



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 №5

ТРПЗ

Тема: Патерни проектування. HTTP-сервер

Виконав:

студент групи IA-31

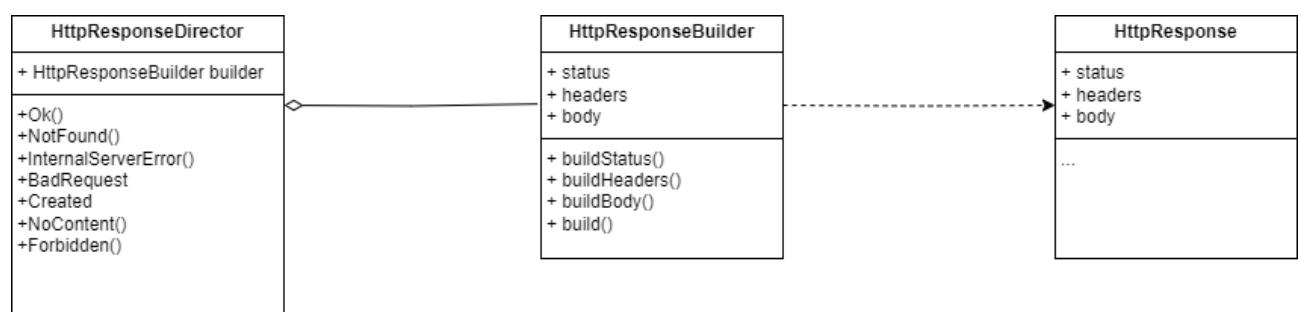
Машин Д. І.

Київ 2025

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.

Шаблон Builder



Лістинг коду

Клас HttpResponse

```
7  public class HttpResponse {
8
9      3 usages
10     private final String status;
11     3 usages
12     private final Map<String, String> headers;
13     4 usages
14     private final String body;
15
16
17     1 usage
18     public HttpResponse(String status, Map<String, String> headers, String body) {
19         this.status = status;
20         this.headers = headers;
21         this.body = body;
22     }
23 }
```

Клас HttpResponseBuilder

```
6  public class HttpResponseBuilder{  
7      2 usages  
8      private String status;  
9      2 usages  
10     private Map<String, String> headers;  
11     2 usages  
12     private String body;  
13  
14  
15     7 usages  
16     > public void buildStatus(String status) { this.status = status; }  
17  
18     7 usages  
19     > public void buildHeaders(Map<String, String> headers) { this.headers = headers; }  
20  
21     6 usages  
22     > public void buildBody(String body) { this.body = body; }  
23  
24  
25     7 usages  
26     > public HttpResponse build() { return new HttpResponse(status, headers, body); }  
27  
28 }  
29
```

Клас HttpResponseDirector

```
5  public class HttpResponseDirector {  
6      private static final HttpResponseBuilder builder = new HttpResponseBuilder();  
7      public static HttpResponse OK(String body, Map<String, String> headers) {  
8          builder.buildStatus("200 OK");  
9          builder.buildHeaders(headers);  
10         return builder.build();  
11     }  
12     public static HttpResponse NotFound(Map<String, String> headers) {  
13         builder.buildStatus("404 Not Found");  
14         builder.buildHeaders(headers);  
15         builder.buildBody(null);  
16         return builder.build();  
17     }  
18     public static HttpResponse InternalServerError(Map<String, String> headers) {  
19         builder.buildStatus("500 Internal Server Error");  
20         builder.buildHeaders(headers);  
21         builder.buildBody(null);  
22         return builder.build();  
23     }  
24     public static HttpResponse BadRequest(Map<String, String> headers) {  
25         builder.buildStatus("400 Bad Request");  
26         builder.buildHeaders(headers);  
27         builder.buildBody(null);  
28         return builder.build();  
29     }  
30     public static HttpResponse Created(String body, Map<String, String> headers) {  
31         builder.buildStatus("201 Created");  
32         builder.buildHeaders(headers);  
33         builder.buildBody(body);  
34         return builder.build();  
35     }  
36     public static HttpResponse NoContent(Map<String, String> headers) {  
37         builder.buildStatus("204 No Content");  
38         builder.buildHeaders(headers);  
39         builder.buildBody(null);  
40         return builder.build();  
41     }  
42     public static HttpResponse Forbidden(Map<String, String> headers) {  
43         builder.buildStatus("403 Forbidden");  
44         builder.buildHeaders(headers);  
45         builder.buildBody(null);  
46         return builder.build();  
47     }  
48 }  
49 }
```

Висновок: у цій лабораторній роботі ми ознайомилися з патернами програмування «ADAPTER», «BUILDER», «COMMAND», «CHAIN OF RESPONSIBILITY», «PROTOTYPE» та реалізували один з них.