



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 8
З дисципліни: «Теорія розробки програмного
забезпечення»

Download Manager
Composite

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

Мета:

Реалізувати паттерн Composite в проєкті на тему DownloadManager

Код з коментарями:

Download Observer

```
1 package com.example.downloadmanager0;
2
3 /**
4  * Інтерфейс спостерігача для завдань завантаження.
5  * Визначає один метод update, який повинен бути реалізований для оновлення стану спостерігача.
6  */
7
8 usages
9
10 public interface DownloadObserver {
11     /**
12      * Оновлює стан спостерігача на основі змін у завданні завантаження.
13      *
14      * @param task завдання завантаження, яке викликає оновлення
15      */
16     1 usage
17     void update(DownloadTask task);
18 }
```

Download Subject

```
1 package com.example.downloadmanager0;
2
3 /**
4  * Інтерфейс суб'єкта для моделі спостереження у системі завантаження.
5  * Визначає основні методи для управління спостерігачами.
6  */
7
8 1 usage 1 implementation
9
10 public interface DownloadSubject {
11     /**
12      * Приєднує спостерігача до суб'єкта.
13      *
14      * @param observer спостерігач, який потрібно приєднати
15      */
16     no usages 1 implementation
17     void attach(DownloadObserver observer);
18
19     /**
20      * Від'єднує спостерігача від суб'єкта.
21      *
22      * @param observer спостерігач, який потрібно від'єднати
23      */
24     no usages 1 implementation
25     void detach(DownloadObserver observer);
26
27     /**
28      * Повідомляє всіх приєднаних спостерігачів про зміну стану.
29      */
30     1 usage 1 implementation
31     void notifyObservers();
32 }
```

DownloadTask

```
1 package com.example.downloadmanager0;
2
3 /**
4  * Представляє окреме завдання завантаження у менеджері завантажень.
5  * Цей клас розширює DownloadQueueComponent та реалізує DownloadSubject для управління спостерігачами завантаження.
6  *
7  * @param observers список спостерігачів завантаження
8  * @param status поточний статус завдання завантаження
9  * @param fileName ім'я файлу для завантаження
10 * @param url URL завантажуваного файлу
11 * @param downloadSpeed швидкість завантаження (у кілобайтах за секунду)
12 * @param isInProgress прапорець, що вказує, чи триває завантаження
13 */
14
15 import lombok.AllArgsConstructor;
16 import service.composite.DownloadQueueComponent;
17
18 import java.util.List;
19
20 1 usage Kepka09
21 @AllArgsConstructor
22 public class DownloadTask extends DownloadQueueComponent implements DownloadSubject {
23
```

DownloadQueue

```
1 package service.composite;
2
3 import lombok.Getter;
4 import lombok.Setter;
5 import java.util.Queue;
6
7 /**
8  * Клас, що реалізує чергу завдань завантаження.
9  * Надає функціональність для управління чергою завдань і виконання операцій завантаження.
10 */
11 no usages Kepka09 *
12 @Getter
13 @Setter
14 public class DownloadQueue extends DownloadQueueComponent {
15
16     Queue<DownloadQueueComponent> downloadQueue;
17     private DownloadQueueComponent currentTask;
18     private DownloadStatus status;
19
20     /**
21      * Додає компонент завдання завантаження до черги.
22      *
23      * @param downloadTask завдання, яке потрібно додати
24      */
25
```

no usages Kepka09

```
24 public void addDownloadComponent(DownloadQueueComponent downloadTask) {  
25     | downloadQueue.add(downloadTask);  
26 }  
27
```

28 ≡ /**

```
29     * Видаляє компонент завдання завантаження з черги.  
30     *  
31     * @param downloadTask завдання, яке потрібно видалити  
32     */
```

no usages Kepka09 *

```
33 @ public void removeDownloadComponent(DownloadQueueComponent downloadTask) {  
34     | if (downloadTask.getStatus() == DownloadStatus.STARTED) {  
35     |     | downloadTask.cancel();  
36     | }  
37     | downloadQueue.remove(downloadTask);  
38 }  
39
```

```
40 // ... тут решта методів класу
```

42 /**

```
43     * Запускає поточне завдання завантаження у черзі.  
44     * Якщо поточне завдання завершене або скасоване, запускає наступне в черзі.  
45     */
```

no usages new *

```

46         @Override
47         public void start() {
48             // код методу
49         }
50
51         /**
52          * Призупиняє поточне завдання завантаження.
53          */
54         no usages  Керка09 *
55         @Override
56         public void pause() {
57             // код методу
58         }
59         /**
60          * Скасовує поточне завдання завантаження і очищає чергу.
61          */
62         1 usage  Керка09 *
63         @Override
64         public void cancel() {
65             // код методу
66         }

```

```

67         /**
68          * Повертає статус поточного завдання завантаження.
69          * Якщо поточне завдання відсутнє, повертає статус як COMPLETED.
70          *
71          * @return статус поточного завдання
72          */
73         1 usage  Керка09 * 2 related problems
74         @Override
75         public DownloadStatus getStatus() {
76             // код методу
77         }

```

Download Queue Component

```
1 package service.composite;
2
3 /**
4  * Абстрактний клас, що представляє компонент черги завантаження.
5  * Визначає основні методи для управління процесом завантаження.
6  */
7 7 usages 2 inheritors Kepka09 *
8 public abstract class DownloadQueueComponent {
9
10     no usages
11     DownloadStatus status;
12
13     /**
14     * Запускає процес завантаження.
15     */
16     no usages 2 implementations Kepka09
17     public abstract void start();
18
19     /**
20     * Призупиняє процес завантаження.
21     */
22     no usages 2 implementations Kepka09
23     public abstract void pause();
24
25     /**
26     * Скасовує процес завантаження.
27     */
28     1 usage 2 implementations Kepka09
29     public abstract void cancel();
30
31     /**
32     * Повертає поточний статус завантаження.
33     *
34     * @return поточний статус завантаження
35     */
36     1 usage 2 implementations Kepka09 2 related problems
37     public abstract DownloadStatus getStatus();
38 }
39
```

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

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

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

