

# **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 2024-2025**

## 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.
- If necessary, we strongly recommend using buffer-based solutions.
- Remember that there are several ways to implement a solution, so choose the one you like best.

# 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** 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, 10th November, 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). The ZIP file should be named **"at2\_yourname.zip"**, where the suffix refers to your name.
- 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 called "**comments**" (in txt or pdf format) 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 **TAX INSPECTORS** in a tax management 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 tax inspectors and commissions (to SQLite)”**
  - For every **TAX INSPECTOR** we need **inspector ID** (String without spaces), **firstname** (String with spaces), **lastname** (String with spaces), **birthday** (LocalDate in format **dd/MM/yyyy**), **commission** (Float) and **cheated taxpayers** (Integer), added to an **ArrayList** of **TAX INSPECTORS**. Every tax **inspector** has a **commission**, which means what it charges for fleecing the long-suffering taxpayers.
  - **Check if the tax inspector 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 **TAX INSPECTORS** will be **inserted in a SQLite database**, table **"Taxinspector"**, **dropping the table before inserting the tax inspectors**. In addition, you must **ensure** that the **date** is **saved** in the **correct format** (**dd/MM/yyyy**). Before saving, you should **check** if the **ArrayList** of **TAX INSPECTORS** is **empty** to **avoid** executing **unnecessary code**.
  - **ATTENTION:** For every **TAX INSPECTOR**, you must **ask** for **each value** (in **loop**) **until** the user enters a **valid data**.
  - **ATTENTION:** For every **TAX INSPECTOR**, the **commission percentage** shall be **between 0 and 30**.

- **Press 2 to “List all tax inspectors (from SQLite)”**

- Just **read** the **SQLite database** and **print** every **TAX INSPECTOR** information.

- **Press 3 to “Delete all tax inspectors (from SQLite)”**

- **Delete** all **tax inspectors** within **"Taxinspector"** table.

**Menu example:**

```
*****
```

```
MENU
```

```
*****
```

```
=====
```

- 0. Exit
- 1. Get tax inspectors and commissions (to SQLite)
- 2. List all tax inspectors (from SQLite)
- 3. Delete all tax inspectors (from SQLite)
- 4. [optional] Dump data from SQLite to MongoDB
- 5. [optional] List all tax inspectors (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. In addition, you must ensure that the date is saved in the correct format (**dd/MM/yyyy**).
- **Press 5 to “[optional] List all tax inspectors (from MongoDB)”**
  - Just read the **MongoDB database** and print every **TAX INSPECTOR** 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 tax inspectors and marks (to SQLite)	The ArrayList is populated properly. Creates and manages the SQLite database structure properly. Writes the data properly to SQLite. The date format is correct.	3
List tax inspectors (from SQLite)	Reads the data properly from SQLite. Prints the data in a proper way.	2
Delete tax inspectors (from SQLite)	Deletes the data properly from SQLite.	2
[optional] Dump data from SQLite to MongoDB. The date format is correct		1
[optional] List tax inspectors 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	COMPETENT/NOT COMPETENT
	A proper text is provided (within the code or in a text file) to describe the AT <b>using THREE PARAGRAPHS</b>	COMPETENT/NOT COMPETENT
Oral skills	Uses a vocabulary appropriate for the purpose	COMPETENT/NOT COMPETENT
	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



Licensed under the [Creative Commons Attribution Share Alike License 4.0](https://creativecommons.org/licenses/by-sa/4.0/)