private void add(final int a, final int b, final int c) throws
NullPointerException, ArrayIndexOutOfBoundsException {
    int optA = mem[a];
    int optB = mem[b];
    mem[c] = optA + optB;
}
```

除法运算有些特殊，需要**先判断除零再存储**：

```
private void div(final int a, final int b, final int c) throws
NullPointerException, ArrayIndexOutOfBoundsException, ArithmeticException {
    int optA = mem[a];
    int optB = mem[b];
    int result = optA / optB;
    mem[c] = result;
}
```

其中set函数顺序赋值，需要判断空值异常和越界异常：

```
private void set(final int a, final int b, final int num) throws
NullPointerException, ArrayIndexOutOfBoundsException {
    for (int i = a; i < b; i++) {
        mem[i] = num;
    }
}
```