CS202 FALL 2019
Assignment 1 : Room Reservation
DUE: 04.11.2019 23:55 – For your questions, TA: emir.arditi@ozu.edu.tr

This assignment covers querying a SQL database using Java JDBC. When writing your program, you should create a new schema/database in your MySQL server just for this assignment. You can create the tables in Workbench, but their instances, and select queries should be created, executed, and displayed programmatically in Java.

This assignment is informally made up of 3 sub-steps.

1) Design the relation schemas for the below scenario and create tables in your MySQL server.
2) Add data into these relations, update and retrieve data from these relations in your Java program via JDBC.
3) Implement a console interface to run the queries listed below.

You should write a **Java standalone console program**, which allows organizations to reserve locations for organizing events such as a concert, a meeting or a sports event. Each location should have different dynamics. For example, you can create a new concert organization in a football stadium; but you cannot create a football game in Istanbul Congress Center. The eligible event types for each location should be specified during creation of the location. In order to make things simple (or complicated :)) **do not** use timestamps and date/time data types in this assignment. Instead use three fields (day, start, end) to represent the time of reservation. **`Start`** and **`end`** should be between 0-24 denoting the working hours, and **`end`** field cannot be less than or equal to the start field. An already reserved location should not be reserved by another request unless it is cancelled. The user should be able to run the following queries from the command line. To achieve this, please prepare a menu displaying the options and associate a number with each option so that when the user enters the number for the chosen action,the program can gather the necessary information from the user and perform the chosen task by modifying/retrieving data in your database.

1. Add Organizer
2. Remove Organizer
3. Add Location
4. Remove Location
5. Make reservation
6. Cancel reservation

7. List all reservations
8. List all reservations of a Location
9. List all reservations of an Organizer
10. Show Organizer of a particular reservation
11. List all reservations of a Particular Event Type(Concert, Meeting or Sports Organization)

Please implement the commands in console interface **in the same order** as it is written in this document to make it easier testing of your program. You should name your Java class file, which contains the main method as "**EventReservationSystem.java**".

In addition to your Java source file, you should submit a report (word document) indicating your data definition queries (create/alter table) and showing your database design (relations and their relationships).

SOME HINTS:

Connecting a database through Java by using JDBC:

```
static final String DB_URL = "jdbc:mysql://localhost:3306/cs202";
static final String USER = "emir";
static final String PASS = "12345";
```

```
/* It is highly recommended to put the following code inside a try-catch
structure. */

    Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
```

Executing an "insert" statement

```
Statement stmt = conn.createStatement();
// Example Insert Query
String sql = "INSERT INTO Organizers (etc., etc., etc.) " + "VALUES
       ('etc.', 'etc.', etc.)";
stmt.executeUpdate(sql);
```

Retrieving data using "select" statement"

```
String sql = "SELECT * FROM Organizers";
  try (ResultSet rs = stmt.executeQuery(sql)) {
     while (rs.next()) {
          System.out.print("ID: " + rs.getInt("id"));
          System.out.print(", Name: " + rs.getString("name"));
     }
  }
```

Closing a connection

```
if (conn != null)
     conn.close();
```