实 验 报 告

课程名称	数据结构（Java）
实验项目	实验一 线性表

学　　院	计算机学院
系　　别	软件工程
班级/学号	软工 2101/2021011561
学 生 姓 名	陈志浩
实 验 日 期	2022.10.08
成　　绩	
指 导 教 师	蒋玉茹

实验一 线性表

一、实验目的

1. 自定义线性表数据结构

2. 在实际问题中应用线性表结构

二、实验内容及要求

【参考指导教师给的文档写】

三、系统设计

1.概述

本实验一共有 7 个 Java 文件，1 个 Python 文件，本次云通过使用 Github 中的 Python 开源库 WordCloud 生成，在 Java 文件中有 5 个类，1 个接口列表展示如下：

- Reader 类
- seqlist 类
- world_seqlist 类
- WordFreq 类
- Writer 类
- LList 接口

2.Reader 类的描述

属性及含义:

Reader 类无属性

方法及功能：

Reader 类的返回类型是 String 数组，Reader 类通过 BufferReader 读取路透社.txt 文本中的内容,当.readline()不为零时，持续读入。新建一个 String 类型的变量 s，将读入的数据进行粘合，最后创建 String 类型的数组 demo 用于接收 s.split()的数据，最后返回 demo 数组。

3.seqlist 类的描述

属性及含义：

seqlist 类含有两个属性，分别为 Object 类型的数组和 int 类型的 len 用于表示线性表的长度

方法及功能：

seqlist 一共有 10 个方法分别是：

1.void clean()方法 清空线性表

2.boolean isEmpty()方法 判断线性表是否为空

3.int size()方法 返回线性表的大小

4.T get(int i)方法 获取第 i 号位置上的元素

5.boolean set(int i, Object x)方法 将第 i 号位置元素设置为 x

6.void add(int I, Object x)方法 在第 i 号位置添加 x

7.T remove(int i)方法 删除第 i 号位置上的元素并

返回该元素

8.T indexOf(T x)方法 查找 x 是否在线性表中

9.boolean find(T x)方法 查找 x 是否在线性表中，若在线性表中则返回 true 反之返回 false

10.String toSting()方法 重构 toString 方法，使线性表能够被 println 语句输出

3.word_seqlist 类的描述

属性及含义：

word_seqlist 类无属性

方法及功能：

creat_seqlist(seqlist<WordFreq> list, int a)方法需要传入两个值分别为含有 WordFreq 对象的 seqlist 数组，以及整形 a，当输入的 a 等于 0 时，该方法将会返回停用词线性表；当输入为 1 时，该方法将会返回不含停用词的线性表

4.WordFreq 类的描述

属性及含义：

WordFreq 类含有两个属性分别为 world 和 value,分别表示被统计词，被统计词的词频

方法及功能：

WordFreq 类一共有四个方法分别为：

1.void set_WordFreq()方法 当调用该方法时 value 的值增加 1

2.String get_world()方法 当调用此方法时返回被统计词

3.int get_value()方法 当调用此方法时返回被统计词的词频

4.String toString()方法 重构 toString 方法，使 WordFreq 中的元素能够被 println 语句输出

5.Writer 类的描述

属性及含义：

Writer 类中无属性

方法及功能：

Writer 类是将词与词频按行一次存入 txt 文件中方便 Python 进行读取。Writer 类中的 void write_file(seqlist<WordFreq> list)没有返回类型，该方法接收一个元素类型为 WordFreq 的线性表,遍历线性表并调用 WordFreq 的 toString 方法，将得到的字符串存进字符串数组中，接着使用 Java IO 操作将字符串数组中的元素写入 txt 文件中保存。

6.LList 接口的描述

属性及含义：

该接口含有 9 个方法：

1.void clean(); // 清空线性表

2.boolean isEmpty(); // 判断线性表是否为空

3.int size(); // 返回线性表的长度

4.T get(int i); // 返回第 i 个元素

5.boolean set(int i, T x); // 将第 i 号位置元素设置为 x

6.void add(int i, T x); // 在第 i 号位置插入元素 x

7.T remove(int i); // 删除第 i 号元素并返回被删除的对象

8.T indexOf(T x); // 查找首次出现的 x 元素

9.boolean find(T x); // 查找首次出现的 x 元素

7.词云生成程序 main.py

该程序调用 Python 中的 imageio 库，matplotlib.pyplot 库，以及 WordCloud 库。该程序首先通过 open 函数读取 Java Writer 类生成的 txt 文件，通过分割函数，分别将词和词频存入两个不同的列表，随后调用 zip 函数将两个列表变成一个字典，使用 WordCloud 中的 fit_words 函数，使 WorldCloud 根据词频生成词云图，最后调用 matplotlib.pyplot 库，将词云图显示出来。

四、类图

四、系统环境

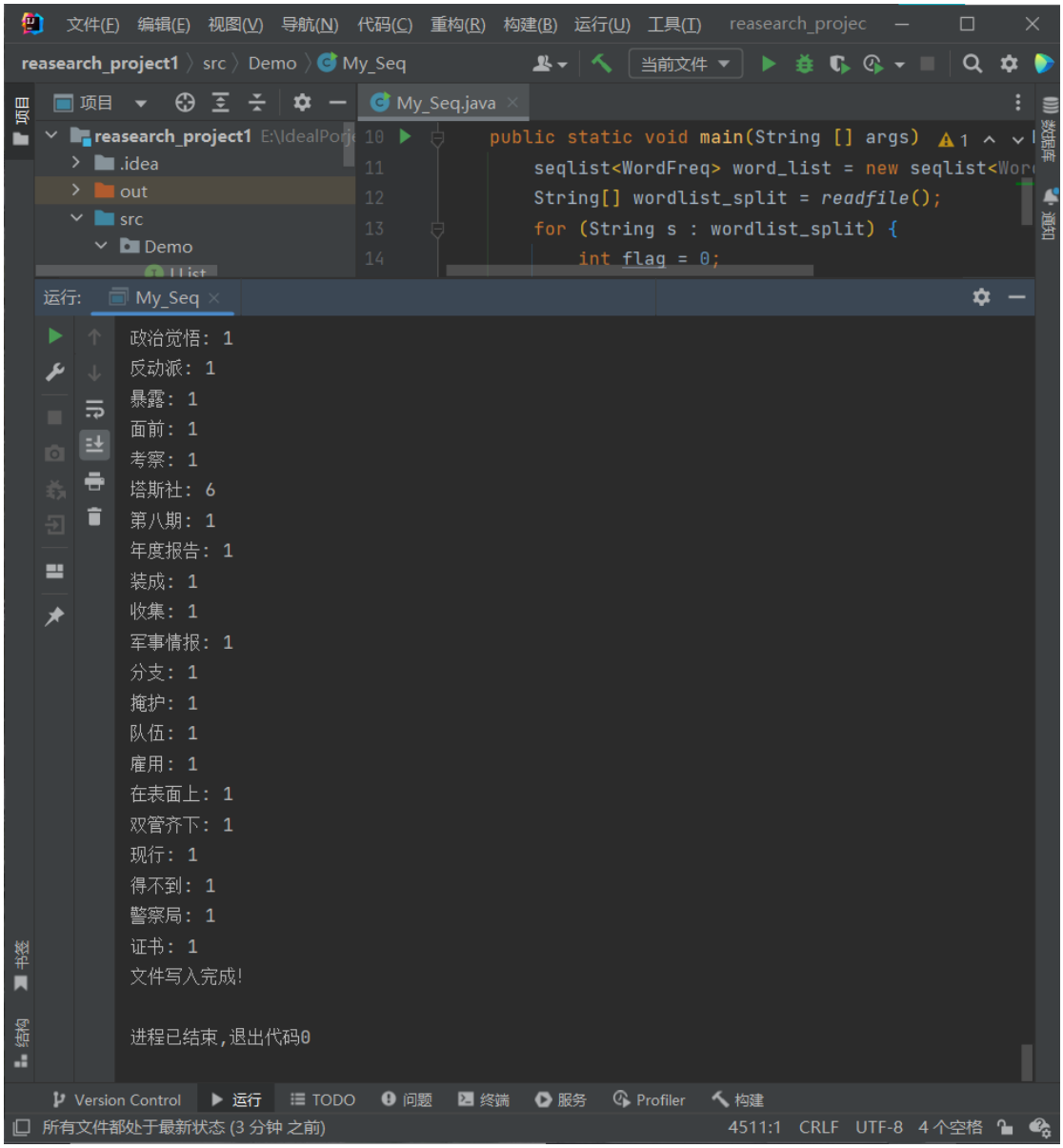
1.操作系统及版本　Windows 10 家庭中文版

2.JDK 及版本 java version “1.8.0_15”

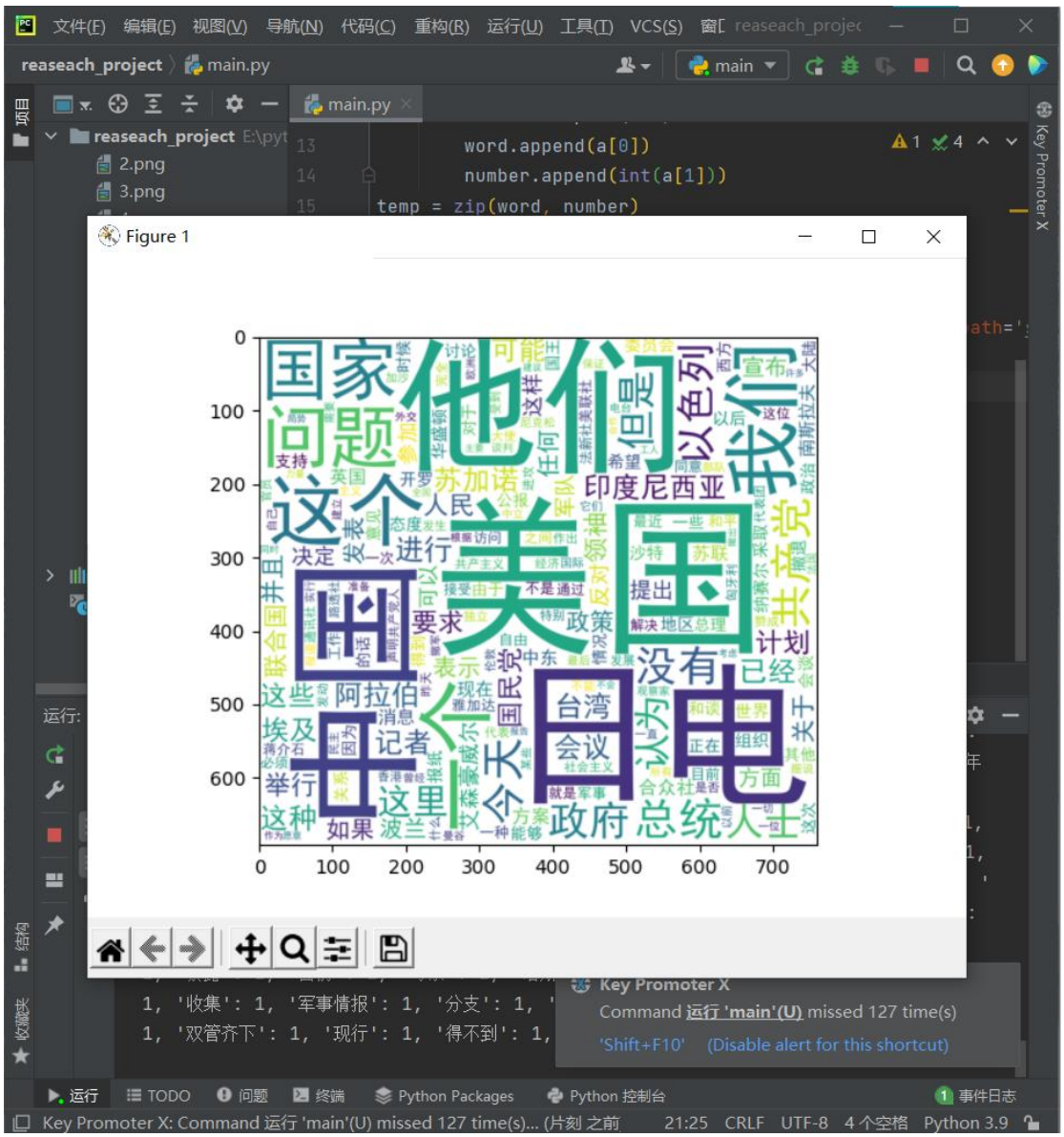
3.其他 JAR 包等 无

五、程序运行过程

【步骤 1】将已分好词的 txt 文件复制到 src 目录下，运行 src 目录下的 My_Seq.java 文件，运行结果如下：



【步骤 2】运行 python 文件：main.py 得到词云图，运行结果如下：



六、编写程序中遇到的问题总结

【依次列举编程中遇到的问题及解决过程】

1. 第一次实现词频统计时使用了其他数据结构

在刚开始实现词频统计时，未使用动态键，我首先通过将读入的字符串分割，得到字符串，通过将数组转化成 list 在转化成 set 进行去重，然后将集合中的元素依次添加到线性表中，随后再次对字符串数字进行遍历，并将线性表中相应的词频加 1。解决办法：通过使用动态键将元素添加到线

性表中，首先遍历被分割的字符数组，同时嵌套一个 for 循环依次，嵌套的 for 循环遍历线性表，当线性表中元素与字符串元素相同时，该词对应的词频加 1。

2.调用 WordCloud 生成词云时报错

WordCloud 库生成词云时，有两个函数分别为.generate()和.fit_words()函数，其中.generate()函数需要传入一段文本才能生成词云，我将压缩得到的数组传递给了.generate()函数发生了错误，导致词云生成失败，解决方法将.generate()函数变为.fit_word()函数。

报错信息：

```
Traceback (most recent call last):
  File "E:\Pythonproject\reaseach_project\main.py", line 19, in <module>
    wordcloud = WordCloud(background_color="white", font_path='simhei.ttf',
 mask=mask, width=1024, height=576, ).generate(
  File "C:\Users\28457\AppData\Local\Programs\Python\Python39\lib\site
 -packages\wordcloud\wordcloud.py", line 639, in generate
    return self.generate_from_text(text)
  File "C:\Users\28457\AppData\Local\Programs\Python\Python39\lib\site
 -packages\wordcloud\wordcloud.py", line 620, in generate_from_text
    words = self.process_text(text)
  File "C:\Users\28457\AppData\Local\Programs\Python\Python39\lib\site
 -packages\wordcloud\wordcloud.py", line 582, in process_text
    words = re.findall(regexp, text, flags)
  File "C:\Users\28457\AppData\Local\Programs\Python\Python39\lib\re.py", line
  241, in findall
    return _compile(pattern, flags).findall(string)
TypeError: expected string or bytes-like object

进程已结束，退出代码为 1
```

七、结果展示

八、参考文献

1. csdn wordcloud 生成词云图(含形状颜色设置) (57 条消息)

wordcloud 生成词云图（含形状、颜色设置）_IRON POTATO

的博客-CSDN 博客_wordcloud 生成词云

九、源代码

接口： LList

```
package Demo;

public interface LList<T> {
    void clean(); // 清空线性表
    boolean isEmpty(); // 判断线性表是否为空
    int size(); // 返回线性表的长度
    T get(int i); // 返回第 i 个元素
    boolean set(int i, T x); // 将第 i 号位置元素设置为 x
    void add(int i, T x); // 在第 i 号位置插入元素 x
    T remove(int i); // 删除第 i 号元素并返回被删除的对象
    T indexOf(T x); // 查找首次出现的 x 元素
    boolean find(T x); // 查找首次出现的 x 元素
}
```

类：My_Seq

```
package Demo;

import java.io.IOException;
import java.util.*;
import static Demo.Reader.readfile;
import static Demo.Writer.write_file;
import static Demo.word_seqlist.creat_seqlist;

public class My_Seq {
    public static void main(String [] args) throws IOException {
        seqlist<WordFreq> word_list = new seqlist<WordFreq>();
        String[] wordlist_split = readfile();
        for (String s : wordlist_split) {
            int flag = 0;
            for (int j = 0; j < word_list.size(); j++) {
                if (Objects.equals(word_list.get(j).get_world(), s)) {
                    word_list.get(j).set_WordFreq();
                    flag = 1;
                }
            }
```

```
            if (flag == 0) {
                word_list.add(word_list.size(), new WordFreq(s, 1));
            }
        }
        seqlist<WordFreq> stop_words_list1 = creat_seqlist(word_list, 1);
        for(int i = 0; i< stop_words_list1.size(); i++) {
            System.out.println(stop_words_list1.value[i].toString());
        }
        write_file(stop_words_list1);
    }
}
```

类：Reader

```
package Demo;

import java.io.*;

public class Reader {
    public static String[] readfile() throws IOException {
        String encoding = "utf-8";
        File file=new
File("E:\\IdealPorject\\reasearch_project1\\src\\2.txt");
        BufferedReader in = new BufferedReader(
                new InputStreamReader(
                        new FileInputStream(file)
                )
        );
        String s = "";
        String line;
        while((line=in.readLine())!=null){
            s += line;
        }
        String[] demo = s.split(" ");
        /*for(String a : demo){
            System.out.println(a);
        }*/
        // System.out.println(demo);
        return demo;
    }
    public static void main(String [] args) throws IOException {
        readfile();
```

```
    }
}
```

类：seqlist

```
package Demo;

import Demo.LList;

public class seqlist<T> implements LList<T> {
    public Object[] value;
    public int len;
    public seqlist(int max_size){
        this.value = new Object[max_size]; // 创建数组
        this.len = 0;
    }
    public seqlist(){
        this.value = new Object[9999]; // 创建数组
        this.len = 0;
    }


    @Override
    public void clean() { // 清空线性表
        this.len = 0;
    }

    @Override
    public boolean isEmpty() {
        return this.len == 0; // 判断线性表是否为空
    }

    @Override
    public int size() {
        return this.len; // 返回线性表的大小
    }

    @Override
    public T get(int i) {
        T a;
        if(i>=0 && i<this.len){
            a =(T)this.value[i];  // 获取第 i 个位置上的元素
        }else{
```

```
            a = null;
        }
        return a;
    }

    @Override
    public boolean set(int i, Object x) {
        if(x==null){ // x为空返回false
            return false;
        }
        if(i>=0 && i < this.len) {
            this.value[i] = x;
            return true;
        }else{
            throw new IndexOutOfBoundsException("index outof bound");
        }
    }

    @Override
    public void add(int i, Object x) {
        if(x == null){
            return ;
        }
        if(this.len == value.length){
            Object[] temp = this.value;
            this.value = new Object[temp.length * 2];
            for(int j = 0; j< temp.length; j++){
                this.value[j] = temp[j]; // 将temp中的代码复制到value中
            }
        }
        if(i<0){
            i=0;
        }
        if(i>this.len){
            i = this.len;
        }
        for(int j=this.len-1; j>=i;j++){
            this.value[j+1] = this.value[j]; // i之后元素向后移动
        }
        this.value[i] = x;
        this.len++;
    }

    @Override
```

```
    public T remove(int i) {
        if(this.len==0 || i<0 || i>=this.size()){
            return null;
        }
        T a = (T)this.value[i];
        for(int j = this.len-1; j>=i; j--){
            this.value[i] = this.value[i+1]; // 后面的元素向前移动
        }
        this.value[this.len-1] = null; // 让之前最后一个元素为零
        this.len--;
        return a;
    }

    @Override
    public T indexOf(T x) {
        for(int i = 0; i<this.len; i++){ // 查找 x 是否在线性表中
            if(x.equals(this.value[i])){ // 判断是否相等
                return (T)this.value[i]; // 强制类型转换
            }
        }
        return null;
    }

    @Override
    public boolean find(T x) {
        for(int i = 0; i<this.len; i++){ // 查找 x 是否在线性表中
            if(x.equals(this.value[i])){ // 判断是否相等
                return true; // 强制类型转换
            }
        }
        return false;
    }

    @Override
    public String toString() { // 重构 toString 方法
        String a = "";
        for(int i = 0; i<this.len; i++){
            a += this.value[i] + " ";
        }
        return a;
    }
}
```

类：word_seqlist

```
package Demo;

import java.util.Objects;

public class word_seqlist {
    public static seqlist<WordFreq> creat_seqlist(seqlist<WordFreq> list, int
a) {
        seqlist<WordFreq> word_list = new seqlist<WordFreq>();
        if(a==0) {
            for (int i = 0; i < list.size(); i++) {
                if (list.get(i).get_world().length() <= 1) {
                    WordFreq temp = list.get(i);
                    word_list.add(list.size(), temp);
                }
            }
        }
        if(a==1) {
            for (int i = 0; i < list.size(); i++) {
                if (list.get(i).get_world().length() > 1) {
                    WordFreq temp = list.get(i);
                    word_list.add(list.size(), temp);
                }
            }
        }
        return word_list;
    }
}
```

类：WordFreq

```
package Demo;

public class WordFreq {
    private String world;
    private int value;

```

```
    public WordFreq(){
        this.world = "";
        this.value = 0;
    }

    public WordFreq(String world, int value){
        this.world = world;
        this.value = value;
    }

    public WordFreq(String world){
        this.world = world;
        this.value = 1;
    }

    public void set_WordFreq(){
        this.value++;
    }

    public String get_world(){
        return this.world;
    }
    public int get_value(){
        return this.value;
    }

    @Override
    public String toString() {
        return this.world + ": " + this.value;
    }
}
```

类：Writer

```
package Demo;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;

public class Writer {
```

```
    public static void write_file(seqlist<WordFreq> list){
        String[] string_list = new String[list.size()];
        for(int i = 0; i< list.size(); i++) {
            string_list[i] = (list.value[i].toString());
        }
        try{
            BufferedWriter out = new BufferedWriter(
                    new FileWriter("wordlist.txt"));
            String s = "";
            for (String value : string_list) {
                s = value + "\n";
                out.write(s);
            }
            out.close();
            System.out.println("文件写入完成！");
        }catch (IOException ignored){}

    }

}
```

Python 程序： main.py

```
import imageio
import matplotlib.pyplot as plt
from wordcloud import WordCloud


file = "E:\\IdealPorject\\reasearch_project1\\wordlist.txt"
with open(file, "r") as f:
    word = []
    number = []
    data = f.readlines()
    for line in data:
        a = line.split(":")
        word.append(a[0])
        number.append(int(a[1]))
temp = zip(word, number)
final_dic = dict(temp)
print(final_dic)
mask = imageio.v3.imread("7.png")
wordcloud = WordCloud(background_color="white", font_path='simhei.ttf',
mask=mask, width=1024, height=576, ).fit_words(
```

```
    final_dic)
wordcloud.to_file('demo7.jpg')
plt.imshow(wordcloud)
plt.show()
```