



学 院 智能与计算

专 业 软件工程

班 级 三班

学 号 3018216144

姓 名 王文君

一、实验要求

编写程序，统计了不起的盖茨比中各个单词出现的频次。

注意事项

1. 尝试使用不同的 stream 进行读文件操作。
2. 异常处理（例如文件不存在，文件没有读权限，文件编码错误等）

输入：

了不起的盖茨比（英文版）.txt

(其中一个)

输出:

为输入文件, 创建一个 output.txt

输出格式如下, 单词+空格+频次, 结果按照单词的频次倒序排列

hello 123

hi 12

i 1

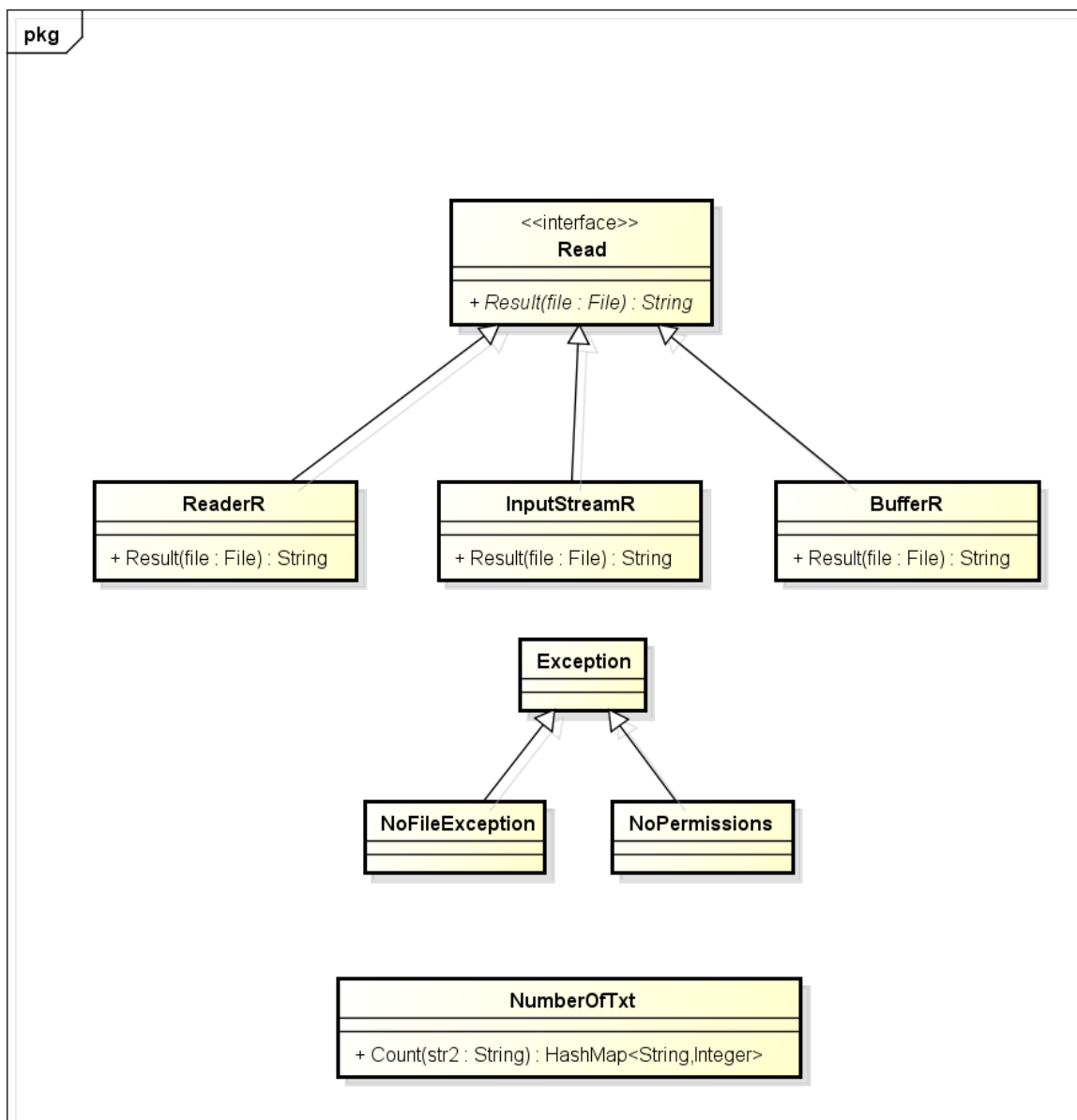
二、设计思路和UML图

NoFileException: 处理文件不存在错误

NoPermissions: 处理文件没有读权限

Read: 不同读文件方式的接口

NumberOfTxt: 计数



三、源代码

```
public interface Read {
    String Result(File file) throws Exception;
}

public class BufferR implements Read {
    public String Result(File file) throws Exception
    {
        String str=null;
        if (!file.exists()) {
            throw new NoFileException();
        }
        else
        {
            FileReader reader = new FileReader(file);
            StringBuilder sb=new StringBuilder();
            BufferedReader br = new BufferedReader(reader);
            String ss=null;

            while ((ss=br.readLine())!= null) {
                sb.append(ss);
                sb.append(" ");
            }
            str=sb.toString();
            reader.close();
            br.close();
        }
        return str;
    }
}

public class InputStreamR implements Read{
    public String Result(File file) throws Exception
    {
        String st=null;
        if (!file.exists()) {
            throw new NoFileException();
        }
        else
        {
            InputStream is = new FileInputStream(file);
            byte data[] = new byte[1024];
            int n=0;
            while((n=is.read(data))!=-1)
            {
                String w=new String(data,0,n);
                st+=w;
            }
            is.close();
        }
    }
}
```

```

    }

    return st;

}

}

public class ReaderR implements Read{
    public String Result(File file) throws Exception
    {
        String st="";
        if (!file.exists()) {
            throw new NoFileException();
        }
        else
        {
            Reader in = new FileReader(file);
            char data[] = new char[1024];
            int temp = 0;
            int foot = 0;
            while((temp = in.read(data))!=-1) {
                String w=new String(data,0,temp);
                st+=w;
            }
            in.close();
        }

        return st;
    }
}

```

```

public class NumberOfTxt {

    HashMap<String,Integer> Count(String str2) {
        HashMap<String,Integer>map=new HashMap<>();
        String str[]=str2.split("\\s+");
        for(int i=0;i<str.length;i++)
        {
            if(str[i].length()>0)
            {
                if(map.containsKey(str[i]))
                {
                    map.put(str[i], map.get(str[i])+1);
                }
                else
                    map.put(str[i], 1);
            }
        }

        return map;
    }
}

```

```

public class Test {
    public static void main(String[] args) throws Exception {

        File file = new File("D:\\下载的东西\\了不起的盖茨比英文.txt");
        File file2 = new File("D:\\下载的东西\\output.txt");
        if (!file.exists()) {
            throw new NoFileException();
        }
        else
        {
            if(!file.canRead())
            {
                throw new NoPermissions();
            }
            else
            {
                StringBuilder sb=new StringBuilder();
                HashMap<String,Integer>map=new HashMap<>();
                file2.createNewFile();
                FileWriter filewriter = new FileWriter(file2,true) ;

                /*      BufferedReader*/
                //          String result=new BufferR().Result(file);

                /*      InputStream */
                //          String result=new  InputStreamR().Result(file);

                /*      Reader          */
                //          String result=new ReaderR().Result(file);

                map=new NumberOfTxt().Count(result);

                List<Map.Entry<String,Integer>> list = new
                ArrayList<Map.Entry<String,Integer>>(map.entrySet());
                //然后通过比较器来实现排序
                Collections.sort(list,new
                Comparator<Map.Entry<String,Integer>>() {
                    //升序排序
                    public int compare(Entry<String, Integer> o1,
                Entry<String, Integer> o2) {
                        return -o1.getValue().compareTo(o2.getValue());
                    }
                });

                for(Entry<String, Integer> mapping:list){
                    String str=(mapping.getKey()+" "+mapping.getValue());
                    str+="\r\n";
                    filewriter.write(str);
                }

                filewriter.close();
            }
        }
    }
}

```

```
}  
  
}  
  
}  
}
```

四、运行结果

