DAM. UNIT 2. ACCESS TO DATABASES. PART 1. NON ASSESSABLE EXERCISES

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

Unit 2. ACCESS TO DATABASES

Part 1. Working with Relational Databases. Non assessable exercises

Abelardo Martínez

Based and modified from Sergio Badal (www.sergiobadal.com)

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.

1. Console mode. Managing a relational database via JDBC in SQLite

Activity (non assessable)

Create a program in Java to manage PIZZAS in an Italian restaurant by printing and using a specific menu. After each option, the user should see the same menu until option zero is pressed. Feel free to share your doubts at the UNIT forum. **You can duplicate this project from UNIT 1**.

ATTENTION: Use the proper exceptions when accessing to databases.

Menu options:

- **Press 0 to "Exit"**
- **Press 1 to "Ask for pizzas until user enters zero as ID"**
  - For every pizza we need the ID (Integer), Name (String with spaces) and Price (Double), added to the ArrayList of pizzas.
  - Check if the pizza 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 pizzas will be saved in a **SQLite database using JDBC**. You decide fields and tablename.
  - **ATTENTION**: ID must be integer! For every pizza, you must ask for an integer (in loop) until the user enters an integer or zero to print again the menu.
  - **ATTENTION**: The table has to be dropped and created again everytime you add the ArrayList.
- **Press 2 to "List all the pizzas"**
  - Just read the **SQLite database using JDBC** and print every pizza information.
- **Press 3 to "Remove all pizzas"**

- Just delete the pizzas stored at the **SQLite database using JDBC**.

Menu example:

```
*****
MENU
*****
================================================
  0. Exit
  1. Add pizzas
  2. List all pizzas
  3. Remove all pizzas
================================================
    Select an option:
```

2. Console mode. Managing a relational database via JDBC in MySQL

Activity (non assessable)

Create a program in Java to manage ANIMALS in a veterinary clinic by printing and using a specific menu. After each option, the user should see the same menu until option zero is pressed. Feel free to share your doubts at the UNIT forum. **You can duplicate this project from UNIT 1**.

ATTENTION: Use the proper exceptions when accessing to databases.

Menu options:

- **Press 0 to "Exit"**
- **Press 1 to "Ask for animals (patients) until user enters zero as ID"**
  - For every animal we need the ID (String), Name (String with spaces), Species (String with spaces), Breed (String with spaces), CurrentAge (Integer) and Sterilised (String), saved in an ArrayList of animals. If an animal is sterilised, we type "yes" and "no" otherwise.
  - Check if the animal 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 animals will be saved in a MySQL database using JDBC. You decide fields and tablename.
  - **ATTENTION**: Age must be integer!
  - **ATTENTION**: The table has to be dropped and created again everytime you add the ArrayList.
- **Press 2 to "List all animals"**

    - Just read the **MySQL database using JDBC** and print every animal information.
- **Press 3 to “Remove all animals”**
    - Just delete the animals stored at the **MySQL database using JDBC**.

Menu example:

```
*****
MENU
*****
================================================
  0. Exit
  1. Add animals
  2. List all animals
  3. Remove all animals
================================================
    Select an option:
```