Team B

Description – Lord of the ring

Bilbo Baggins is going on a long journey to help the dwarves return to the Lonely Mountain. Create an app that can capture a hero, such a hero has a name (unique), belief (bad-good), power, will, magic (all as **int**).

Heroes form squads. A squad can have many heroes, and at the same time all heroes in the squad must have the same belief. The clash of evil and good squads means a fight. In combat, strength, will or magic is randomly selected, it is added up for all the heroes of the squad and the squad with the higher value wins.

Tasks

- Analyze the task and create a class diagram.

- According to the design, implement the ORM in Java (choose the technology yourself: hibernate, etc.).

- Use lombok for easy getters, setters , constructors etc. :
  https://www.baeldung.com/intro-to-project-lombok

- In your project use M:N cardinality:

  https://www.baeldung.com/jpa-many-to-many

- Find out what the fetch and mappedBy parameters mean

- Implement DAO layer and CRUD.

- Describe the following three services using a sequence diagram:

- Service for assembling a squad

- Clash of two or more mobs

- Combat

- Implement the services layer, according to sequence diagrams.

- Test service layer of application:

  - unit tests

- Create an import.sql file in the resources folder. Create test data here.

- Add actuator:

https://www.baeldung.com/spring-boot-actuators

- Add Springfox for generating swagger specification:

https://www.baeldung.com/swagger-2-documentation-for-spring-rest-api

- Add javadoc

- Add Micrometer for metrics

https://www.tutorialworks.com/spring-boot-prometheus-micrometer/

-For DTO layer use mapstruct maping between BO and DTO:

https://www.baeldung.com/mapstruct

Convention

Unless otherwise addressed or it does not appear from the context that,

  - methods mean instance methods,

  - you can define your helper attributes and methods and

  - all non-abstract classes you implement should have a parameterless constructor.

  - Solve problematic conditions cross all application. For example, using try catch. Useloggers (error, warning, debug, info, trace). You always need to know what is happening with the application (error message).

  - add in repository readme file with list of thinks that are done and that are not done for better tracking