



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  
З дисципліни: «Теорія розробки програмного забезпечення»

Download Manager  
Template Method

Виконав:  
студент групи ІА-14  
Юлдашев А.Х.

## Мета:

Реалізувати паттерн Template Method в проєкті на тему DownloadManager

Код з коментарями:

### AuthenticatedHttpDownloader

```
1 package service.template.impl;
2
3 import service.template.HttpDownloaderTemplate;
4
5 /**
6  * Клас, який реалізує шаблон HttpDownloaderTemplate для аутентифікованого HTTP-завантаження.
7  * Надає аутентифікацію для завантаження за допомогою імені користувача та пароля.
8  */
9 no usages  Kepka09 *
10 public class AuthenticatedHttpDownloader extends HttpDownloaderTemplate {
11     1 usage
12     private String username;
13     1 usage
14     private String password;
15
16     /**
17     * Конструктор класу.
18     *
19     * @param username Ім'я користувача для аутентифікації.
20     * @param password Пароль для аутентифікації.
21     */
22     no usages  Kepka09
23     public AuthenticatedHttpDownloader(String username, String password) {
24         this.username = username;
25         this.password = password;
26     }
27 }
```

```

24      /**
25       * Підготовка до завантаження з аутентифікацією.
26       * Перевизначений метод з базового класу.
27       */
28
29       1 usage  Kerka09
30       @Override
31       protected void prepareDownload() {
32           // Підготовка к загрузке с аутентификацией
33       }
34
35       /**
36       * Виконання завантаження з аутентифікацією.
37       * Перевизначений метод з базового класу.
38       *
39       * @param url URL-адреса файлу для завантаження.
40       */
41       1 usage  Kerka09
42       @Override
43       protected void executeDownload(String url) {
44           // Логика загрузки с аутентификацией
45       }
46
47

```

```

44      /**
45       * Завершення завантаження з аутентифікацією.
46       * Перевизначений метод з базового класу.
47       */
48
49       1 usage  Kerka09
50       @Override
51       protected void finalizeDownload() {
52           // Завершение загрузки с аутентификацией
53       }
54
55

```

## HttpDownloader

```
1 package service.template.impl;
2
3 import service.template.HttpDownloaderTemplate;
4
5 /**
6  * Клас, який реалізує шаблон HttpDownloaderTemplate для HTTP-завантаження.
7  * Надає реалізацію методів базового класу для завантаження файлів по HTTP.
8  */
9 no usages  Kerka09 *
10 public class HttpDownloader extends HttpDownloaderTemplate {
11     /**
12      * Підготовка до завантаження.
13      * Перевизначений метод з базового класу.
14      */
15     1 usage  Kerka09
16     @Override
17     protected void prepareDownload() {
18         // Логика для підготовки к загрузке (например, проверка URL)
19     }
```

```
18 /**
19  * Виконання завантаження файлу по HTTP.
20  * Перевизначений метод з базового класу.
21  *
22  * @param url URL-адреса файлу для завантаження.
23  */
24 1 usage  Kerka09
25  @Override
26  protected void executeDownload(String url) {
27      // Реальная логика для загрузки файла по HTTP
28  }
29 /**
30  * Завершення процесу завантаження.
31  * Перевизначений метод з базового класу.
32  */
33 1 usage  Kerka09
34  @Override
35  protected void finalizeDownload() {
36      // Завершающие действия (например, проверка целостности файла)
37  }
```

## TransactionalHttpDownloader

```
1 package service.template.impl;
2
3 import service.template.HttpDownloaderTemplate;
4
5 /**
6  * Клас, що реалізує шаблон HttpDownloaderTemplate для HTTP-завантаження з використанням транзакцій.
7  * Надає механізм для завантаження файлів по HTTP з використанням транзакцій, які не можна призупинити - тільки скасувати.
8  */
9 no usages 1 Керка09 *
10 public class TransactionalHttpDownloader extends HttpDownloaderTemplate {
11
12     //Завантажку нельзя остановить - только отменить
13
14     /**
15      * Підготовка до завантаження з використанням транзакцій.
16      * Перевизначений метод з базового класу.
17      */
18     1 usage 1 Керка09
19     @Override
20     protected void prepareDownload() {
21         // Подготовка к загрузке с транзакцией
22     }
```

```
22
23     /**
24      * Виконання завантаження з використанням транзакцій.
25      * Перевизначений метод з базового класу.
26      *
27      * @param url URL-адреса файлу для завантаження.
28      */
29     1 usage 1 Керка09
30     @Override
31     protected void executeDownload(String url) {
32         // Логика загрузки с транзакцией
33     }
34
35     /**
36      * Завершення завантаження з використанням транзакцій.
37      * Перевизначений метод з базового класу.
38      */
39     1 usage 1 Керка09
40     @Override
41     protected void finalizeDownload() {
42         // Завершение загрузки с транзакцией
43     }
44 }
```

## HttpDownloaderTemplate

```
1 package service.template;
2
3 /**
4  * Абстрактний клас, що визначає шаблонний метод для процесу HTTP-завантаження.
5  * Надає каркас для реалізації алгоритму завантаження файлів по HTTP.
6  */
7 6 usages 3 inheritors Kepka09
8 public abstract class HttpDownloaderTemplate {
9     // Шаблонний метод
10     no usages Kepka09
11     public final void download(String url) {
12         prepareDownload();
13         executeDownload(url);
14         finalizeDownload();
15     }
16
17     1 usage 3 implementations Kepka09
18     protected abstract void prepareDownload(); // Підготовка к загрузке
19     1 usage 3 implementations Kepka09
20     protected abstract void executeDownload(String url); // Выполнение загрузки
21     1 usage 3 implementations Kepka09
22     protected abstract void finalizeDownload(); // Завершение загрузки
23 }
24
```

Висновок: В даній лабораторній роботі була написана частина програми “Download Manager” з використанням патерну Template Method.

