# DAM. UNIT 2. ACCESS TO DATABASES. ASSESSABLE TASK 2

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

### Unit 2. ACCESS TO DATABASES

**Assessable Task 2**

**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 2 (Week1-Week2) to meet a provided specification.

## 2.TIMELINE AND EXPECTATIONS

- **Percentages within the TERM**: 50% of TERM total (AT1 would make the other 50%)
- **Percentages within the TASK**: 100% ADA skills (English skills must be PASSED).
- **Due/Deadline**: **11:59pm on Sunday, 12th November, 2023** (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 CHESS PLAYERS in a Chess Tournament by printing and using a specific menu. After each option, the user should see the same menu until option zero is pressed. **This is a continuation of the AT1. Feel free to duplicate the code and apply the required changes**.

**ATTENTION**: Use the proper exceptions when accessing to databases.

**Menu options**:

- **Press 0 to “Exit”**

- **Press 1 to "Get chess players and scores (to SQLite)"**
  - For every CHESS PLAYER we need player ID (String), full name (String with spaces), country (String with spaces), score/rating of game 1 (score1, Float), score/rating of game 2 (score2, Float) and score/rating of game 3 (score3, Float), added to an ArrayList of CHESS PLAYERS. A player who wins his/her game, or who wins by default, receives one point (1); a player who loses his/her game, or who loses by default, receives zero points (0); and a player who ends his/her game in a draw receives half a point (½). Therefore, there are only 3 possible score values: 0, 0.5 and 1 point.
  - **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.
  - Once zero is entered as ID, all CHESS PLAYERS will be **inserted in a SQLite database**, table CHESSPLAYERS, **dropping the table before inserting the chess players**. Before saving, you should check if the ArrayList of CHESS PLAYERS is empty to avoid executing unnecessary code.
  - **ATTENTION**: score1, score2 and score3 must be FLOAT! For every CHESS PLAYER, you must ask for each value (in loop) until the user enters a valid float.
  - **ATTENTION**: the **total score is not mandatory in this AT**.
- **Press 2 to "List all chess players (from SQLite)"**
  - Just read the **SQLite database** and print every CHESS PLAYER information.
- **Press 3 to "Delete all chess players (from SQLite)"**
  - Delete all chess players within CHESSPLAYERS table.

**Menu example**:

```
*****
MENU
*****
================================================================================
  0. Exit
  1. Get chess players and scores (to SQLite)
  2. List all chess players (from SQLite)
  3. Delete all chess players (from SQLite)
```

```
  4. [optional] Dump data from SQLite to MongoDB
  5. [optional] List all chess players (from MongoDB)
========================================================================
    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.

**ATTENTION**: Use the proper exceptions when accessing to databases.

**Menu options**:

- **Press 4 to "[optional] Dump data from SQLite to MongoDB"**
  - Create a **MongoDB database** and dump the information from SQLite.
- **Press 5 to "[optional] List all chess players (from MongoDB)"**
  - Just read the **MongoDB database** and print every CHESS PLAYER information.

# 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) |
|---|---|---|
| Classes and methods | Classes and methods are structured properly. | 0.5 |
| Menu | The menu complies with the specifications. | 0.5 |
| Get chess players and marks (to SQLite) | The ArrayList is populated properly.<br>Creates and manages the SQLite database structure properly.<br>Writes the data properly to SQLite. | 3 |
| List chess players (from SQLite) | Reads the data properly from SQLite.<br>Prints the data in a proper way. | 2 |
| Delete chess players (from SQLite) | Deletes the data properly from SQLite. | 2 |
| [optional] Dump data from SQLite to MongoDB | | 1 |
| [optional] List chess players from MongoDB | | 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.<br>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.<br>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 |