DAM. UNIT 4. ACCESS USING COMPONENTS. ASSESSABLE TASK 4

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

Unit 4. ACCESS USING COMPONENTS

Assessable Task 4

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 4** to meet a provided specification.

2.TIMELINE AND EXPECTATIONS

- **Percentages within the TERM**: 50% of TERM total (AT3 would make the other 50%)
- **Percentages within the TASK**: 100% ADA skills (English skills must be COMPETENT).
- **Due/Deadline**: **11:59pm on Sunday, 16th February, 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 "**at4_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.

B1. Mandatory features

Activity (ASSESSABLE)

Create a program in Java to manage basic operations of a MySQL database called ADAU4DBTaxinspectors with the structure provided via JDBC, to get an Output like the one also provided, importing a jar library containing several JavaBeans (amongst other auxiliary classes). See appendices for detailed information.

Please, do follow these TECHNICAL SPECIFICATIONS:

- RDBMS: **MySQL**
- Language: **Java**
- Framework: **None**
- ORM: **None**
- DAO: **JavaBeans (jar file)**

The application (method main) will go this way:

1. Read a random tax inspector, an office and a taxpayer from the database.
2. Simulate the tax inspector opens an investigation to the taxpayer (Taxfile).
3. Simulate the taxpayer gets an appointment in that office with the tax inspector.
4. Simulate the taxpayer meets the appointment with the tax inspector.
5. Simulate the taxpayer receives a tax inspector's positive ruling.

You should create a JavaBean class called TaxpayerBean, acting as a SOURCE, which will be interacting with 3 more JavaBean classes called MessageBean, TaxfileBean and FineBean, acting as a LISTENERS. Additionally, you will be using 3 more AUXILIARY classes called Inspector, Office and DBBean.

a) When the taxpayer receives a tax inspector's demand, a property called NextDeadLineDate will change to the specified date and the listeners will react to this

change ...

- **MessageBean** will insert a message into messages table and print it out (see output screenshot).
- **TaxfileBean** will insert a tax file into Taxfile table with a deadline of today plus 15 days, and update the taxpayer table indicating that is under investigation (see output screenshot).

b) When the taxpayer gets an appointment in an office, a property called NextAppDate will change to the specified date and the listeners will react to this change ...

- **MessageBean** will insert a message into messages table and print it out (see output screenshot).
- **TaxfileBean** will update an appointment in that office in the current date into Taxfile table (see output screenshot).
- **FineBean** will check whether it is a meeting after the deadline or not. If the meeting is after, it will update the fine with the default amount (10000) at Taxfile table and update the taxpayer table indicating that has been fined (see output screenshot).

c) When the taxpayer meets an appointment with the tax inspector (and submits all required documentation), a property called NextAppDate will change to NULL and the listeners will react to this change ...

- **MessageBean** will insert a message into messages table and print it out (see output screenshot).
- **TaxfileBean** will update the appointment date into Taxfile table (see output screenshot).

d) When the taxpayer receives a tax inspector's positive ruling, a property called NextDeadLineDate will change to NULL and the listeners will react to this change ...

- **MessageBean** will insert a message into messages table and print it out (see output screenshot).
- **TaxfileBean** will set the fine to 0 at Taxfile table and update the taxpayer table indicating that is free and no longer has an investigation (see output screenshot).

B2. Optional features

Activity (ASSESSABLE)

In order to get the maximum score, you should add the additional code to prevent:

- Any tax inspector from investigating a taxpayer who is already under investigation (no need to be so despicable).
- Any taxpayer from scheduling an appointment with a tax inspector if an appointment has already been made.

C. Learning Rubric

C1. 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)
JAR File	Jar file is imported.	0.5
Beans	Use and interaction amongst the JavaBeans.	3.5
Classes	Use of auxiliary classes.	1
Database	Interaction with the database.	3
[optional] Prevent any tax inspector from investigating a taxpayer who is already under investigation (no need to be so despicable). Prevent any taxpayer from scheduling an appointment with a tax inspector if an appointment has already been made.		2

C2. 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

D. Appendices

D1. Database and data

```
CREATE DATABASE IF NOT EXISTS ADAU4DBTaxinspectors CHARACTER SET utf8mb4 COLLATE
utf8mb4_es_0900_ai_ci;

CREATE USER 'mavenuser'@'localhost' IDENTIFIED BY 'ada0486'; --
GRANT ALL PRIVILEGES ON ADAU4DBTaxinspectors.* to 'mavenuser'@'localhost';

USE ADAU4DBTaxinspectors;

-- 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)
);

-- Taxpayer
CREATE TABLE Taxpayer (
nif         VARCHAR(9) PRIMARY KEY,
fullname    VARCHAR(110) NOT NULL,
status      VARCHAR(15) DEFAULT 'Free',
CONSTRAINT taxp_sta_ck CHECK (status IN ('Free','Investigated','Fined'))
);

-- Taxfile
CREATE TABLE Taxfile (
numfile     INTEGER AUTO_INCREMENT,
inspdni     VARCHAR(9) NOT NULL,
taxpaynif   VARCHAR(9) NOT NULL,
office      VARCHAR(5),
deadline    TIMESTAMP,
appointment TIMESTAMP,
fine        FLOAT DEFAULT 0,
```

```
PRIMARY KEY (numfile),
CONSTRAINT txf_dni_fk FOREIGN KEY (inspdni) REFERENCES Inspector(dni),
CONSTRAINT txf_nif_fk FOREIGN KEY (taxpaynif) REFERENCES Taxpayer(nif),
CONSTRAINT txf_off_fk FOREIGN KEY (office) REFERENCES Office(code),
CONSTRAINT txf_fin_ck CHECK (fine >= 0)
);

-- Message
CREATE TABLE Message (
messageID   INTEGER AUTO_INCREMENT,
taxpaynif   VARCHAR(9),
description VARCHAR(500),
PRIMARY KEY (messageID),
CONSTRAINT mes_tpa_fk FOREIGN KEY (taxpaynif) REFERENCES Taxpayer(nif)
);

-- Data
INSERT INTO Office (code, city) VALUES ('01', 'Corrupoly Central');
INSERT INTO Office (code, city) VALUES ('02', 'Bermuda Islands');
INSERT INTO Inspector (dni, firstname, lastname) VALUES ('11111111A', 'Dirty', 'Harry');
INSERT INTO Inspector (dni, firstname, lastname) VALUES ('22222222B', 'Paco', 'Todoparamí');
INSERT INTO Taxpayer (nif, fullname) VALUES ('33333333C', 'John Broke');
INSERT INTO Taxpayer (nif, fullname) VALUES ('44444444D', 'Notengo Niunduro');
INSERT INTO Taxpayer (nif, fullname) VALUES ('55555555E', 'Toi Sinblanca');
INSERT INTO Taxpayer (nif, fullname) VALUES ('66666666F', 'George Hill');
```

D2. Minimum classes and recommended properties

TaxpayerBean class suggested properties:

- private String stNif;
- private String stFullname;
- private String stStatus;
- private Inspector objInspector;
- private LocalDateTime ldtNextDeadLineDate;
- private LocalDateTime ldtNextAppDate;

TaxfileBean class suggested properties:

- private TaxpayerBean objTaxpayerBean;
- private float fFine;
- private Office objOffice;
- private LocalDateTime ldtDeadLine;
- private LocalDateTime ldtAppointment;

FineBean class suggested properties:

- private TaxfileBean objTaxfileBean;

MesssageBean class suggested properties:

- private TaxpayerBean objTaxpayerBean;

DBBean class suggested properties:

- private Connection cnDB = null;
- private PreparedStatement pstaSQL = null;

D3. Suggested Output

This is an example, but there are many possibilities.

```
Reading a random tax inspector...
=> Inspector: 'Dirty Harry' ...
Reading a random office...
=> Office: 'Corrupoly Central' ...
Reading a random taxpayer...
=> Taxpayer: 'Notengo Niunduro' ...
-------------------------
HI, MESSAGEBEAN SPEAKING!
                    ********
New taxfile by 'Dirty Harry' and registered to taxpayer 'Notengo Niunduro'
Deadline on date 2025-02-01 and time 11:00
                    ********
Inserted message.
-------------------------
HI, TAXFILEBEAN SPEAKING!
Inserted Taxfile.
Updated taxpayer. Status set to Investigated.
-------------------------
HI, MESSAGEBEAN SPEAKING!
                    ********
New appointment by taxpayer 'Notengo Niunduro'
at office 'Corrupoly Central' on date 2025-02-11 and time 10:00
                    ********
Inserted message.
-------------------------
HI, TAXFILEBEAN SPEAKING!
Inserted Appointment.
Updated taxfile. Has an appointment.
-------------------------
HI, MESSAGEBEAN SPEAKING!
                    ********
The taxpayer 'Notengo Niunduro' meets the appointment
on date 2025-02-11 and time 10:00
                    ********
Inserted message.
-------------------------
HI, TAXFILEBEAN SPEAKING!
```

```
Appointment made.
Updated taxfile. No longer has an appointment.
-------------------------
HI, MESSAGEBEAN SPEAKING!
                    ********
Taxpayer 'Notengo Niunduro'. Positive file ruling by tax inspector 'Dirty Harry'
                    ********
Inserted message.
-------------------------
HI, TAXFILEBEAN SPEAKING!
                    ********
Taxpayer 'Notengo Niunduro' is free.
Updated taxfile. Fine set to 0.
Updated taxpayer. No longer has an investigation.
                    ********
```