

软件设计模式实验报告

**2021/2022(2)**



实验题目 装饰模式的应用

学生姓名 卢鹏宇

学生学号 201906061909

学生班级 软件工程1904

任课教师 黄洪

提交日期 2022/05/05

**计算机科学与技术学院、软件学院**

**一、设计问题**

在论坛中为了防止用户发表不雅言论，常常需要将某些敏感词汇屏蔽。假设某论坛是将发帖内容保存在磁盘文件中，请利用装饰模式的概念编写一个过滤流类ReplaceWriter，使用该类的Write字符串系列方法会将敏感词汇用“\*”替换掉。需要替换的敏感词汇保存在文件badwords.txt中。请按实验的要求提交实验报告和项目实现的源代码。（提示：参考FilterWriter)

**二、问题分析与模式选用**

分析上诉需求可以得出，该设计问题要求对Write类的功能进行扩展，使得其在将字符串写入文件时，会对字符串内容进行过滤，将敏感词汇用“\*”替换。因此，该问题可以用装饰模式来实现，Write作为被装饰的对象。

装饰模式包含如下角色：

* Component（抽象构件）
* ConcreteComponent（具体构件）
* Decorator（抽象装饰类）
* ConcreteDecorator（具体装饰类）

装饰模式结构图如下：

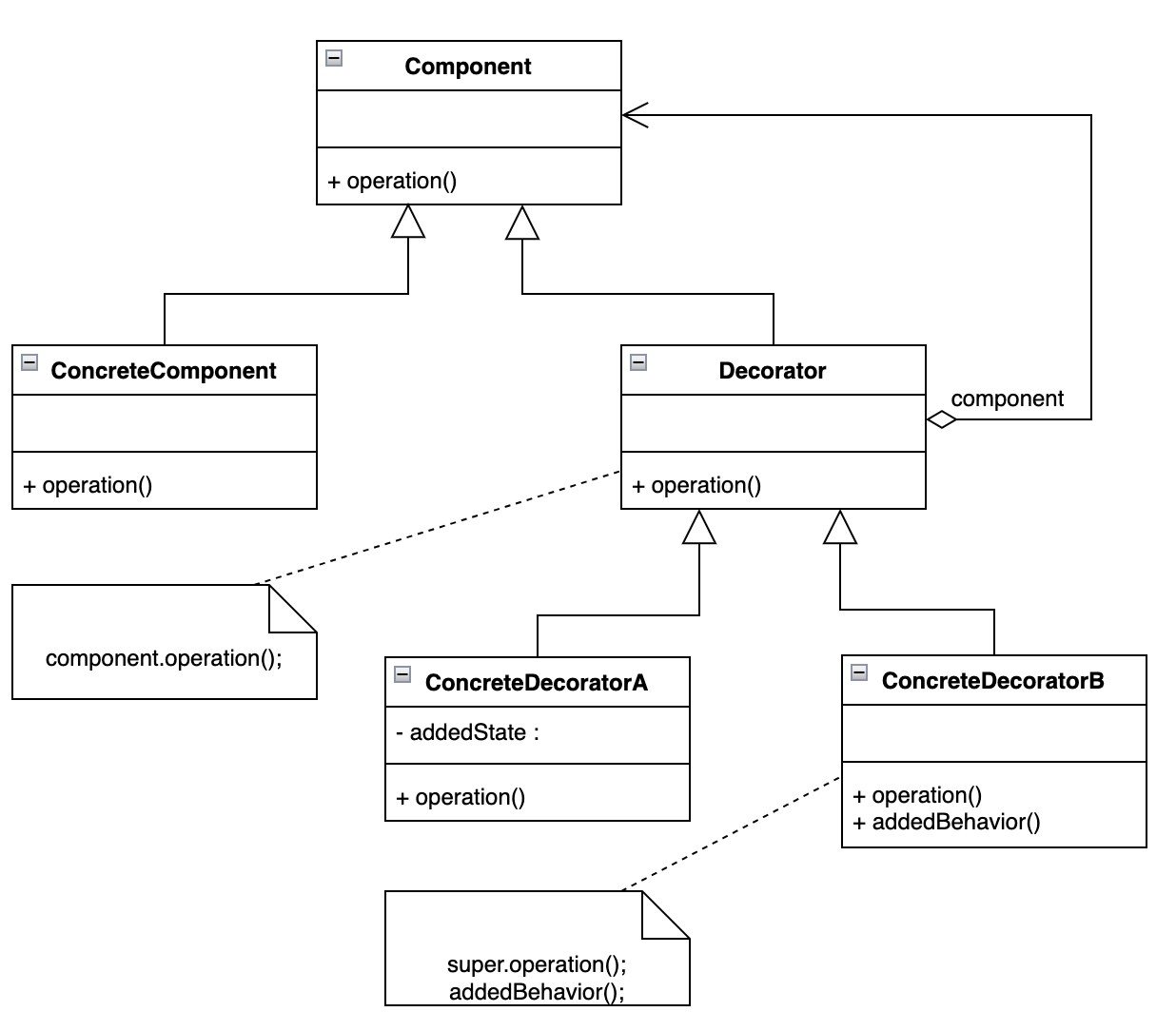


图 2-1 装饰模式结构图

**三、设计方案**

分析需求可得，解决以上问题需解决以下几点：

* 对发帖内容的敏感词汇进行过滤；
* 敏感词汇保存在badwords.txt文件中；
* 敏感词汇用“\*”替换。

所以，可以将Writer类设置为抽象构件，FileWriter为其具体构件，再设置以下两个类来实现：

* ReplaceWriter类

该类为一个具体装饰类，重写了write()方法，同时，也增加了readBadWords()方法和doFilter()方法，分别实现了对敏感词汇的读入功能和将敏感词汇用“\*”替换的功能。

* Main类

该类为一个主类，在该类中创建ReplaceWriter对象并调用调用其方法实现功能。

**四、运行结果及效果分析**

1.运行结果：若过滤成功，则在控制面板输出“Replace success!”

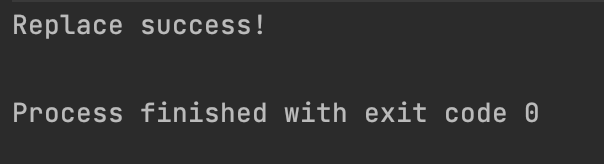


图 4-1 运行结果

1. 执行后的结果保存到result.txt文件中，执行"fuck,tmd.wc,you are so nice!傻逼"语句后，result.txt文件中的结果为：

截屏2022-05-05 21.34.22

图 4-2 result.txt结果

1. **关键代码（算法）及其说明**
2. readBadWords方法：该方法实现了将badwords.txt中的敏感词汇读取出并保存到列表中。

public void readBadWords(String path) {  
 try (Reader in = new FileReader(path)) {  
 char[] buf = new char[1024 \* 1024];  
 //读取  
 int len = in.read(buf);  
 String s = String.*valueOf*(buf, 0, len);  
 //分割词语  
 String[] words = s.split("[ ]+");  
 //加载词语到列表中  
 badWords = new ArrayList<>(Arrays.*asList*(words));  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
}

1. doFilter方法：该方法实现了将敏感词汇替换为“\*”

private void doFilter(char[] cbuf) {  
 //转为string  
 String s = String.*valueOf*(cbuf);  
 for (String word : badWords) {  
 //匹配每个敏感词，用相同长度的\*代替  
 s = s.replaceAll(word, String.*join*("", Collections.*nCopies*(word.length(), "\*")));  
 }  
 //将过滤后的string转为char数组  
 s.getChars(0, s.length(), cbuf, 0);  
}

**六、实验体会**

通过本次实验，我加深了对装饰模式的理解，并且能够用装饰模式来解决具体的问题。