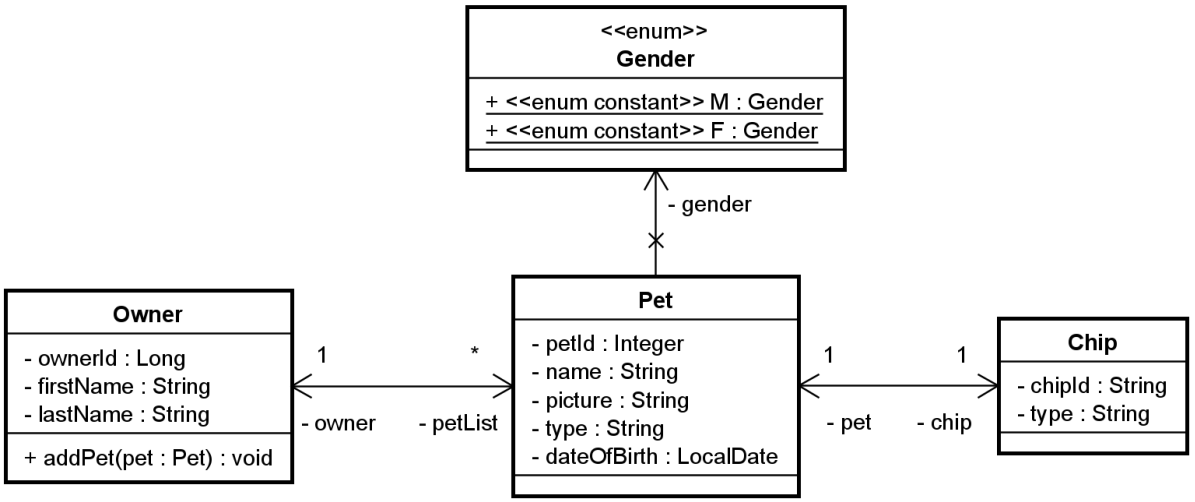
JPA PetShop

SPRING BOOT, JSON, LOMBOK, JACKSON, JPA

Create the database **`petshop`** according to the given class-diagram with a JPA mapping.



Implement a Spring-Boot-Application that reads data from the file **`petShop.json`**, creates the POJO-Objects and persists them to the database. After a successful import, the number of created datasets is printed to the console.

Structure of the application:

- **`pojos`**-package contains pojo-classes according to the class diagram
- **`database`**-package contains the class for the database initialization
- **`repository`**-package contains the repository-interfaces
- **`web`**-package: Rest-controller-classes to support required endpoints

Mapping constraints for the Pojo-classes:

`pojos.Pet.java`

- JPA-mapping according to the class diagram
- The primary-key value is created by the application.
- The associations to classes **`Chip`** and **`Owner`** are bidirectional.
- Use the Enum **`Gender`** for the **`gender`** variable. Save the values as String in the database.

`pojos.Chip.java`

- JPA-mapping according to the class diagram
- The primary-key value is created in the constructor.
- The association to class **`Pet`** is bidirectional.

`pojos.Gender.java`

- Enumeration with the values **`M`** and **`F`**.

---

pojos.Owner.java

- JPA-mapping according to the class diagram
- The primary-key value is created by the database.
- The association to class **Pet** is bidirectional.

Required Repository-interfaces:

- **repository.PetRepository.java**
- **repository.ChipRepository.java**
- **repository.OwnerRepository.java**

---

database.InitDatabase.java

- **importPetsData()**: Read the file **petshop.json**. Distribute the values to the different pojo-objects and persist them to the database **petshopdb**. Primary key values for classes **Pet** and **Chip** must be created by the application. Make sure that all bidirectional associations are set up!
- Print number of imported datasets to the console:
  Pets imported: 1000
  Chips imported: 1000
  Owner imported: 493

Implement the following Endpoints in the **XxxController** classes:

- **GET localhost:8080/petshop/pets/all** returns a list of pets and supports the following query-parameters:
  - **pageNo**: page-number, defaults to 0. page number
  - **pageSize**: page-size, defaults to 10.
  - **petType**: filter on petType. If not present, no filter is applied.
- **GET localhost:8080/petshop/pets/types** returns all different pet-types in alphabetical order
- **GET localhost:8080/petshop/chips/all** returns a list of all different chip-types in alphabetical order.