**Java реалізація**

/\*\*

\* **@author** Maxym Vlasov

\* **@version** 1.0

\*/

**public** **interface** IActionType {

**void** setActionId(**int** id);

**void** setActionName(String name);

}

/\*\*

\* **@author** Maxym Vlasov

\* **@version** 1.0

\*/

**public** **interface** IBindingToTag **extends** IObject, IDefaultUser, ITag{

}

/\*\*

\* **@author** Maxym Vlasov

\* **@version** 1.0

\*/

**public** **interface** IDefaultUser **extends** IGroup {

**void** setDefaultUserId(**int** id);

}

/\*\*

\* **@author** Maxym Vlasov

\* **@version** 1.0

\*/

**public** **interface** IGroup {

**void** setGroupId(**int** id);

**void** setGroupTitle(String title);

}

/\*\*

\* **@author** Maxym Vlasov

\* **@version** 1.0

\*/

**import** java.text.DateFormat;

**import** java.text.SimpleDateFormat;

**import** java.util.Date;

**public** **interface** IAction **extends** IObject, IMessage, IActionType, IDefaultUser {

DateFormat ***date*** = **new** SimpleDateFormat("yyyy/MM/dd HH:mm:ss");

**public** **default** DateFormat getDate() {

**return** ***date***;

}

}

/\*\*

\* **@author** Maxym Vlasov

\* **@version** 1.0

\*/

**public** **interface** IMessage {

**void** setMessageId(**int** id);

**void** setMessagePublicized(**boolean** publicized);

**void** setMessagePhotoURL(String photoURL);

**void** setMessageCommentary(String commentary);

}

/\*\*

\* **@author** Maxym Vlasov

\* **@version** 1.0

\*/

**import** org.json.\*;

**public** **interface** IObject {

**void** setObjectId(**int** id);

**void** setObjectName(String name);

**void** setObjectPublicized(**boolean** publicized);

**void** setObjectGeometry(Object geometry);

}

/\*\*

\* **@author** Maxym Vlasov

\* **@version** 1.0

\*/

**public** **interface** ITag {

**void** setTagId(**int** id);

**void** setTagName(String name);

}

/\*\*

\* **@author** Maxym Vlasov

\* **@version** 1.0

\*/

**public** **class** Object **extends** ObjectDao **implements** IObject {

**private** **int** id;

**private** String name;

**private** **boolean** publicized;

**private** Object geometry;

/\*\*

\* Constructor for Geometry

\* **@version** 0.1

\* **@param** string

\*/

**public** Object(String string) {

String Object = string;

}

/\*\*

\* Constructor for create Object in class 'Run'

\*/

**public** Object() {

}

@Override

**public** **void** setObjectId(**int** id) {

**this**.id = id;

}

@Override

**public** **void** setObjectName(String name) {

**this**.name = name;

}

@Override

**public** **void** setObjectPublicized(**boolean** publicized) {

**this**.publicized = publicized;

}

@Override

**public** **void** setObjectGeometry(Object geometry) {

**this**.geometry = geometry;

}

**public** **int** getId() {

**return** id;

}

**public** String getName() {

**return** name;

}

**public** **boolean** getPublicized() {

**return** publicized;

}

**public** Object getGeometry() {

**return** geometry;

}

}

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.PreparedStatement;

**import** java.util.ArrayList;

**import** java.util.List;

/\*\*

\* **@author** Maxym Vlasov

\* **@version** 1.0

\*/

**public** **class** ObjectDao {

// Need to connect to the database and CRUD-operations

**public** **static** String *url* = "jdbc:mysql://127.0.0.1:3306/Map\_of\_KPI";

**public** **static** String *name* = "root";

**public** **static** String *password* = "1111";

**public** **static** Connection *con* = *ConnectToDB*(*url*, *name*, *password*); // Are you

// Con,

// Bill?

// (:

/\*\*

\* CRUD: Creating an object in the database

\*

\* **@param** obj

\* **@return** obj

\* **@throws** SQLException

\*/

**public** **static** Object create(Object obj) **throws** SQLException {

String sql = "INSERT INTO Object (object\_id, name, publicized, geometry) VALUES (?,?,?,?);";

**try** (PreparedStatement ps = *con*.prepareStatement(sql)) {

/\*

\* This block is required in order to avoid errors when entering the

\* same values in the "ps" and "obj"

\*/

**int** tempId = 1;

String tempName = "Obj1";

**boolean** tempPublicized = **false**;

Object tempGeometry = **new** Object("125.6, 10.1");

ps.setInt(1, tempId);

ps.setString(2, tempName);

ps.setBoolean(3, tempPublicized);

ps.setObject(4, "some coordinates");

ps.executeUpdate();

obj.setObjectId(tempId);

obj.setObjectName(tempName);

obj.setObjectPublicized(tempPublicized);

obj.setObjectGeometry(tempGeometry);

}

**return** obj;

}

/\*\*

\* CRUD: Reading an object in the database

\*

\* **@param** obj

\* **@param** id

\* **@return** obj

\* **@throws** SQLException

\*/

**public** **static** Object read(Object obj, **int** id) **throws** SQLException {

String sql = "SELECT \* FROM Object WHERE object\_id = ?;";

**try** (PreparedStatement ps = *con*.prepareStatement(sql)) {

ps.setInt(1, id);

ResultSet rs = ps.executeQuery();

rs.next();

obj.setObjectId(rs.getInt("object\_id"));

obj.setObjectName(rs.getString("name"));

obj.setObjectPublicized(rs.getBoolean("publicized"));

obj.setObjectGeometry((Object) rs.getObject("geometry"));

}

System.***out***.println("Object id: " + obj.getId());

System.***out***.println("Name: " + obj.getName());

System.***out***.println("Publicized: " + obj.getPublicized());

System.***out***.println("Geometry: " + obj.getGeometry());

**return** obj;

}

/\*\*

\* CRUD: Updating an object in the database

\*

\* **@param** obj

\* **@param** id

\* **@throws** SQLException

\*/

**public** **static** **void** update(Object obj, **int** id) **throws** SQLException {

String sql = "UPDATE Object SET name = ?, publicized = ?,"

+ " geometry = ? WHERE object\_id = ?;";

**try** (PreparedStatement ps = *con*.prepareStatement(sql)) {

/\*

\* This block is required in order to avoid errors when entering the

\* same values in the "ps" and "obj"

\*/

String tempName = "Name of object";

**boolean** tempPublicized = **true**;

Object tempGeometry = **new** Object("137.1 24.4");

ps.setString(1, tempName);

ps.setBoolean(2, tempPublicized);

ps.setObject(3, tempGeometry);

obj.setObjectName(tempName);

obj.setObjectPublicized(tempPublicized);

obj.setObjectGeometry(tempGeometry);

}

**return**;

}

/\*\*

\* CRUD: Deleting an object in the database

\*

\* **@param** obj

\* **@throws** SQLException

\*/

**public** **static** **void** delete(Object obj) **throws** SQLException {

String sql = "DELETE FROM Object WHERE object\_id = ?;";

**try** (PreparedStatement ps = *con*.prepareStatement(sql)) {

ps.setInt(1, 4);

ps.executeUpdate();

obj.setObjectPublicized(**false**);

}

**return**;

}

/\*\*

\* Displaying all objects from the database

\*

\* **@param** obj

\* **@return** list of objects

\* **@throws** SQLException

\*/

**public** **static** List<Object> readAll(Object obj) **throws** SQLException {

String sql = "SELECT \* FROM Map\_of\_KPI.Object;";

List<Object> list = **new** ArrayList<Object>();

**try** (PreparedStatement ps = *con*.prepareStatement(sql)) {

ResultSet rs = ps.executeQuery();

**while** (rs.next()) {

obj.setObjectId(rs.getInt("object\_id"));

obj.setObjectName(rs.getString("name"));

obj.setObjectPublicized(rs.getBoolean("publicized"));

obj.setObjectGeometry((Object) rs.getObject("geometry"));

list.add(obj);

System.***out***.println("\n");

System.***out***.println("Object id: " + obj.getId());

System.***out***.println("Name: " + obj.getName());

System.***out***.println("Publicized: " + obj.getPublicized());

System.***out***.println("Geometry: " + obj.getGeometry());

}

System.***out***.println("\n");

**return** list;

}

}

/\*\*

\* Connect to the database

\*

\* **@param** url

\* **@param** name

\* **@param** password

\* **@return** null

\*/

**public** **static** Connection ConnectToDB(String url, String name,

String password) {

**try** {

Class.*forName*("com.mysql.jdbc.Driver");

Connection con = DriverManager.*getConnection*(url, name, password);

System.***out***.println("Connected.");

**return** con;

} **catch** (SQLException | ClassNotFoundException e) {

e.printStackTrace();

}

**return** **null**;

}

/\*\*

\* Close connection

\*

\* **@throws** SQLException

\*/

**public** **static** **void** End() **throws** SQLException {

*con*.close();

}

}

/\*\*

\* **@author** Maxym Vlasov

\* **@version** 1.0

\*/

**public** **class** Run **extends** Object {

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

Object objectT = **new** Object();

*create*(objectT);

*readAll*(objectT);

*update*(objectT, 1);

*read*(objectT, 1);

*create*(objectT);

*read*(objectT, 2);

*delete*(objectT);

*readAll*(objectT);

*End*();

}

}