DAM. UNIT 3. ACCESS USING OBJECT-RELATIONAL MAPPING (ORM). ASSESSABLE TASK 3

DAM. Acceso a Datos (ADA) (a distancia en inglés)

Unit 3. ACCESS USING OBJECT-RELATIONAL MAPPING (ORM)

Assessable Task 3

Abelardo Martínez

Year 2023-2024

Aspects to bear in mind

Important

> **If you look for the solutions surfing the Internet or asking the oracle of ChatGPT you will be fooling yourself**. Keep in mind that **ChatGPT is not infallible or all-powerful**.
>
> It is a great tool to speed up your work once you have mastered a subject, but using it as a shortcut when acquiring basic skills and knowledge seriously undermines your learning. If you use it to get solutions or advice on your own, check the proposed solutions carefully as well. Try to solve the activities using the resources we have seen and the extended documentation you will find in the "Virtual Classroom".

Tips for programming

We advice to follow the next coding standards:

- One instruction per line.
- Add comments to make your code clearer and more readable.
- Use the Hungarian notation to recognise the type of variables at first sight.
- Remember that there are several ways to implement a solution, so choose the one you like best. **We strongly recommend using buffer-based solutions**.

A. Instructions and guidelines

> **The project MUST be carried out in Java**. Other technologies -such as Spring Boot- will not be supported. Any of the IDEs proposed in unit 1 can be used for its development, although **Eclipse is strongly recommended**.

1. OVERVIEW

You are required to create a Java application **on your own** that utilises concepts taught during **UNIT 3** to meet a provided specification.

2.TIMELINE AND EXPECTATIONS

- **Percentages within the TERM**: 50% of TERM total (AT4 would make the other 50%)
- **Percentages within the TASK**: 100% ADA skills (English skills must be PASSED).
- **Due/Deadline**: **11:59pm on Sunday, 7th January, 2024** (3 WEEKS)

3. GRADING

You must get 5 marks out of 10 in ADA and a COMPETENT in English to pass this ASSESSABLE TASK.

A detailed grading scale will be providing with this document (check LEARNING RUBRIC).

4.RESOURCES

You should make a comprehensive reading of all the materials provided by your teacher as well as the non-assessable tasks, but also dive the Internet to find examples which provide similar outcomes to the ones required by this task.

Feel free to copy & paste code from ANY resource as long as you understand every piece of it since you will be required to defend your work in an individual meeting.

5. PLAGIARISM

You must not allow other students to copy your work and must take care to safeguard against this happening.

In case of suspected plagiarism, an additional oral interview might be required.

6. HANDING AND FEEDBACK

- **The task will be delivered ONLY in a ZIP format file, compressing the project folder from your IDE** (i.e. Eclipse).
- Afterwards, **you WILL BE REQUIRED to attend an oral interview** with your teacher to discuss certain aspects of your task in English for a maximum of 15 minutes.

- You will receive your marks broken down by each criteria, and the total, together with any comments giving suggestions on how you could have done better.

B. Assessment details

> **ONLY ENGLISH IS ALLOWED** for the implementation of the assessable task, both comments and explanatory/clarifying texts.

1. **EVERY METHOD MUST BE PROPERLY DESCRIBED IN YOUR OWN WORDS**. At the beginning of each method you must add comments to explain in your own words how it works.
2. **ALSO, YOU MUST ADD A TEXT EXPLAINING IN YOUR OWN WORDS, YOUR EXPERIENCE IMPLEMENTING THIS SOLUTION**.

Create a text file and copy it into the project folder or create the text file within the project itself in the Eclipse IDE.

- **PARAGRAPH 1**. Describe briefly the solution provided.
- **PARAGRAPH 2**. Describe briefly the difficulties found.
- **PARAGRAPH 3**. Describe briefly several possible extensions you recommended.

B.1. Mandatory features

Activity (ASSESSABLE)

Create a program in Java to manage a database of CHESS TOURNAMENTS as shown in the Entity-Relationship diagram you can find below by printing and using a specific menu. After each option, the user should see the same menu until option zero is pressed. **You can reuse the menu you built for AT1&AT2. Feel free to duplicate the code and apply the required changes**.

Please, do follow these TECHNICAL SPECIFICATIONS:

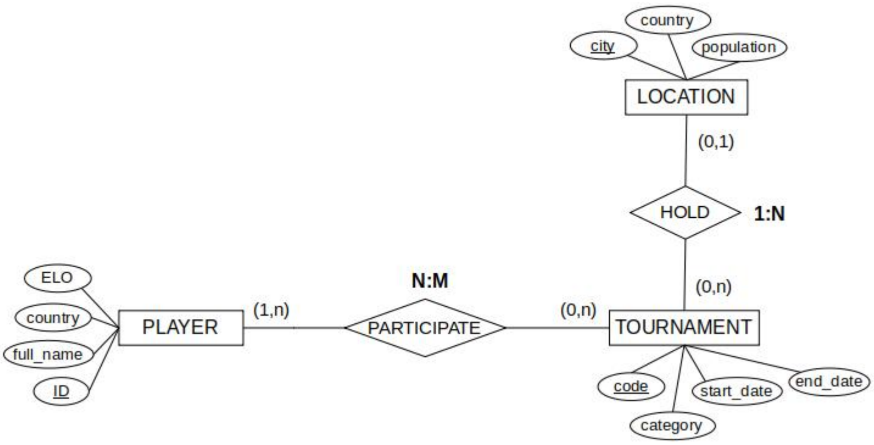
- RDBMS: **MySQL**
- Language: **Java**
- Framework: **Maven**
- ORM: **Hibernate with annotations**
- DAO: **POJO**

ATTENTION: Use the proper exceptions when accessing to databases via Hibernate.

---

1) Entity-Relationship diagram and DDL sentences:

CHESS TOURNAMENTS



```
DROP DATABASE IF EXISTS DBChessTournaments;
CREATE DATABASE DBChessTournaments CHARACTER SET utf8 COLLATE utf8_spanish_ci;

CREATE USER 'mavenuser'@'localhost' IDENTIFIED WITH mysql_native_password BY '
GRANT ALL PRIVILEGES ON DBChessTournaments.* to 'mavenuser'@'localhost';

USE DBChessTournaments;

DROP TABLE IF EXISTS Game;
DROP TABLE IF EXISTS Tournament;
DROP TABLE IF EXISTS Player;
DROP TABLE IF EXISTS Location;

-- Player
CREATE TABLE Player (
playerID    INTEGER,
fullname    VARCHAR(100),
country     VARCHAR(50),
ELO         INTEGER,
CONSTRAINT pla_id_pk PRIMARY KEY (playerID),
CONSTRAINT pla_elo_ck CHECK (ELO > 0)
);

-- Location
CREATE TABLE Location (
city        VARCHAR(100),
country     VARCHAR(50),
population  INTEGER,
CONSTRAINT loc_cit_pk PRIMARY KEY (city),
CONSTRAINT loc_pop_ck CHECK (population > 0)
);

-- Tournament
CREATE TABLE Tournament (
code        VARCHAR(10),
category    VARCHAR(20),
start_date  TIMESTAMP,
end_date    TIMESTAMP,
city        VARCHAR(100),
CONSTRAINT tou_cod_pk PRIMARY KEY (code),
CONSTRAINT tou_cat_ck CHECK (category IN ('AMATEUR','PROFESSIONAL','MASTER','S
CONSTRAINT tou_cit_fk FOREIGN KEY (city) REFERENCES Location(city) ON UPDATE S
);

-- Game
CREATE TABLE Game (
playerID    INTEGER,
code        VARCHAR(10),
CONSTRAINT gam_pco_pk PRIMARY KEY (playerID, code),
CONSTRAINT FOREIGN KEY (playerID) REFERENCES Player(playerID) ON DELETE CASCAD
CONSTRAINT FOREIGN KEY (code) REFERENCES Tournament(code) ON DELETE CASCADE
);
```

2) Menu options:

- **Press 0 to "Exit"**
- **Press 1 to "Insert & List chess players"**
    - This option will ask for items in loop until zero is entered.
    - For every item we need player ID (Integer), full name (String with spaces), country (String with spaces) and ELO (Integer > 0).
    - For every item given, we will store it in an ArrayList. Once zero is entered as a player ID, all items will be inserted into the PLAYER table and a list of players will be shown. Before it, you should check if the ArrayList is empty to avoid executing unnecessary code.
    - **Check if the chess player ID already exists in the array list. If yes, you must display a message on the screen. You must ask for each value (in loop) until the user enters a valid ID**.
    - **ATTENTION**: player ID and ELO must be an INTEGER! For every PLAYER, you must ask for each value (in loop) until the user enters a valid integer.
- **Press 2 to "Insert & List tournaments"**
    - This option will ask for items in loop until zero is entered.
    - For every item we need code (String without spaces), category (String without spaces), city (String with spaces), start date (DateTime) and end date (DateTime).
        - Also, for every item, we need to be able to associate as many chess players as we want, until zero is entered as a player ID (the player ID typed different than zero must exists).
    - For every item given, we will store it in an ArrayList. Once zero is entered as a code, all items will be inserted into the TOURNAMENT table and a list of tournaments will be shown. Before it, you should check if the ArrayList is empty to avoid executing unnecessary code.
    - **Check if the tournament code already exists in the array list. If yes, you must display a message on the screen. You must ask for each value (in loop) until the user enters a valid code**.
    - **ATTENTION**: start date and end date must be DateTime! For every TOURNAMENT,

you must ask for each value (in loop) until the user enters a valid date.

- **Press 3 to "Insert & List locations"**
  - This option will ask for items in loop until zero is entered.
  - For every item we need city (String with spaces), country (String with spaces) and population (Integer > 0).
    - Also, for every item, we need to associate as many tournaments as we want, until zero is entered as a code (the code typed different than zero must exists).
  - For every item given, we will store it in an ArrayList. Once zero is entered as a code, all items will be inserted into the LOCATION table and a list of locations will be shown. Before it, you should check if the ArrayList is empty to avoid executing unnecessary code.
  - **Check if the city already exists in the array list. If yes, you must display a message on the screen. You must ask for each value (in loop) until the user enters a valid city**.
  - **ATTENTION**: population must be an INTEGER! For every LOCATION, you must ask for each value (in loop) until the user enters a valid integer.
- **Press 4 to "Delete chess players"**
  - This option will ask for items in loop until zero is entered.
  - For every item given, we will store it in an ArrayList. Once zero is entered as a player ID, all items will be (cascade) deleted in table PLAYER (and associated tables).
- **Press 5 to "Delete tournaments"**
  - This option will ask for items in loop until zero is entered.
  - For every item given, we will store it in an ArrayList. Once zero is entered as a code, all items will be (cascade) deleted in table TOURNAMENT (and associated tables).
- **Press 6 to "Delete locations"**
  - This option will ask for items in loop until zero is entered.
  - For every item given, we will store it in an ArrayList. Once zero is entered as a city, all items will be (cascade) deleted in table LOCATION and (cascade) updated in associated tables.

Menu example:

```
*****
MENU
*****
========================================================================
  0. Exit
  1. Insert & List chess players
  2. Insert & List tournaments
  3. Insert & List locations
  4. Delete chess players
  5. Delete tournaments
  6. Delete locations
  7. [optional] Find a chess player
  8. [optional] Find a tournament
========================================================================
    Select an option:
```

B.2. Optional features

Activity (ASSESSABLE)

Optionally, you can implement these following entries within the menu to reach more than 8 marks out of 10 at this ASSESSABLE TASK.

Please, do follow these TECHNICAL SPECIFICATIONS:

- DQL (Data Query Language): **HQL criteria**

Menu options:

- **Press 7 to “[optional] Find a chess player”**
  - This option will ask for a number (1-9) or zero to back to the menu.
  - With that number, a list of chess players with player ID starting with that number will be shown, order by player ID ascending.
- **Press 8 to “[optional] Find a tournament”**
  - This option will ask for a letter (A-Z) or zero to back to the menu.
  - With that letter, a list of tournaments with code starting with that letter will be shown, order by code ascending.

C. Learning Rubric

C.1. ADA skills

Minimum of 5 out of 10 required for this part.

These marks will be invalidated (mark 4) if you fail to defend your work in an oral interview.

ASSESSMENT ITEMS		ASSESSMENT ITEM DETAILS	SCORE (POINTS)
Hibernate/Maven		Hibernate/Maven are set-up properly.	0.5
Menu		The menu complies with the specifications.	0.5
POJO objects		Defines and uses the right Hibernate annotations. Configures the ORM mapping correctly.	2
CRUD operations	Insert data	Inserts the data properly. Writes the data properly to MySQL.	2
	List data	Reads the data properly from MySQL. Lists/prints the data in a proper way.	1.5
	Delete data	Deletes the data properly from MySQL. Modifies the data properly.	1.5
[optional] Find a chess player			1
[optional] Find a tournament			1

C.2. English skills

Mandatory to be COMPETENT to pass this part.

ASSESSMENT ITEMS	ASSESSMENT ITEM DETAILS	SCORE
Writing skills	Every method is described properly. A proper text is provided (within the code or in a text file) to describe the AT using THREE PARAGRAPHS.	COMPETENT/NOT COMPETENT
Oral skills	Uses a vocabulary appropriate for the purpose. Shows fluency and confidence.	COMPETENT/NOT COMPETENT
Comprehension skills		Accomplished since all materials are in English
Reading skills		Accomplished since all materials are in English