JAVA：作业5

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

1.机器哲学家

本题按照题目描述编写代码即可。

在子线程中，首先获得`Com`实例。在初次进入循环时，调用`com.getNumber()`来获得消息编号，然后调用`wait()`方法等待消息更新。

线程被唤醒之后，重复调用`com.getNumber()`来更新自己的消息编号，并调用`star`方法。如果消息编号没有被更新，回到循环开始即可。

注意`com`需要加锁，防止竞争条件。

主要代码如下：

```
Com com = Com.getInstance();
Integer msgNumber;
Integer tempNumber;
while (true){
    synchronized (com) {
        if (msgNumber == null) {
            msgNumber = com.getNumber();
            try {
                com.wait();
            } catch (InterruptedException e) {
                // Restore the interrupted status
                Thread.currentThread().interrupt();
            }
        } else {
            if ((tempNumber = com.getNumber()) != msgNumber) {
                msgNumber = tempNumber;
                com.star(msgNumber, this.id);
            } else {
                continue;
            }
        }
    }
}
```

2.关键词识别

首先将`URL`中的文本按行存储在`List`中。

每输入一个关键词，就对文本进行一次遍历，反复调用`indexOf()`方法获得关键词的下一个出现位置，并更新计数器。

统计结果放在`LinkedHashMap`中，保证输出顺序和输入顺序相同。

注意按词频的统计需要使用稳定排序，比如`Collections.sort`。

统计词频的代码如下：

```
int count = 0;
for (String line: lines) {
    int beginIndex = 0;
    while((beginIndex = line.indexOf(word, beginIndex)) != -1) {
        count++;
        beginIndex++;
    }
}
mCounter.put(word, count);
```

3.网络攻击

客户端在循环中不断建立`socket`连接，连接成功后获得当前的`hack`值。

因为`hack ^ key = data`，所以`hack = key ^ data`.

每次发送完一个`hack`之后，将输出流和`socket`关闭，并调用`wait()`来等待`Cloud`的唤醒。唤醒后重新进行下一轮`hack`.

注意：为了防止`Cloud`更新之前就`hack`，需要对`Cloud.class`进行同步。

主要代码如下：

```
// TODO do as the problem description
try {
    while (true) {
        final Socket socket = new Socket(InetAddress.getLoopbackAddress(),
11111);
        PrintStream os = new PrintStream(new
BufferedOutputStream(socket.getOutputStream()));
        synchronized (Cloud.class) {
            int hack = Cloud.getKey() ^ Cloud.getData();
            os.println(hack);
            os.flush();
            os.close();
            socket.close();
            Cloud.class.wait();
        }
    }
} catch (Exception e) {
    //nop
}
```