



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  
**Теорія розробки програмного забезпечення**  
ШАБЛОН «MEDIATOR», «FACADE», «BRIDGE», «TEMPLATE  
METHOD».  
Тема: «Музичний програвач»

Виконав  
студент групи ІА–14:  
Галаган Є.В.

Київ 2023

## **Музичний програвач (iterator, command, memento, facade, visitor, client-server)**

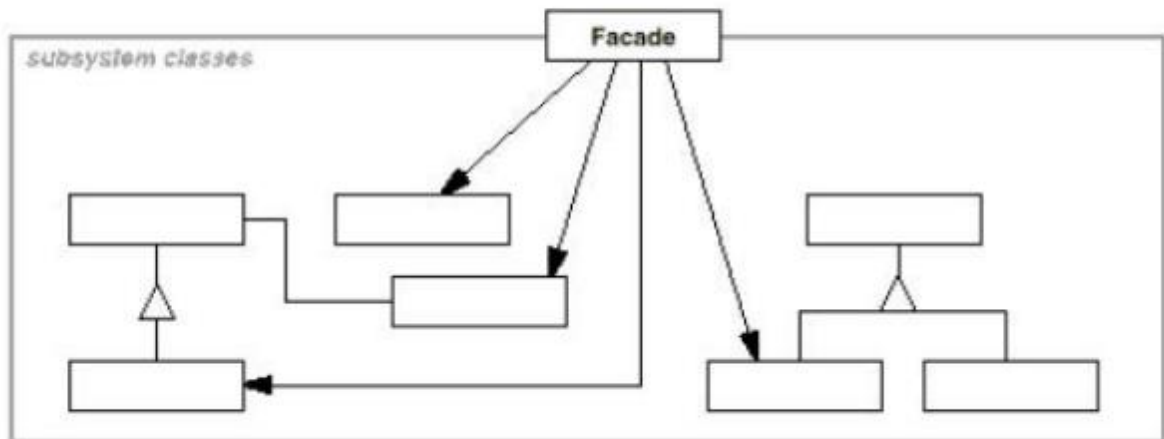
Музичний програвач становить собою програму для програвання музичних файлів або відтворення потокової музики з можливістю створення, запам'ятовування і редагування списків програвання, перемішування/повторення (shuffle/repeat), розпізнавання різних аудіоформатів, еквалайзер.

### **Завдання.**

- Ознайомитися з короткими теоретичними відомостями.
- Реалізувати частину функціонала робочої програми у вигляді класів і їх взаємодій для досягнення конкретних функціональних можливостей.
- Застосування одного з даних шаблонів при реалізації програми.

## Шаблон «Facade»

Структура:



```
1 usage 1 implementation  🧑 GeV8
public interface IMediaPlayerFacade {
    no usages 1 implementation  🧑 GeV8
    public void addSongToPlaylist(long idOfSong, long idOfPlaylist) throws SQLException;

    1 usage 1 implementation  🧑 GeV8
    public void addSong(String name, String path, String genre) throws SQLException;

    no usages 1 implementation  🧑 GeV8
    public void createEqualizer(String name, int volume, int bassBooster) throws SQLException;

    no usages 1 implementation  🧑 GeV8
    public void createPlaylist(String name) throws SQLException;
}
```

Интерфейс IMediaPlayerFacade

```

16 public class MediaPlayerFacade implements IMediaPlayerFacade {
17
18     3 usages
19     private final PlaylistService playlistService;
20     2 usages
21     private final SongService songService;
22     2 usages
23     private final EqualizerService equalizerService;
24
25     1 usage 1 GeV8
26     public MediaPlayerFacade(Connection connection) throws SQLException {
27         this.playlistService = new PlaylistService(new PlaylistRepository(connection));
28         this.songService = new SongService(new SongRepository(connection));
29         this.equalizerService = new EqualizerService(new EqualizerRepository(connection));
30     }
31
32     no usages 1 GeV8
33     @Override
34     public void createPlaylist(String name) throws SQLException {
35         playlistService.createPlaylist(new Playlist(name));
36     }
37
38     1 usage 1 GeV8
39     @Override
40     public void addSong(String name, String path, String genre) throws SQLException {
41         songService.addSong(new Song(name, path, genre));
42     }
43
44     no usages 1 GeV8
45     @Override
46     public void createEqualizer(String name, int volume, int bassBooster) throws SQLException {
47         equalizerService.addNewEqualizer(new Equalizer(name, volume, bassBooster));
48     }
49
50     no usages 1 GeV8
51     @Override
52     public void addSongToPlaylist(long idOfSong, long idOfPlaylist) throws SQLException {
53         playlistService.addSongToPlaylist(idOfPlaylist, idOfSong);
54     }
55 }

```

Класс MediaPlayerFacade, який імплементує інтерфейс