



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №7

ТРПЗ

Тема: Патерни проектування. HTTP-сервер

Виконав:

студент групи IA-31

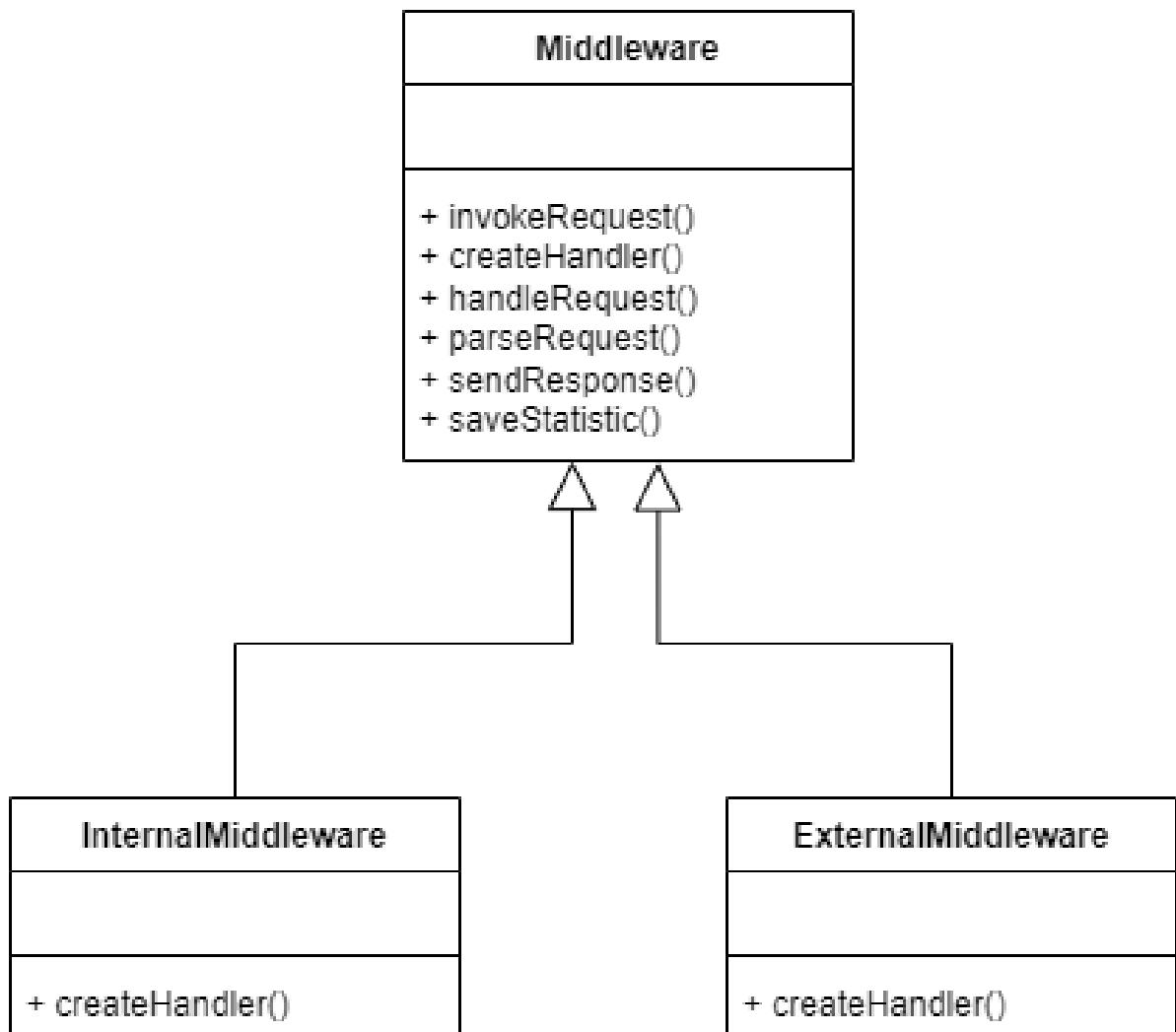
Машин Д. І.

Київ 2025

Мета: Вивчити структуру шаблонів «Mediator», «Facade», «Bridge», «Template method» та навчитися застосовувати їх в реалізації програмної системи

Завдання

- Ознайомитись з короткими теоретичними відомостями.
- Реалізувати частину функціоналу робочої програми у вигляді класів та їхньої взаємодії для досягнення конкретних функціональних можливостей.
- Реалізувати один з розглянутих шаблонів за обраною темою.
- Реалізувати не менше 3-х класів відповідно до обраної теми.
- Підготувати звіт щодо виконання лабораторної роботи. Поданий звіт повинен містити: діаграму класів, яка представляє використання шаблону в реалізації системи, навести фрагменти коду по реалізації цього шаблону.



```
14 @↓ public abstract class Middleware {
15   ↗ 1 usage 2 implementations
16   ↗ public abstract Handler createHandler();
17
18   2 usages
19   ↗ public void invokeRequest(Socket socket) throws IOException {
20     ↗ long requestTime = System.currentTimeMillis();
21     ↗ HttpRequest request = parseRequest(socket.getInputStream());
22     ↗ Handler handler = createHandler();
23     ↗ HttpResponse response = handleRequest(handler, request);
24     ↗ long responseTime = System.currentTimeMillis();
25     ↗ sendResponse(response, socket.getOutputStream());
26     ↗ long time = responseTime - requestTime;
27     ↗ //saveStatistic(request, response, time);
28   }
29
30   1 usage
31   ↗ public HttpResponse handleRequest(Handler handler, HttpRequest request) { return handler.handle(request); }
32
33   1 usage
34   ↗ public HttpRequest parseRequest(InputStream inputStream) throws IOException {
35     ↗ HttpRequestParser httpRequestParser = new HttpRequestParser();
36     ↗ return httpRequestParser.parse(inputStream);
37   }
38
39   1 usage
40   ↗ public void sendResponse(HttpResponse response, OutputStream outputStream) throws IOException {
41     ↗ response.send(outputStream);
42   }
43
44   no usages
45   ↗ public void saveStatistic(){}
46
47
48   1 usage
49   ↗ public class InternalMiddleware extends Middleware{
50     ↗ 2 usages
51     ↗ Map<String, HttpRequestCallback> routes;
52
53     ↗ 1 usage
54     ↗ public InternalMiddleware(Map<String, HttpRequestCallback> routes){
55       ↗ this.routes = routes;
56     }
57
58     ↗ 1 usage
59     ↗ @Override
60     ↗ public Handler createHandler() { return new InternalHandler(routes); }
61   }
62
63
64   3 public class ExternalMiddleware extends Middleware{
65
66
67     ↗ 1 usage
68     ↗ @Override
69     ↗ public Handler createHandler() {
70       ↗ return new ExternalHandler();
71     }
72
73   }
```