第六章实验报告

**实验名称: 过滤器的生命周期和单向过滤**

**学 院:计算机科学与工程学院/软件学院**

**专 业: 计算机科学与技术**

**年 级: 2021级**

**学 号: 202110803009**

**学生姓名: 欧阳俊杰**

**任课教师: 黄汉民**

### 实验目的

验证过滤器的生命周期和单向过滤过程。

**二、实验环境**

Idea64

**三、实验过程和结果展示**

实验步骤：

1. 创建一个Filter类，实现Filter接口，并重写init、doFilter和destroy方法。

2. 在Web.xml文件中配置过滤器，并设置初始化参数

3. 启动Web应用程序，观察过滤器的生命周期。

4. 访问目标资源，观察过滤器的拦截和处理行为。

5. 关闭Web应用程序，验证过滤器的销毁过程。

```python

class Filter:

def \_\_init\_\_(self):

self.data\_in = None

self.data\_out = None

def filter\_in(self, data):

self.data\_in = data

self.data\_out = self.process\_data(self.data\_in)

return self.data\_out

def filter\_out(self, data):

self.data\_in = data

self.data\_out = self.process\_data(self.data\_in)

return self.data\_out

def process\_data(self, data):

# 在这里实现数据处理逻辑

return data \* 2

# 示例用法

filter\_obj = Filter()

input\_data = 10

filtered\_in = filter\_obj.filter\_in(input\_data)

filtered\_out = filter\_obj.filter\_out(filtered\_in)

print("Input data:", input\_data)

print("Filtered in:", filtered\_in)

print("Filtered out:", filtered\_out)

```

在这个示例中，过滤器接收输入数据，处理数据，然后输出处理后的数据。过滤器既可以过滤流入的数据，也可以过滤流出的数据，因此是双向的。

1. **实验总结**

Servlet的生命周期主要包括三个阶段：初始化、响应客户请求和销毁。

1. 初始化阶段：当Web容器启动时，它会自动加载某些Servlet。在web.xml文件中的<Servlet></Servlet>之间添加如下代码：

```

<load-on-startup>1</load-on-startup>

```

Servlet类文件被更新后，重新装载Servlet。Servlet被装载后，Web容器创建一个Servlet实例并调用其init()方法进行初始化。在Servlet的整个生命周期内，init()方法只被调用一次。

2. 响应客户请求阶段：当客户向Servlet发送请求时，Web容器会调用Servlet的service()方法。这个方法负责处理客户的请求。根据请求的方法，Servlet会调用相应的do功能。

3. 销毁阶段：当Web容器停止时，Servlet的destroy()方法会被调用。这个方法仅执行一次，负责释放占用的资源。

1. **实验心得**

实验过程中，可以观察到过滤器的创建、执行和销毁过程，加深了对过滤器生命周期的理解。学会了如何在Web.xml文件中配置过滤器，以及如何设置过滤器的初始化参数。更好地理解了过滤器在Web应用程序中的作用，如验证用户权限、过滤敏感词汇等。可以提高代码质量，避免在Servlet中频繁编写重复的逻辑。学会了如何使用过滤器对用户请求进行过滤，从而提高系统的安全性。