Java Technologies - Lab 5

[valid 2022-2023]

Java Persistence API (JPA)
Continue the application created in the previous lab.

Compulsory (1p)

- Define the *persistence unit* using a *data source* configured as a JDBC Resource.
- Define at least one entity class and perform a simple test in order to verify the communication with the database.

Homework (2p)

Rewrite the persistence layer of the application created for the previous laboratory using a technology that implements the **JPA specifications**.

- Create the *EntityManager* objects using dependency injection.
- Define the mappings using JPA-only annotations.
- Create named queries using **JPA-QL**.
- Implement the repository classes.
- Create a complete test unit for the CRUD operations of at least one entity class.

Bonus (2p)

Each team has *players*. Each player has a name, age, position, etc. Populate the database with 25 players for each team (you may use Java Faker).
Add a *"player search page"*. This page will allow specifying various filters: partial name, age interval, position, team, etc.

- Each filter will have a checkox - if it is checked then the filter will be taken into consideration.
- The query must be implemented using **JPA Criteria API**.

For an additional bonus, you may consider that each team has also a *coach*. Players and coaches are all *persons*.
Use **inheritance** mapping in order to define this new model.

Notes

- It is **required** to use a JPA implementation and not just *any* third-party ORM library.

Bibliography:

- Slides
- Java EE Tutorial: Persistence
- Eclipse Link Documentation Center
- Hibernate EntityManager
- Dynamic, typesafe queries in JPA
- Criteria Queries
- Java Persistence Performance