

# 浙江大学实验报告

课程名称: JAVA 应用技术 指导老师: 翁恺 姓名: 吴雅芳

实验名称: Log Analyzer 实验类型: 编程实现 学号: 3120104690

## 一、实验目的和要求

目的:

要求:

Web servers typically maintain log files of client accesses to the web pages that they store. Given suitable tools, these logs enable web service managers to extract and analyze useful information such as:

- which are the most popular pages they provide;
- which ip took the most pages from the site;
- whether other sites appear to have broken links to this site's pages;
- how much data is being delivered to clients;
- the busiest periods over the course of a day, or week, or month.

Such information might help managers to determine, for instance, whether they need to upgrade to more powerful server machines, or when the quietest periods are in order to schedule maintenance activities.

The log analyzer project contains an application that performs an analysis of data from such a web server. The server writes a log line to a file each time an access is made. A sample log file called access.log is provided in the Resource division.

## 二、实验内容和原理（必填）

用一个自定义类 log 来存储一条记录中的各个部分值。用 substring 方法截取记录的相关值，放到 log 类中。用 arraylist 来存放单个的 log，并进行操作比较。

用 TreeMap 来承接 arraylist 中的值。Map 的 key 对应 arraylist 的相关字段，value 对应 arraylist 中该字段出现的次数。从而实现不重复的值。之所以用 TreeMap 是因为它可以进行排序，也很方便进行非重复值的存取。

用一个迭代器来对 TreeMap 进行遍历和排序。取出最值得到结论。

## 三、主要仪器设备（必填）

Eclipse

## 四、操作方法和实验步骤

- 1.复习 map,TreeMap 的结构和操作的基本知识，以及查阅 Date 类的操作方法。
- 2.编写程序代码
- 3.调试运行
- 4.测试

## 五、实验数据记录和处理

运行结果如图：

---

以行为单位读取文件内容，一次读一整行：

finished load!

the ip took the most pages from the site is:10.15.173.174

the most popular page is:/logout.php HTTP/1.1

the most busiest hour is:12

the most quietest hour is:3

other sites appear to have broken links to this site's pages

the total data delivered to clients is 1129276709 byte

## 六、实验结果与分析（必填）

运行正确。

## 七、讨论、心得

### 1.Date 类：

#### （1）日期比较：

```
java.util.Date nowdate=new java.util.Date();
```

```
String myString = "2008-09-08";
```

```
impleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd", Locale.CHINA);
```

```
Date d = sdf.parse(myString); boolean flag = d.before(nowdate);
```

```
if(flag) System.out.print("早于今天") else System.out.print("晚于今天")
```

#### （2）日期转化

1. 字符串转换成日期类型：

2. 方法 1：

3.

4. 也是最简单的方法 `Date date=new Date("2008-04-14");`

5.

6. 方法 2：

7.

8. `SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd");` //小写的 mm 表示的是分钟

9. `String dstr="2008-4-24";`

10. `java.util.Date date=sdf.parse(dstr);`

11.

12.

13.

14. 日期转换成字符串：

15.

16. `SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd");`

17. `java.util.Date date=new java.util.Date();`

18. `String str=sdf.format(date);`

### 2.字符串分割语法：

语法：

将一个字符串分割为子字符串，然后将结果作为字符串数组返回。

stringObj.split([separator, [limit]])参数

stringObj 必选项。要被分解的 String 对象或文字。该对象不会被 split 方法修改。

separator 可选项。字符串或 正则表达式对象，它标识了分隔字符串时使用的是一个还是多个字符。  
如果忽略该选项，返回包含整个字符串的单一元素数组。

limit 可选项。该值用来限制返回数组中的元素个数。

split 方法的结果是一个字符串数组，在 stingObj 中每个出现 separator 的位置都要进行分解,separator 不作为任何数组元素的部分返回。

第一次时以空格分割：

```
//分割字符串
//10.171.204.201 - - [25/Mar/2012:21:28:47 +0800] "GET ,
String ss[]=tempString.split(" ");
//数据库赋值
//ip:10.171.204.201
ip=ss[0];
//date:25/Mar/2012:21:28:47
String joinDate;
SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
joinDate=ss[3].substring(1,ss[3].length()-1);
try{
    date=sdf.parse(joinDate);
}catch(Exception e1){}
//page:/showCourse.php?id=55
page=ss[6];
//mark:200
mark=Integer.valueOf(ss[8]).intValue();
//datamount
datamount=Integer.valueOf(ss[9]).intValue();
```

出现问题：

ss9: 1604

```
Exception in thread "main" java.lang.NumberFormatException: For input string: "symgdj.wma"
    at java.lang.NumberFormatException.forInputString(Unknown Source)
    at java.lang.Integer.parseInt(Unknown Source)
    at java.lang.Integer.valueOf(Unknown Source)
    at Analyzer.Load(Analyzer.java:80)
    at Analyzer.main(Analyzer.java:44)
```

原因是：

```
310 27.10.157.42 - - [18/Mar/2012:17:37:32 +0800] "GET /~xgy/music/xialala/background/ygm - symgdj.wma HTTP/1.1" 404 231
311 222.205.12.54 - - [18/Mar/2012:17:37:32 +0800] "GET /showCourse.php?id=54%20ad%20acsi/substring(load_file/
```

并不是每条记录空格出现次数都相同。

最后采取按照特地子串分割的方式。

### 3.treemap 方法

(1) java.util.TreeMap.containsKey() method.: 判断是否包含某个键值

Exp:

System.out.println("Value for key 6 exists: "+ treemap.containsKey(6));

Value for key 6 exists: true

(2)treemap 遍历：

```
//这是用 TreeMap 的 keySet()方法，生成的对象是由 key 对象组成的 Set
//再利用 TreeMap 的 get(key)方法，得到对应的 value 值
Iterator it = tm.keySet().iterator();
    while (it.hasNext()) {
        //it.next()得到的是 key，tm.get(key)得到 obj
        System.out.println(tm.get(it.next()));
    }
```