

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 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 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 PASSED).
- **Due/Deadline:** **11:59pm on Sunday, 18th February, 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 basic operations of a MySQL database called DBChessGames 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 chess player and a tournament from the database (pl+t).
2. Simulate the player is playing a chess game at that tournament (Game).
3. Simulate the same (pl+t) requests a postponement of a chess game (Deferral).

You should create a JavaBean class called PlayerBean, acting as a SOURCE, which will be interacting with 3 more JavaBean classes called MessageBean, GameBean and DeferralBean, acting as a LISTENERS.

Additionally, you will be using 2 more AUXILIARY classes called Tournament and DBBean.

a) When the chess player plays a match in a tournament, a property called nextMatchDate will change from NULL 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).

- **GameBean** will insert a match into Game table and update the players table indicating that has a match (see output screenshot).

b) When the chess player asks for a deferral in a tournament, a property called nextDeferralDate will change from NULL 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).
- **DeferralBean** will insert a postponement into Deferral table and update the players table indicating that has a deferral (see output screenshot).

c) When the chess player finishes a match, a property called nextMatchDate will change from the specified date 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).
- **GameBean** will update the match at Game table and update the players table indicating that no longer has a match (see output screenshot).

d) When the chess player gets the verdict of a postponement, a property called nextDeferralDate will change from the specified date 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).
- **GameBean** will update the postponement at Deferral table and update the players table indicating that no longer has a deferral (see output screenshot).

B.2. Optional features

Activity (ASSESSABLE)

In order to get the maximum score, you should add the additional code to prevent any player from requesting a deferral if there's no GAME in status PENDING related to that player.

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)
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 requesting a deferral if there's no game in status pending		2

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

D. Appendices

D.1. Database and data

```

DROP DATABASE IF EXISTS DBChessGames;
CREATE DATABASE DBChessGames CHARACTER SET utf8 COLLATE utf8_spanish_ci;

-- CREATE USER 'mavenuser'@'localhost' IDENTIFIED WITH mysql_native_password E
-- GRANT ALL PRIVILEGES ON DBChessGames.* to 'mavenuser'@'localhost';

USE DBChessGames;

DROP TABLE IF EXISTS Player;
DROP TABLE IF EXISTS Tournament;
DROP TABLE IF EXISTS Game;
DROP TABLE IF EXISTS Deferral;
DROP TABLE IF EXISTS Message;

-- Player
CREATE TABLE Player (
  playerID      INTEGER,
  fullname      VARCHAR(100) NOT NULL,
  has_match     BOOLEAN DEFAULT FALSE,
  has_deferral  BOOLEAN DEFAULT FALSE,
  CONSTRAINT pla_id_pk PRIMARY KEY (playerID)
);

-- Tournament
CREATE TABLE Tournament (
  code          VARCHAR(10),
  name          VARCHAR(100),
  CONSTRAINT tou_cod_pk PRIMARY KEY (code)
);

```

```

-- Game
CREATE TABLE Game (
gameID      INTEGER AUTO_INCREMENT,
code        VARCHAR(10),
playerID    INTEGER,
matchdate   DATETIME,
result      VARCHAR(10) DEFAULT 'PENDING',
CONSTRAINT gam_gam_pk PRIMARY KEY (gameID),
CONSTRAINT gam_cod_fk FOREIGN KEY (code) REFERENCES Tournament(code),
CONSTRAINT gam_pla_fk FOREIGN KEY (playerID) REFERENCES Player(playerID),
CONSTRAINT gam_res_ck CHECK (result IN ('PENDING', 'WON', 'LOST', 'DRAWS'))
);

-- Deferral
CREATE TABLE Deferral (
deferralID  INTEGER AUTO_INCREMENT,
code        VARCHAR(10),
playerID    INTEGER,
defdate     DATETIME,
result      VARCHAR(10) DEFAULT 'REQUESTED',
CONSTRAINT def_def_pk PRIMARY KEY (deferralID),
CONSTRAINT def_cod_fk FOREIGN KEY (code) REFERENCES Tournament(code),
CONSTRAINT def_pla_fk FOREIGN KEY (playerID) REFERENCES Player(playerID),
CONSTRAINT def_res_ck CHECK (result IN ('REQUESTED', 'GRANTED', 'REJECTED'))
);

CREATE TABLE Message (
messageID   INTEGER AUTO_INCREMENT,
playerID    INTEGER,
description VARCHAR(500),
CONSTRAINT mes_mmc_pk PRIMARY KEY (messageID),
CONSTRAINT mes_pla_fk FOREIGN KEY (playerID) REFERENCES Player(playerID)
);

-- Data
INSERT INTO Player (playerID, fullname) VALUES (1, 'Anatoly Karpov');
INSERT INTO Player (playerID, fullname) VALUES (2, 'Garry Kasparov');

```

```
INSERT INTO Player (playerID, fullname) VALUES (3, 'Pepe Tableros');  
INSERT INTO Player (playerID, fullname) VALUES (4, 'John Lost');  
INSERT INTO Tournament (code, name) VALUES ('T01', 'Torneo local de Quart de F');  
INSERT INTO Tournament (code, name) VALUES ('T02', 'Torneo Internacional de Aj');  
INSERT INTO Tournament (code, name) VALUES ('T03', 'Torneo Jaque con tomate');  
INSERT INTO Tournament (code, name) VALUES ('T04', 'Torneo Maestro Yoda');
```

D.2. Minimum classes and recommended properties

PlayerBean class suggested properties:

- private String stFullname;
 - private Tournament objTournament;
 - private boolean bHasAMatch;
 - private boolean bHasADeferral;
 - private LocalDateTime ldtNextMatchDate;
 - private LocalDateTime ldtNextDeferralDate;
-

GameBean class suggested properties:

- private PlayerBean objPlayerBean;
-

DeferralBean class suggested properties:

- private PlayerBean objPlayerBean;
-

MesssageBean class suggested properties:

- private PlayerBean objPlayerBean;
-

Tournament class suggested properties:

- private PlayerBean objPlayerBean;
-

DBBean class suggested properties:

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

D.3. Suggested Output

```

Reading a random player...
=> Player: 'Pepe Tableros' ...
Reading a random tournament...
=> Tournament: 'Torneo Maestro Yoda' ...

-----
HI, MESSAGEBEAN SPEAKING!

*****

Game match set as PENDING and registered to player 'Pepe Tableros'
at tournament 'Torneo Maestro Yoda' on date 2024-03-08 and time 18:00

*****

Inserted message.

-----
HI, GAMEBEAN SPEAKING!
Inserted Game.
Updated player. Has a match appointment.

-----
HI, MESSAGEBEAN SPEAKING!

*****

Deferral set as REQUESTED and registered to player 'Pepe Tableros'
at tournament 'Torneo Maestro Yoda' on date 2024-03-08 and time 18:00

*****

Inserted message.

-----
HI, DEFERRALBEAN SPEAKING!
Inserted Deferral.
Updated player. Has a deferral requested.

-----
HI, MESSAGEBEAN SPEAKING!

*****

Chess game completed set as DRAWS

```

Inserted message.

HI, GAMEBEAN SPEAKING!

Updated Game (DRAWS).

Updated player. No longer has a match appointment.

HI, MESSAGEBEAN SPEAKING!

Deferral request set as REJECTED

Inserted message.

HI, DEFERRALBEAN SPEAKING!

Updated Deferral (REJECTED).

Updated player. No longer has a deferral request.



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