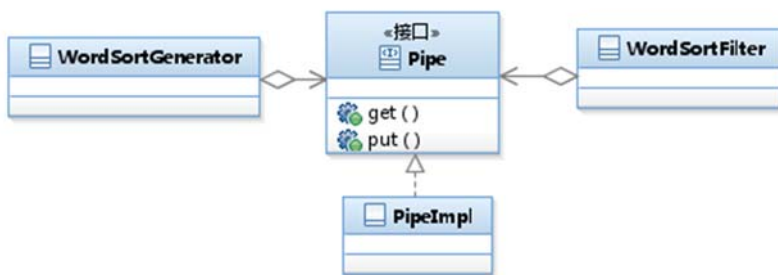


## 管道-过滤器风格案例：一个单词排序程序

一个管道-过滤器风格的例子是单词排序程序，该程序将一个文本文件中的所有单词按照字典顺序进行排列。一般来说，这种文本的每一行包含一个单词；同样，排好序的输出文件中每行也是一个按照顺序排列的单词。

该程序可以使用**管道-过滤器风格**来实现，其具体解决方案如下图所示，其中单词产生器 WordSortGenerator 负责从磁盘中读取文件，并将数据流输入到 Pipe 当中；Pipe 则负责数据的传输工作，将数据传送到单词排序过滤器 WordSortFilter 进行处理；WordSortFilter 是具体的过滤器，能够对传入的数据流进行排序然后将结构写入文件。



单词排序程序类图

该程序的输入为 word.txt 文件，文件中每一行包含一个单词。文件中的每一个单词经由管道 PipeImpl 进入到排序过滤器 WordSortFilter 中后，WordSortFilter 执行单词排序操作，并输出按顺序排列的单词。其具体代码实现过程如下：

(1) 首先是管道接口：

```
interface Pipe {
    public boolean put(Object obj);
    public Object get() throws InterruptedException;
}
```

(2) 管道实现代码：

```
public class PipeImpl implements Pipe {
    private List buffer = new ArrayList();
    public synchronized boolean put(Object obj) {
        boolean bAdded = buffer.add(obj);
        notify();
        return bAdded;
    }
    public synchronized Object get() throws InterruptedException {
        while(buffer.isEmpty()) wait(); //pipe empty - wait
        Object obj = buffer.remove(0);
        return obj;
    }
}
```

(3) 单词产生器代码：

```
public class WordSortGenerator extends Thread {
    private Pipe pipe = null;
    public WordSortGenerator(Pipe _pipe) {
        pipe = _pipe;
    }
    public void run() {
        try {
            BufferedReader br = new BufferedReader(new FileReader("words.txt"));
            String word = null;
            while((word = br.readLine()) != null)
                pipe.put(word);
        }
    }
}
```

```

        pipe.put(null); //null signals no more input
        br.close();
    } catch (IOException ioex) {
        ioex.printStackTrace();
    }
}
}

```

(4) 单词排序过滤器代码:

```

public class WordSortFilter extends Thread {
    private Pipe pipe = null;
    private List<String> wordList = new ArrayList<String>();
    public WordSortFilter(Pipe _pipe) {
        pipe = _pipe;
    }
    public void run() {
        String word = null;
        try {
            while ((word = (String) pipe.get()) != null)
                wordList.add(word);
        } catch (InterruptedException intrtex) {}
        //now sort the word list
        Collections.sort(wordList);
        //print the sorted word list and write it to a file
        try {
            FileWriter fw = new FileWriter("sortedwords.txt");
            for (String s : wordList) {
                System.out.println(s);
                fw.write(s + "\n");
            }
            fw.close();
        } catch (IOException ioex) {
            ioex.printStackTrace();
        }
    }
}

```

(5) 具体调用过程:

```

Pipe pipe = new PipeImpl();
Thread wordGenerator = new WordSortGenerator(pipe);
Thread wordSortFilter = new WordSortFilter(pipe);
wordGenerator.start();
wordSortFilter.start();

```