****

**Java课程上机报告**



**题 目： 上机作业3**

**学 号：**

**姓 名：**

**教 师：**

**日期：2019/10/22**

**一、上机内容或题目：**

**5-3**

用封装LinkedList的方式实现一个Stack容器。

**5-4**

用LinkedList实现一个类，该类有一个方法，该方法接受一个字符串作为参数，依次读取字符串中的字符，每次遇到“+”时就将“+”后面的一个字符压入栈内，每次遇到“-”时就将当前栈顶的字符弹出并在控制台打印，直至字符串遍历完，最后输出栈上剩余内容。实现main()方法调用这个方法，并输出字符串"+U+n+c- - -+e+r+t- - -+a-+i-+n+y- - -+ -+r+u–+l+e+s- - -"，观察输出。

**5-7**

从一个文件中读入语句，并进行词频统计。进一步改写程序，统计文件中Java关键字出现频率，在统计时，创建一个Set存储所有的Java关键字，在判断一个字符串记号是否为关键字时访问该Set。

**二、上机步骤及实验结果：**

（抓取截图，并文字说明）

**5-3：**

**思路**：LinkedList:双向链表，每个节点都有两个指针指向上一节点和下一节点。因为addLast(),addFirst(),getFirst(),getLast(),removeFirst()等方法，能把它当作Stack(栈)来用，java中的Stack接口已经废弃不用。在此中，使用自定义方法pop（出栈）,push（入栈），peek（查找栈顶元素），isEmpty（判断栈是否为空函数）和size（返回栈长函数）来定义栈的操作。基本思路就是在Stack类中封装一个LinkedList对象实例，在默认构造方法内将LinkedList对象初始化，剩下对栈的操作直接调用LInkedList的方法。

**代码：**

package p11\_3;

import java.util.LinkedList;

import java.util.Random;

public class Stack<E> {

private LinkedList<E> store;//封装一个LinkedList对象实例

public Stack(){

store=new LinkedList<E>();//初始化

}

public void push(E e){

store.addLast(e);//将元素添加在末尾就相当于入栈了

}

public E pop(){

return store.removeLast();//从末尾移除元素就相当于出栈了

}

public E peek(){

return store.getLast();

}

public boolean isEmpty(){

return store.isEmpty();

}

public int size(){

return store.size();

}

public static void main(String[] args){

Stack<Integer> s\_int=new Stack<Integer>();

Random rand=new Random(34);

for(int i=0;i<20;i++){

s\_int.push(rand.nextInt(40));

System.out.print(s\_int.peek()+" ");

}

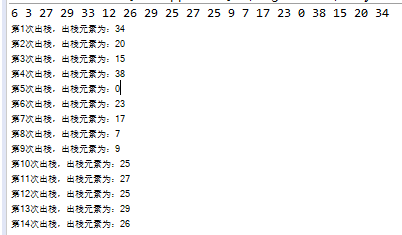
System.out.print("\n");

for(int i=0;i<20;i++)

System.out.println("第"+(i+1)+"次出栈，出栈元素为："+s\_int.pop());

}

}



**5-4：**

**思路：**定义一个Test\_LinkedList类，对输入待测试的字符串进行循环，利用charAt()读取字符与”+”和”-”进行判断，如果与”+”相同，则addLast()；遇到”-”,则输出removeLast()。

**代码：**

package p11\_3;

import java.util.LinkedList;

import java.util.Scanner;

public class Test\_LinkedList {

private LinkedList<Character> stack\_char;

public Test\_LinkedList(){

stack\_char=new LinkedList<Character>();

}

public void test\_string(String s){

for(int i=0;i<s.length();i++){

if(s.charAt(i)=='+'){

stack\_char.addLast(s.charAt(i+1));//入栈下一个字符

i+=1;//給计数变量i多加一，避免对下一个字符的检验（万一下一个字符是‘+’或‘-’呢？

}

else if(s.charAt(i)=='-')//出栈栈顶元素

System.out.println("pop栈顶元素："+stack\_char.removeLast());

else;

}

while(!stack\_char.isEmpty())

System.out.print(stack\_char.removeLast()+" ");

System.out.println();

}

public static void main(String[] args){

String s=null;

Scanner sc=new Scanner(System.in);

System.out.print("请输入待测试的字符串：");

s=sc.nextLine();

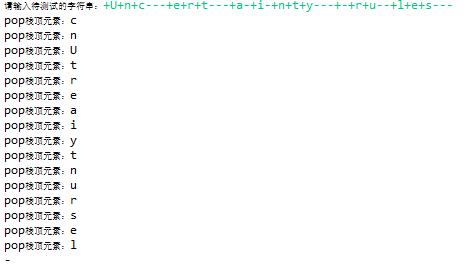
Test\_LinkedList t=new Test\_LinkedList();

t.test\_string(s);

sc.close();

}

}



**5-7：**

**思路：**创建Java关键字String数组，再加到HashSet存储。为了统计词频，最好使用Map接口，利用其键值之间的关系，在这里采用HashMap容器类。

文件读取：BufferedRead类，readline函数一次一行读取

词语匹配：StringTokenizer类，可以将字符串按指定分割符分成一个个的单词

关于读取匹配的关键字单词，如果获得的单词不为空则value加1，如果为空，则赋值为1.

**代码：**

package p11\_3;

import java.io.\*;

import java.util.\*;

public class words\_sta {

private File fd;//文件类

private Map<String,Integer> statistics;

private int count;//对文件中出现的词语总个数计数，为频率计算做准备

private static String[] key\_set\_str={"abstract","default","null","switch","boolean","do","if","package","nchronzed","break","double","implements","private","this","byte","else","import","protected","throw","throws","case","extends","instanceof","public","transient","catch","false","int","return","true","char","final","interface","short","try","class","finally","long","static","void","float","native","strictfp","volatile","continue","for","new","super","while","assert","enum"};

private static Set<String> key\_set\_of\_java=new HashSet<String>();

public words\_sta(String fileName){//默认构造方法是要传入一个文件名作为参数的

fd=new File(fileName);

statistics=new HashMap<String,Integer>();

count=0;

for(String s:key\_set\_str)

key\_set\_of\_java.add(s);//把java关键词加到Set里

}

public void work() throws FileNotFoundException,IOException{

String temp=null;

BufferedReader reader=new BufferedReader(new FileReader(fd));

while((temp=reader.readLine())!=null){//一行一行读取

StringTokenizer st=new StringTokenizer(temp,"{\r\n\t[0123456789]=\"，+：\\；;.}(),<>-! \*:");//分隔符

while(st.hasMoreTokens()){

String key=st.nextToken();

count++;

if(statistics.get(key)!=null){

statistics.put(key, statistics.get(key)+1);

}

else//map里不存在这个词语

statistics.put(key, 1);

}

}

reader.close();

}

//词频分析

public void print\_statistics(){

System.out.println("词频分析结果打印：\n"+statistics+"\n");

}

public void print\_key\_of\_java(){

for(Map.Entry<String,Integer> item:statistics.entrySet())//遍历Map

if(key\_set\_of\_java.contains(item.getKey()))

System.out.println("关键词“"+item.getKey()+"”的出现频率为："+((double)item.getValue()/count));

}

public static void main(String[] args) throws FileNotFoundException,IOException{

words\_sta w=new words\_sta("C://Users//natur//eclipse-workspace//p11\_3//src//p11\_3//TestMap.java");

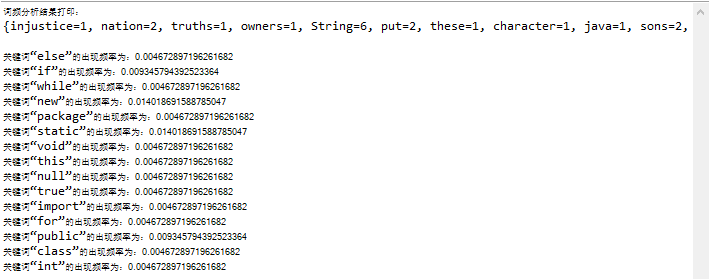
w.work();

w.print\_statistics();

w.print\_key\_of\_java();

}

}



**三、上机总结**

（自己完成本次上机后的心得）

1. 明白了List，Set，Map接口，掌握了它们容器类的初始化和使用方法
2. 具有自动扩充容量的List,其中LinkedList()中的addLast(),removeLast(),getLast()等方法。
3. charAt方法返回指定索引位置的char值
4. 使用Bufferedread类中readLine()时注意：

1.读入的数据要注意有/r或/n或/r/n

2.没有数据时会阻塞，在数据流异常或断开时才会返回null