一种基于 Java 编程的脏字过滤器的设计

孙文华
（南昌工程学院 网络信息中心，江西 南昌 330099）

摘 要：为了减少网络中的不良信息对青少年造成的危害，设计了一种脏字过滤的软件，可以发现那些网页内容中含有的不良信息，便于网络管理员对网络文化的维护。

关键词：Java；脏字；过滤器；设计
中图分类号：TP393 **文献标识码**：A **文章编号**：1674-7720(2012)09-0018-02

Design of dirty word filter based on Java programming

Sun Wenhua
(Center of Network Information, Nanchang Institute of Technology, Nanchang 330099, China)

Abstract: In order to reduce the bad information in the network to teenagers harm, the author designed a dirty word filter software, the undesirable information content is contained in the webpage can be found, it can make administrate the network become easy.

Key words: Java; dirty word; filter; design

良好的网络文化对培养青少年的爱国意识、创新精神、促进青少年良好的个性发展以及文化学习等方面都有积极的作用。但是网络文化中混杂着种种不良因素，对青少年造成许多负面影响：网络中的不健康内容不利于青少年的成长，甚至造成许多青少年犯罪行为的不断发生；网络世界的虚拟性还会造成青少年对现实社会的不满，青少年对网络世界的过分迷恋会导致网络孤独，网络中多元化的内容会导致青少年认识偏差，网络的隐匿性容易使青少年道德弱化[1]。特别令人担忧的是不良的网络文化对青少年的犯罪起着推动作用，值得全社会关注和重视。

本文提出了一种脏字过滤器软件的设计，对网络中不良的内容进行查找、发现，避免这些不良网络文化侵蚀青少年的心灵健康。

1 脏字过滤器的设计原理

脏字过滤器的原理图如图 1 所示。其原理如下：(1) 对脏字库的内容进行分割，把脏字库中所有的脏字或词组分开，并把这些脏字或词组存入数组中；(2) 将待测文件库中的文件进行逐个读取，并记录文件的内容；(3) 在待测文件中查找是否存在刚存放脏字或脏词组内容的数组里面的内容，如果有，进行标注等操作；如果没有，继续检查待测文件库中的下一个待测文件，直到待测文件库中的待测文件都被检查完为止；(4) 输出结果。即输出待测文件库中每个待测文件中包含脏字或脏词组的个数及出处等信息。

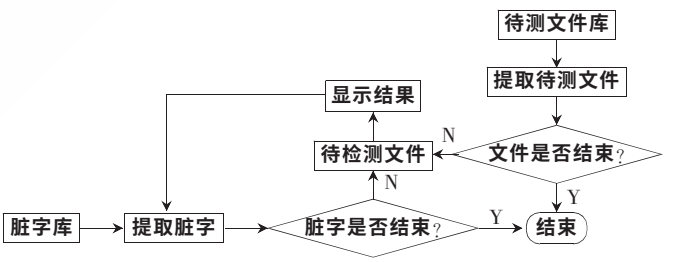

图 1 过滤器原理图

2 算法实现

软件开发环境：myeclipse 平台，Java 语言。首先以 myeclipse 平台新建一个 java project，在新建的 project 中需要导入下面相关文件：

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
```

```
import java.io.InputStreamReader;
import java.util.StringTokenizer;
```

过滤器软件中 main 函数的主要内容如下：

```
public static void main(String[] args) {
    BufferedReader br =new BufferedReader (new
InputStreamReader(System.in));
    String ans=null;
    int cnt=0;                          //字库中关键词个数
    int number=0;                       //脏字出现次数
     String filepath="D:                \\ 脏字待测文件库";
//待检测文件路径,此文件夹下可以放若干个待检测的文件
     String fileLibrarypath="D:        \\ 脏字典 \\file.txt";
                                        //脏字库文件的存放路径
    File file = new File(filepath);
    try {
                                        //读入用户输入的回车键信息
        Systemoutprintln("请按回车键,查看过滤信息:");
        String str = null;
        str = br.readLine();
        if (str != null) {
            if (! file.isDirectory()) {
                System.out.println("待检测文件路径
                                    不对,请修改路径。");
            } else if (file.isDirectory()) {
                ans=getcontent(fileLibrarypath);
                int k;
                StringTokenizer sst=new
                                    StringTokenizer(ans, "|");
                k = sst.countTokens();
                String[] record = new String[k];
                while (sst.hasMoreElements()) {
                    record[cnt] = sst.nextToken();
                    cnt++;
                }
                String[] filelist = file.list();
            for (int i = 0, flen = filelist.length; i
                                        < flen; i++){
                    String temp = filepath +
                                        "\\" + filelist[i];
                number = searchkeyword(record,
                                        cnt, temp);
                System.out.println("第"+(i+1)
            +"文件中脏字出现的次数:" + number);
                                        //字库中关键词个数
                }
            } else {
                                        //提示用户按回车键
                System.out.println("你还没有输入回
                                        车键信息");
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
        //输出查询结果
        if (ans != null) {
            System.out.println("字库中关键词个数:"+
                                cnt);//字库中关键词个数
                System.out.println("脏字库内容如下:"+
                                ans);
        } else {
            System.out.println("没有可以匹配的信息");
                                        //输出脏字库中的内容
        }
    }
    //得到指定路径文件中的内容
    private static String getcontent(String filepath) {
        String all = "";
        File file = new File(filepath);
        try {
            if (! file.isFile()) {
                System.out.println("文件路径不对,请修
                                        改路径");
            } else {
                File readfile = new File(filepath);
                BufferedReader br = new BufferedReader
                        (new FileReader(readfile));
                String ss = br.readLine();
                while (ss != null) {
                    all = all + ss;
                        //all 中存放读取的文件内容信息
                    ss = br.readLine();
                }
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return all;
    }
    //在待测文件中匹配脏字出现的次数
    private static int searchkeyword (String [] str, int cnt,
                                        String filepath){
        int number = 0;
        String s = "";
```

```
        s = getcontent(filepath);
        for (int i = 0; i < cnt; i++) {
            if (s.indexOf(str[i]) > -1) {
                number++;
            }
        }
        return number;
    }
```

至此，完成了脏字过滤器软件代码的编写工作，接下来可以进行 run 操作，即可以得到待测文件库中的待测文件包含脏字次数及出处等相关信息的结果。

3 实验结果分析

脏字库的存放路径：D:\脏字典\file.txt；脏字库文件中的内容略。

待测文件库的存放路径：D:\脏字待测文件库；文件库中存放了三个文件，分别为：test1.txt、test2.txt、test3.txt。

运行该过滤器软件后，得出的检测结果如图 2 所示。由图可以看到把待测文件中脏字及脏词组出现的次数全部显示出来，结果与实际情况完全一致。

图 2 进行检测后输出的结果信息

本文设计的脏字过滤器软件，已在 myeclipse 环境下通过 Java 语言实现，并对整个过滤器软件进行了测试，测试结果显示该设计完全可以达到对网页文件中的脏字进行过滤，还能指出这些脏字的数目及其出处。为网络管理员的管理带来方便，并给网络管理方面的编程人员提供了一个良好的开发平台。

参考文献

[1] 周伟文，侯建华.网络改变了什么：青少年的网络生存[M].石家庄：河北人民出版社，2005：292-294.

（收稿日期：2011-12-13）

作者简介：

孙文华，男，1981 年生，硕士研究生，助教，主要研究方向：计算机网络管理，网站开发和软件开发与设计等。