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 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 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 COMPETENT).
- **Due/Deadline**: **11:59pm on Sunday, 5th January, 2025** (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

This is an **individual task**, so 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 "**at3_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 a database of TAX INSPECTORS 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**.

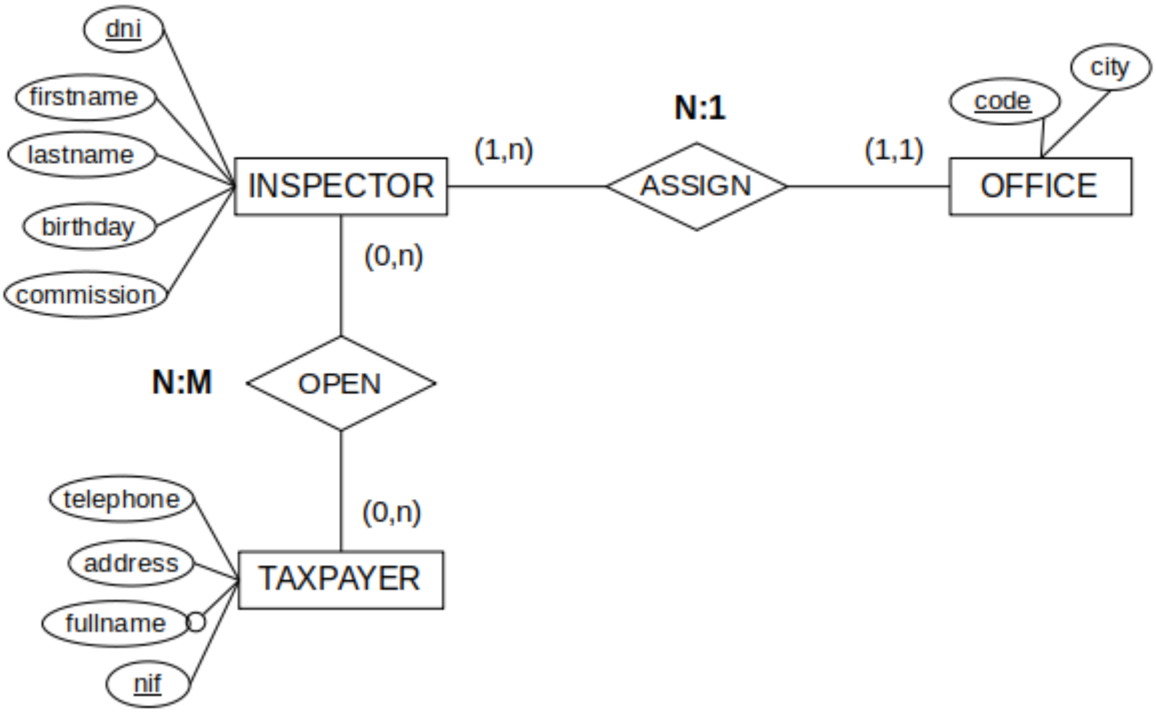
Please, do follow these TECHNICAL SPECIFICATIONS:

- 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:



The **commission** is expressed as a percentage.

```
CREATE DATABASE IF NOT EXISTS ADAU3DBTaxinspectors CHARACTER SET utf8mb4 COLLATE
utf8mb4_es_0900_ai_ci;

CREATE USER 'mavenuser'@'localhost' IDENTIFIED BY 'ada0486'; --
GRANT ALL PRIVILEGES ON ADAU3DBTaxinspectors.* to 'mavenuser'@'localhost';

USE ADAU3DBTaxinspectors;

-- Office
CREATE TABLE Office (
code    VARCHAR(5) PRIMARY KEY,
city    VARCHAR(100)
);

-- Inspector
CREATE TABLE Inspector (
dni         VARCHAR(9) PRIMARY KEY,
firstname   VARCHAR(30),
lastname    VARCHAR(80),
birthday    DATE,
commission  DECIMAL(5,2),
code        VARCHAR(5) NOT NULL,
```

```sql
CONSTRAINT ins_off_fk FOREIGN KEY (code) REFERENCES Office(code) ON UPDATE CASCADE,
CONSTRAINT ins_com_ck CHECK (commission BETWEEN 0 AND 30)
);

-- Taxpayer
CREATE TABLE Taxpayer (
nif         VARCHAR(9) PRIMARY KEY,
fullname    VARCHAR(110) NOT NULL,
address     VARCHAR(150),
telephone   VARCHAR(12)
);

-- Taxfile
CREATE TABLE Taxfile (
inspdni     VARCHAR(9),
taxpaynif   VARCHAR(9),
PRIMARY KEY (inspdni, taxpaynif),
CONSTRAINT taxf_dni_fk FOREIGN KEY (inspdni) REFERENCES Inspector(dni) ON DELETE CASCADE,
CONSTRAINT taxf_nif_fk FOREIGN KEY (taxpaynif) REFERENCES Taxpayer(nif) ON DELETE CASCADE
);
```

---

2) Menu options:

- **Press 0 to “Exit”**
- **Press 1 to “Insert & List offices”**
  - This option will ask for items in loop until zero is entered.
  - For every item we need code (String) and city (String with spaces).
  - 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 Office table and a list of offices will be shown. Before it, you should check if the ArrayList is empty to avoid executing unnecessary code.
  - **Check if the office 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**.
- **Press 2 to “Insert & List taxpayers“**
  - This option will ask for items in loop until zero is entered.
  - For every item we need Nif (String), full name (String with spaces), address (String with spaces) and telephone (String).

  - For every item given, we will store it in an ArrayList. Once zero is entered as a Nif, all items will be inserted into the Taxpayer table and a list of taxpayers will be shown. Before it, you should check if the ArrayList is empty to avoid executing unnecessary code.
  - **Check if the Nif 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 Nif**.

- **Press 3 to “Insert & List tax inspectors“**
  - This option will ask for items in loop until zero is entered.
  - For every item we need Dni (String), first name (String with spaces), lastname (String with spaces), birthday (LocalDate in format **dd/MM/yyyy**) and commission (Float).
    - Also, for every item, we need to be able to associate an office. The Code typed must exist.
    - Also, for every item, we need to be able to associate as many taxpayers as we want, until zero is entered as a Nif (the Nif typed different than zero must exists).
  - For every item given, we will store it in an ArrayList. Once zero is entered as a Dni, all items will be inserted into the Inspector table and a list of tax inspectors will be shown. Before it, you should check if the ArrayList is empty to avoid executing unnecessary code.
  - **Check if the Dni 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 Dni**.
  - **ATTENTION**: For every tax inspector, the **commission percentage** shall be **between 0 and 30**.

- **Press 4 to “List taxpayers under investigation (by tax inspector)”**
  - This option will ask for a Dni (String).
  - Once the Dni has been entered, we will display the list of all taxpayers (nif and fullname) under investigation by that tax inspector.
  - **ATTENTION**: Dni must exist in the table Inspector, you must ask for each value (in loop) until the user enters a valid Dni.

- **Press 5 to “Update office”**
  - This option will ask for a Code (String).
  - Once the Code has been entered, we will ask for the "new Code" to update.

  - Once the "new Code" has been entered, it will be updated in table Office (and associated tables).
  - **ATTENTION**: Code must exist in the table Office, you must ask for each value (in loop) until the user enters a valid Code.

- **Press 6 to “Delete taxpayers”**
  - 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 Nif, all items will be (cascade) deleted in table Taxpayer (and associated tables).
- **Press 7 to “Delete tax inspectors”**
  - 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 Dni, all items will be (cascade) deleted in table Inspector (and associated tables).

Menu example:

```
*****
MENU
*****
========================================================================
  0. Exit
  1. Insert & List offices
  2. Insert & List taxpayers
  3. Insert & List tax inspectors
  4. List taxpayers under investigation (by tax inspector)
  5. Update office
  6. Delete taxpayers
  7. Delete tax inspectors
  8. [optional] Find tax inspectors
  9. [optional] Find taxpayers
========================================================================
    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 8 to “[optional] Find tax inspectors”**
  - This option will ask for a number (Float).
  - With that number, a list of tax inspectors with commission percentage greater than that number will be shown, order by lastname, firstname ascending.
  - **ATTENTION:** commission percentage shall be between 0 and 30.
- **Press 9 to “[optional] Find taxpayers”**
  - This option will ask for a Dni or zero to back to the menu.
  - With that Dni, the full name and the total number of taxpayers under investigation by that tax inspector will be shown.

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
	Update/ delete data	Updates/deletes the data properly from MySQL. Modifies the data properly.	1.5
[optional] Find tax inspectors			1
[optional] Find taxpayers			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