

Este programa Java consta de tres clases principales: DatabaseConnection, UserService y Main. Cada clase tiene una función específica para gestionar la conexión a una base de datos MySQL y realizar consultas sobre la tabla users. Este programa implementa el patrón Singleton en la clase DatabaseConnection. El patrón Singleton asegura que una clase tenga solo una instancia y proporciona un punto de acceso global a esa instancia. A continuación, se presenta una explicación detallada de cada clase y sus métodos.

## 1. Clase DatabaseConnection

La clase DatabaseConnection es una implementación del patrón Singleton que gestiona una única conexión a una base de datos MySQL.

```
public class DatabaseConnection {
    private static final DatabaseConnection instancia = new DatabaseConnection();
    private Connection connection;
    private static final String url = "jdbc:mysql://localhost:3306/mydatabase";
    private static final String username = "root";
    private static final String password = "Pa$$w0rd";

    private DatabaseConnection() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            connection = DriverManager.getConnection(url, username, password);
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
            throw new RuntimeException("Error al conectar a la base de datos", e);
        }
    }

    public static DatabaseConnection getInstancia() {
        return instancia;
    }

    public Connection getConnection() {
        return connection;
    }
}
```

### Atributos:

- **instancia:** Una instancia estática y final de DatabaseConnection para asegurar el patrón Singleton.
- **connection:** Un objeto Connection que representa la conexión a la base de datos.
- **url:** La URL de la base de datos.

- **username:** El nombre de usuario para la base de datos.
- **password:** La contraseña para la base de datos.

#### Métodos:

- **DatabaseConnection():** Constructor privado que establece la conexión a la base de datos.
- **getInstancia():** Devuelve la instancia única de DatabaseConnection.
- **getConnection():** Devuelve el objeto Connection.

## 2. Clase UserService

La clase UserService utiliza la clase DatabaseConnection para realizar consultas sobre la tabla users.

```
public class UserService {
    private DatabaseConnection manager;

    public UserService() {
        this.manager = DatabaseConnection.getInstancia();
    }

    public void printAllUsers() {
        Connection connection = manager.getConnection();
        try (Statement statement = connection.createStatement()) {
            ResultSet resultSet = statement.executeQuery("SELECT * FROM users");
            while (resultSet.next()) {
                System.out.println("User ID: " + resultSet.getInt("id"));
                System.out.println("User Name: " + resultSet.getString("name"));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

#### Atributos:

- **manager:** Una instancia de DatabaseConnection para gestionar la conexión a la base de datos.

#### Métodos:

- **UserService():** Constructor que obtiene la instancia única de DatabaseConnection.

- **printAllUsers():** Ejecuta una consulta SQL para obtener todos los usuarios de la tabla users y los imprime en la consola.

### 3. Clase Main

La clase Main contiene el método main, el punto de entrada del programa.

```
public class Main {  
    public static void main(String[] args) {  
        UserService userService = new UserService();  
        userService.printAllUsers();  
    }  
}
```

#### Métodos:

- **main(String[] args):** Método principal que crea una instancia de UserService y llama al método printAllUsers() para imprimir todos los usuarios de la tabla users.

#### Funcionamiento del Programa

1. **Conexión a la Base de Datos:** La clase DatabaseConnection se encarga de establecer y gestionar una conexión a la base de datos MySQL utilizando el patrón Singleton.
2. **Consulta y Recuperación de Datos:** La clase UserService utiliza la conexión proporcionada por DatabaseConnection para ejecutar una consulta SQL que recupera todos los registros de la tabla users.
3. **Impresión de Resultados:** Los resultados de la consulta se imprimen en la consola a través del método printAllUsers() de la clase UserService.
4. **Ejecución del Programa:** El método main en la clase Main inicia el proceso creando una instancia de UserService y llamando al método printAllUsers().

Este programa es útil para ilustrar cómo gestionar conexiones a bases de datos en Java utilizando el patrón Singleton y cómo realizar consultas básicas a una base de datos MySQL.