Animals: JDBC

Create a copy of the previous Animals exercise. Modify the project to use a database for persistent storage.

Database

- In DBeaver, ensure there is a connection to your MariaDB database server (should be called “localhost”)
- Create a database called “animals_db”
- Use the provided SQL script to generate the “animals” and “categories” tables

Animal repository

Write a CRUD repository class to handle communication with the database. For now, all you need is to add query methods (no adding, removing, or updating required).

Public query methods

- Category getCategory(int id)
- ArrayList<Category> getCategories()
- Animal getAnimal(int id)
- ArrayList<Animal> getAnimals()
- ArrayList<Animal> getAnimalsByCategory(int categoryId)

Test

- Ensure that all of your public repository methods work correctly
- To do so, it might be helpful to create a simple console application (not web application) for testing

Animal service

Modify the implementation of the AnimalService class to use a repository to access the database. Previously it was using static hash maps of categories and animals to simulate a database.

- Delete the hash maps
- Replace them with an AnimalRepository instance variable (not a static variable)
- Initialize the repository in the default constructor
- Rewrite the definition of all the public methods in the class (no need to add or remove any public methods) to use the repository instead of the hash maps to access the animals and categories
- Keep the code that sorts the lists of animals and categories
- Bonus: Modify your repository SQL list queries to order the records by name, so that the database will sort them, instead of the service

Test

- Test the web application to make sure that everything works correctly