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 №9
Теорія розробки програмного забезпечення
РІЗНІ ВИДИ ВЗАЄМОДІЇ ДОДАТКІВ: CLIENT-SERVER, PEER-TO-PEER, SERVICE-ORIENTED ARCHITECTURE».
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.

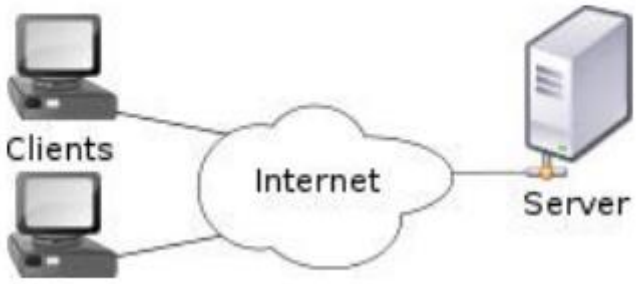
Завдання.

1. Ознайомитися з короткими теоретичними відомостями.

2. Реалізувати частину функціоналу робочої програми у вигляді класів і їх взаємодій для досягнення конкретних функціональних можливостей.

3. Реалізувати взаємодію програми в одній з архітектур відповідно до обраної теми.

Клієнт-серверні додатки.



Передбачає наявність сервера, та клієнта.

Клієнт надсилає запити на сервер.

Сервер на запит формує відповідь.

Клієнт отримує цю відповідь.

Реалізовано за допомогою Java Sockets.

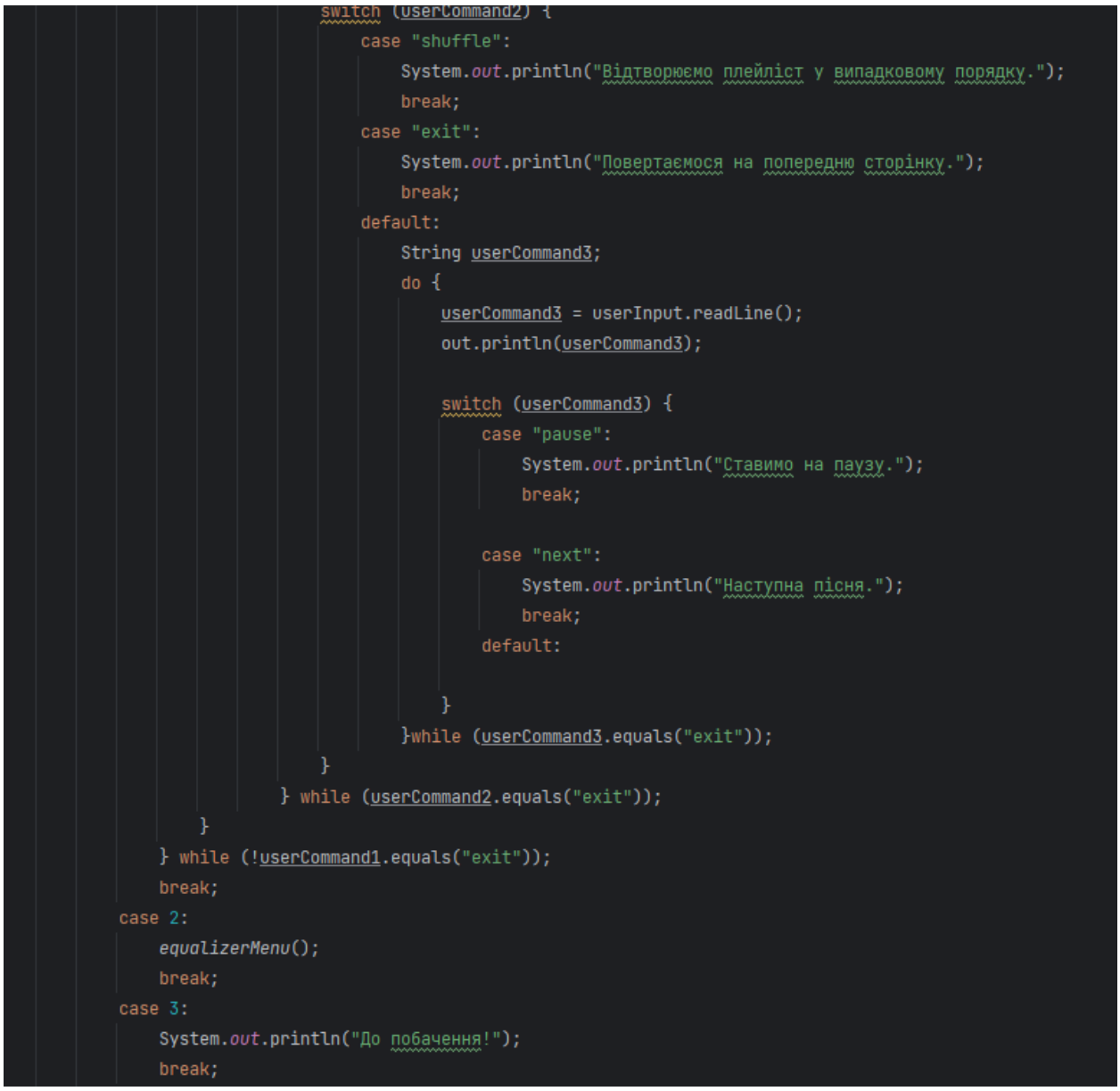
```java
GeV8 *
public class Main {
    1 usage
    private static final String url = "jdbc:mysql://localhost:3306/trpz_galagan ";
    1 usage
    private static final String user = "root";
    1 usage
    private static final String password = "159085";

    no usages
    static int port = 5003;

    GeV8 *
    public static void main(String[] args) {

        try(Connection connection = DriverManager.getConnection(url,user,password)) {
            ServerSocket serverSocket = new ServerSocket( port: 12345);
            Socket clientSocket = serverSocket.accept();
            BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
            PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), autoFlush: true);
            ObjectOutputStream objectOutputStream = new ObjectOutputStream(clientSocket.getOutputStream());

            MediaPlayerFacade mediaPlayerFacade = new MediaPlayerFacade(connection);
            System.out.println("Сервер запущено. Очікування клієнта...");
            System.out.println("Клієнт підключений.");
            String inputLine;
```

```java
while ((inputLine = in.readLine()) != null) {
    System.out.println("Клієнт відправив: " + inputLine);
    // Обробка введених значень клієнта
    switch (inputLine) {
        case "1":
            String nestedInput1;
            do{
                LinkedList<Playlist>playlistsToWrite = mediaPlayerFacade.getAllPlaylist();
                objectOutputStream.writeObject(playlistsToWrite);
                nestedInput1 = in.readLine();
                System.out.println(nestedInput1);
                // Вкладений switch-case
                switch (nestedInput1) {
                    case "exit":
                        break;
                    default:
                        System.out.println(nestedInput1);
                        Playlist playlist = mediaPlayerFacade.getChoosenPlaylist(Long.parseLong(nestedInput1));
                        objectOutputStream.writeObject(playlist);
                        String nestedInput2;
                        do{
                            nestedInput2=in.readLine();
                            switch (nestedInput2){
                                case "shuffle":
                                    break;
                                case "exit":
                                    break;

                                default:
                                    mediaPlayerFacade.playSong(Long.parseLong(nestedInput2));
                                    String nestedInput3;
                                    do{
                                        nestedInput3=in.readLine();
                                        switch (nestedInput3){
                                            case "pause":
                                                mediaPlayerFacade.pauseSong();
                                            case "exit":
                                                break;
                                            default:

                                        }
                                    }while (nestedInput3.equals("exit"));
```

Реалізація серверної частини

```java
import entity.Playlist;
import visitor.MediaPlayerVisitor;

import java.io.*;
import java.net.Socket;
import java.util.LinkedList;
import java.util.Objects;
import java.util.Scanner;

public class Main {
    no usages
    static int port = 5003;

    public static void main(String[] args) throws IOException {

        try {
            Socket socket = new Socket( host: "localhost",  port: 12345);
            BufferedReader userInput = new BufferedReader(new InputStreamReader(System.in));
            BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            PrintWriter out = new PrintWriter(socket.getOutputStream(),  autoFlush: true);
            ObjectInputStream objectInput = new ObjectInputStream(socket.getInputStream());
            MediaPlayerVisitor mediaPlayerVisitor =  new MediaPlayerVisitor();

            int userCommand;

            System.out.println("Підключено до сервера. Введіть значення (1, 2, 3 або exit для виходу):");

            do {
                System.out.println("Головне меню:");
                System.out.println("1. Отримати список плейлістів");
                System.out.println("2. Отримати список Еквалайзерів");
                System.out.println("3. Вийти");
                System.out.print("Введіть ваш вибір: ");
                userCommand = Integer.parseInt(userInput.readLine());
```

```
switch (userCommand) {
    case 1:
        out.println(userCommand);
        mediaPlayerVisitor.visitAllPlaylists((LinkedList<Playlist>) objectInput.readObject());

        String userCommand1;
        do {

            System.out.println("\nМеню плейлістів:");
            System.out.println("1. Обрати плейліст(id)");
            System.out.println("2. На головну");
            System.out.print("Введіть ваш вибір: ");
            userCommand1 = userInput.readLine();
            out.println(userCommand1);
            switch (userCommand1) {
                case "exit":
                    break;
                default:
                    Playlist playlist = (Playlist) objectInput.readObject();
                    if(playlist==null){
                        System.out.println("Не вірно введено айді");
                    }
                    mediaPlayerVisitor.visitPlaylist(playlist);
                    String userCommand2;
                    do {
                        System.out.println("\nМеню вибору плейлісту:");
                        System.out.println("1. Обрати пісню плейліста");
                        System.out.println("2. Шафл плейліст");
                        System.out.println("3. На попередню сторінку (Вибір плейлісту)");
                        System.out.print("Введіть ваш вибір: ");
                        userCommand2 = userInput.readLine();
                        out.println(userCommand2);
```



```
                        switch (userCommand2) {
                            case "shuffle":
                                System.out.println("Відтворюємо плейліст у випадковому порядку.");
                                break;
                            case "exit":
                                System.out.println("Повертаємося на попередню сторінку.");
                                break;
                            default:
                                String userCommand3;
                                do {
                                    userCommand3 = userInput.readLine();
                                    out.println(userCommand3);

                                    switch (userCommand3) {
                                        case "pause":
                                            System.out.println("Ставимо на паузу.");
                                            break;

                                        case "next":
                                            System.out.println("Наступна пісня.");
                                            break;
                                        default:

                                    }
                                }while (userCommand3.equals("exit"));
                        }
                    } while (userCommand2.equals("exit"));
                }
            } while (!userCommand1.equals("exit"));
            break;
        case 2:
            equalizerMenu();
            break;
        case 3:
            System.out.println("До побачення!");
            break;
```

Реалізація клієнтської частини