



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 6
З дисципліни: «Теорія розробки програмного
забезпечення» Download Manager
Observer

Виконав:

студент групи ІА-14

Юлдашев А.Х.

Київ 2023

Мета:

Реалізувати паттерн Observer в проєкті на тему Download Manager

Код з коментарями:

StartDownloadObserver

```
1 package service.observers;
2
3 import model.observer.DownloadObserver;
4 import model.observer.DownloadTask;
5
6 /**
7  * Клас, який відповідає за спостереження за початком завантаження.
8  * Реалізує інтерфейс DownloadObserver.
9  */
10 no usages Kepka09 *
11 public class StartDownloadObserver implements DownloadObserver {
12     /**
13      * Метод, який оновлює спостережувача про початок завантаження.
14      *
15      * @param task Об'єкт завдання завантаження, що було оновлено.
16      */
17     @Override
18     public void update(DownloadTask task) {
19         //Todo do something if download is started
20     }
21 }
```

DownloadObserver

```
1 package model.observer;
2
3 /**
4  * Інтерфейс, що визначає метод оновлення спостережувача про події у завданнях завантаження.
5  * Класи, що реалізують цей інтерфейс, повинні реалізувати метод update(), щоб слідкувати за змінами у завданнях завантаження.
6  */
7 8 usages 1 implementation Kepka09 *
8 public interface DownloadObserver {
9     /**
10     * Метод для оновлення спостережувача про події у завданнях завантаження.
11     *
12     * @param task Об'єкт завдання завантаження, що був оновлений.
13     */
14     void update(DownloadTask task);
15 }
```

DownloadStatus

```
1 package model.observer;
2
3 /**
4  * Перерахування, що визначає можливі статуси завантаження.
5  * Містить константи для статусів: STARTED (почато), PAUSED (призупинено), COMPLETED (завершено),
6  * FAILED (не вдалося), CANCELLED (скасовано).
7  */
8 2 usages Kepka09
9 public enum DownloadStatus {
10     no usages
11     STARTED,
12     no usages
13     PAUSED,
14     no usages
15     COMPLETED,
16     no usages
17     FAILED,
18     no usages
19     CANCELLED
20 }
```

DownloadSubject

```
4      * Інтерфейс, який визначає методи для роботи зі спостережувачами завантаження.
5      * Класи, що реалізують цей інтерфейс, можуть додавати, видаляти спостережувачів
6      * та повідомляти їх про зміни у завантаженні.
7      */
1     1 usage  1 implementation  👤 Kepka09 *
8      public interface DownloadSubject {
9          /**
10         * Додає спостережувача для відстеження змін у завантаженні.
11         *
12         * @param observer Об'єкт спостережувача для додавання.
13         */
14         no usages  1 implementation  👤 Kepka09
15         void attach(DownloadObserver observer);
16         /**
17         * Видаляє спостережувача, що був раніше доданий для відстеження змін у завантаженні.
18         *
19         * @param observer Об'єкт спостережувача для видалення.
20         */
21         no usages  1 implementation  👤 Kepka09
22         void detach(DownloadObserver observer);
23         /**
24         * Повідомляє всіх спостережувачів про зміни у завантаженні.
25         */
26         1 usage  1 implementation  👤 Kepka09
27         void notifyObservers();
28     }
```

DownloadTask

```
53         /**
54         * Встановлює статус завантаження та сповіщає спостережувачів про зміну статусу.
55         *
56         * @param status Новий статус завантаження.
57         */
58         no usages  👤 Kepka09
59         public void setStatus(DownloadStatus status) {
60             this.status = status;
61             notifyObservers();
62         }
```

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

