# Databases Frameworks – Hibernate and Spring Data

# Retake Exam

Exam problems for the [Databases Advanced – Hibernate and Spring Data @ SoftUni](https://softuni.bg/courses/databases-advanced-hibernate).

Download the provided **skeleton** and use it in your solutions. You are **not allowed** to change the structure or the names of the provided **interfaces**, **classes** and **methods**. **It is very important to use the specified dependency versions in the provided pom.xml file.** Any usage of other dependencies is at your own risk.

Submit your project solutions in the SoftUni judge system, as a zip archive file – only the **src** folder and the **pom** file, **not including the test folder**.

Your task is to create a database console application using **Hibernate and Spring Data** using the **Code First** approach. Design the **domain models** and **methods** for manipulating the data, as described below.

# Pet Clinic

Doctor John Dolittle’s clinic is starting to drastically grow. He now has more clients, he has employeed doctors and is organizing his clinic’s records on paper is becoming hard. He has contacted you to help him with the changes he wants to make and wants you to design a database to keep track of the processes in the clinic.

## Project Skeleton Overview

You are given a project skeleton, which you are not allowed to change, nor add exceptions to.

**Controllers** – the main classes responsible for importing exporting data. Use their importDataFromJSON, importDataFromXML and export methods

**DTO’s and Entities** – your database models and import/export classes

**Parser –** the interface which has to be implemented by your JSON and XML parsers

**Repositories** – the repositories assigned to a single model in the database

**Service interfaces –** interfaces which have to be implemented by your own service implementations. Place the main validation and insertion logic there

## Problem 1. Model Definition (50 pts)

In his clinic Dr. Dolittle has employed **different kinds of** **doctors**, who serve patients – **animals**. Different **procedures** are performed on an animal, which have a pre-defined cost.

Design the following database models:

### Animal (10 pts)

* id – **integer** number, primary identification field
* name – a string with minimum size of **3 characters** and **no longer than 20, required**
* type – type of the animal, e.g “cat”, “dog”, “parrot” еtc; a string consisting of **no more than 20 characters** and **longer or equal to 3, required**
* age – **integer** value, **cannot be negative or 0**
* passport – the passport of the animal

### Passport (10 pts)

* serialNumber – a string consisting of exactly **10 characters** and **ending with 3 digits, primary identification field**
* animal – the animal to which the passport is registered
* ownerPhoneNumber – the phone number of the animal’s owner, **required**, make sure it matches the following requirements:
  + either starts with **+359** and **9 digits following**
  + consists of exacly **10** digits, starting with **0**
* ownerName – the name of the animal’s owner; minimum size of **3** characters and maximum **30**
* registrationDate – the date on which the passport was registered

### Vet (10 pts)

* id – **integer** number, primary identification field
* name – a string with minimum size of **3** characters and no longer than **40**
* profession – a string with minimum size of **3** characters and no longer than **50**
* age – integer number, minimum value of **22** years and maximum **65**
* phoneNumber – required, make sure it matches the following requirements:
  + either starts with **+359** and **9 digits following**
  + consists of exacly **10** digits, starting with **0**

### Procedure (10 pts)

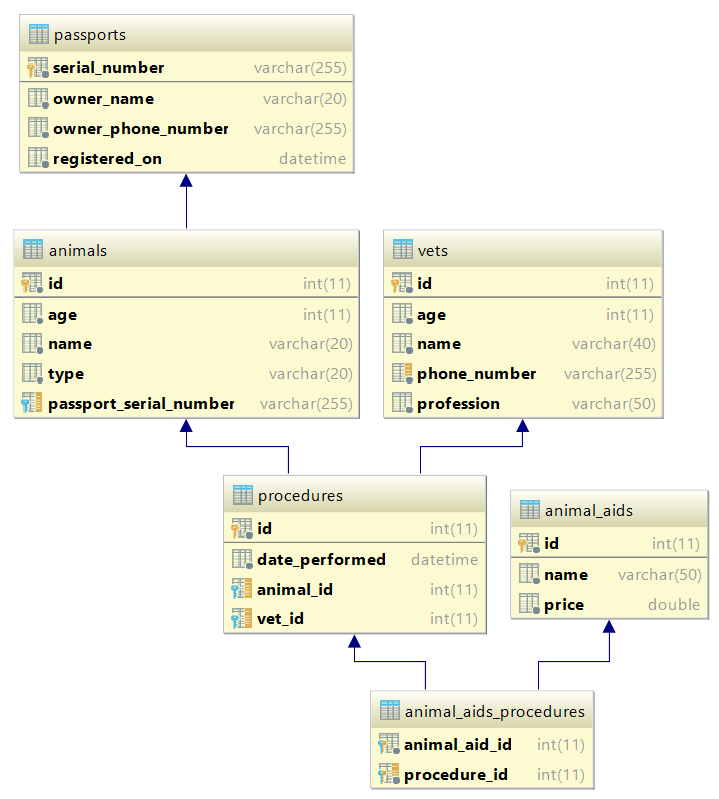
* id – **integer** value, primary indentification field
* services – **collection of services** performed to the animal
* animal – the **animal** to which the procedure is performed
* cost – the **cost of the procedure**, **calculated by sumating the price of the different services performed;** does not need to be inserted in the database
* vet – the clinic’s employed doctor servicing the patient
* date – the date on which the given procedure is performed

### AnimalAid (10 pts)

* id – **integer** number, primary identification field
* name – a string containing at least 3 characters with
* price – decimal value, cannot be a negative number or 0

#### Diagram

You are given the following diagram to help you:



## Problem 2. Data Import (25 pts)

For the functionality of the application, you need to create several methods that manipulate the database. The **project skeleton** already provides you with these methods, inside the Parser inteface. Implement those methods in the JSONParser and XMLParser objects accordingly. Use the provided **Data Transfer Objects** and implement them as needed.

Use the provided **JSON** and **XML** files to populate the database with data. Import all the information from those files into the database.

**You are not allowed to modify the provided JSON and XML files.**

Write parsing logic in your parser classes to import the provided json and xml files..

### JSON Import

## Import Animal Aids

Start by importing the least dependent entity – the **different types of animal aid** that the clinic provides.

##### Constraints

* If any validation errors occur (such as if a service **name** **is too long/short**) do not import the service
* If a service **exists and has the same price ignore it**
* If a service **exists and has a different price same price, update it's price in the database**

**If a record does not meet the requirements from the first section, print an error message:**

|  |  |
| --- | --- |
| **Success message** | **Error message** |
| Record {animal aid name} successfully imported. | Error: Invalid data. |

##### Example

|  |
| --- |
| **animal-aid.json** |
| [  {  "name": "Internal Deworming",  "price": 8.00  },  {  "name": "Fecal Test",  "price": 7.50  },  {  "name": "H3N8",  "price": 30.00  },  {  "name": "Nasal Bordetella",  "price": 5.60  },  {  "name": "External Deworming",  "price": -35.00  },  {  "name": "Bordetella",  "price": 7.50  },  … |
| **Output** |
| Record Internal Deworming successfully imported.  Record Fecal Test successfully imported.  Record H3N8 successfully imported.  Record Nasal Bordetella successfully imported.  Error: Invalid data.  … |

## Import Animals

Proceed by importing the next entity – the **animals**.

##### Constraints

* If any validation errors occur (such as if a service **name** **is too long/short**) do not import the service
* If a service **exists and has the same price ignore it**
* If a service **exists and has a different price same price, update it's price in the database**

**If a record does not meet the requirements from the first section, print an error message:**

|  |  |
| --- | --- |
| **Success message** | **Error message** |
| Record {animal name} Passport №: {passport serial number} successfully imported. | Error: Invalid data. |

##### Example

|  |
| --- |
| **animal-aid.json** |
| [  {  "name":"Bella",  "type":"cat",  "age": 2,  "passport": {  "serialNumber": "etyhGgH678",  "ownerName": "Sheldon Cooper",  "ownerPhoneNumber": "0897556446",  "registrationDate": "12-03-2014"  }  },  {  "name":"Charlie",  "type":"cat",  "age": 3,  "passport": {  "serialNumber": "anothev650",  "ownerName": "Magda Bjork",  "ownerPhoneNumber": "+35989776512",  "registrationDate": "15-04-2015"  }  },  {  "name":"Chester",  "type":"dog",  "age": 11,  "passport": {  "serialNumber": "adoggoo451",  "ownerName": "Adriana Lima",  "ownerPhoneNumber": "35989776512",  "registrationDate": "31-12-2016"  }  },  {  "name":"Lucy",  "type":"cat",  "age": 6,  "passport": {  "serialNumber": "acattee321",  "ownerName": "Ivan Ivanov",  "ownerPhoneNumber": "0887446123",  "registrationDate": "10-06-2015"  }  }  … |
| **Output** |
| Record Bella Passport №: etyhGgH678 successfully imported.  Error: Invalid data.  Error: Invalid data.  Record Lucy Passport №: acattee321 successfully imported.  … |

### XML Import

## Import Vets

The next key figure in our app are the vets who take care of the patients. The info about them is given in the **vets.xml** file

**If a record does not meet the requirements from the first section, print an error message:**

|  |  |
| --- | --- |
| **Success message** | **Error message** |
| Record {vet name} successfully imported. | Error: Invalid data. |

##### Constraints

* Validate each row of information according to the first section. If the validation fails, **do not import the vet**

**Example:**

|  |
| --- |
| **vets.xml** |
| <?xml version="1.0" encoding="UTF-8"?>  <vets>  <vet>  <name>Michael Jordan</name>  <profession>Emergency and Critical Care</profession>  <age>45</age>  <phone-number>0897665544</phone-number>  </vet>  <vet>  <name>Melanie Bennington</name>  <profession>Surgery</profession>  <age>21</age>  <phone-number>+359284566778</phone-number>  </vet>  <vet>  <name>Edmond Halley</name>  <profession>Veterinary Nursing</profession>  <age>24</age>  <phone-number>+359284566778</phone-number>  </vet>  <vet>  <name>Niels Bohr</name>  <profession>Internal Medicine</profession>  <age>32</age>  <phone-number>0879557712</phone-number>  </vet>  <vet>  <name>Werner Heisenberg</name>  <profession>Pediatrics, Genetics and Reproduction</profession>  <age>55</age>  <phone-number>0879535712</phone-number>  </vet>  … |
| **Output** |
| Record Michael Jordan successfully imported.  Error: Invalid data.  Record Edmond Halley successfully imported.  Record Niels Bohr successfully imported.  Record Werner Heisenberg successfully imported.  … |

## Import Procedures

Now it's time to import the records of the procedures done on the animals. Parse the information from the **procedures.xml** file.

**If a record does not meet the requirements and following constraits, print an error message:**

|  |  |
| --- | --- |
| **Success message** | **Error message** |
| Record successfully imported. | Error: Invalid data. |

##### Constraints

* Do not import a procedure if:
  + If a vet with such name does not exists
  + If an animal with given serial number does not exist
  + If an animal aid with given name does not exist

**Example:**

|  |
| --- |
| **vets.xml** |
| <?xml version="1.0" encoding="UTF-8"?>  <procedures>  <procedure>  <vet>Niels Bohr</vet>  <animal>acattee321</animal>  <animal-aids>  <animal-aid>  <name>Nasal Bordetella</name>  </animal-aid>  <animal-aid>  <name>Internal Deworming</name>  </animal-aid>  <animal-aid>  <name>Fecal Test</name>  </animal-aid>  </animal-aids>  </procedure>  <procedure>  <vet>Jennifer Evans</vet>  <animal>bernied355</animal>  <animal-aids>  <animal-aid>  <name>Lyme Test</name>  </animal-aid>  <animal-aid>  <name>Fecal Test</name>  </animal-aid>  </animal-aids>  </procedure>  <procedure>  <vet>Michael Jordan</vet>  <animal>barkeer355</animal>  <animal-aids>  <animal-aid>  <name>Injectable Bordetella</name>  </animal-aid>  <animal-aid>  <name>Canine Heartworm Test</name>  </animal-aid>  </animal-aids>  **</procedure>**  … |
| **Output** |
| Record successfully imported.  Record successfully imported.  Record successfully imported.  Record successfully imported.  … |

## Problem 2. Data Export (25 pts)

### JSON Export

Your task is to write logic in the provided exportAnimalsByOwnerPhoneNumber(String phoneNumber) method in the AnimalsController. Export all **animals** by their **owner's** **number** sorted by **age** **ascending**, then by **serial** **number** **alphabetically**.

#### Example:

|  |
| --- |
| exportAnimalsByOwnerPhoneNumber("0887446123") |
| [  {  "ownerName": "Ivan Ivanov",  "animalName": "Jessy",  "age": 3,  "serialNumber": "jessiii355",  "registeredOn": "11-Jan-2015"  },  {  "ownerName": "Ivan Ivanov",  "animalName": "Lucy",  "age": 6,  "serialNumber": "acattee321",  "registeredOn": "06-Jan-2015"  }  ] |

### XML Export

Implement the provided exportAllProcedures() method in the ProceduresController. Export all **records with information about the animal, procedure date,**

|  |
| --- |
| exportProcedures() |
| <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  <procedures>  <procedure animal-passport="jghjyuu355">  <owner>+359897765122</owner>  <animal-aids>  <animal-aid name="Fecal Test" price="7.5"/>  </animal-aids>  </procedure>  <procedure animal-passport="jghjyuu355">  <owner>+359897765122</owner>  <animal-aids>  <animal-aid name="Injectable Bordetella" price="8.4"/>  </animal-aids>  </procedure>  <procedure animal-passport="jessiii355">  <owner>0887446123</owner>  <animal-aids>  <animal-aid name="Canine Rabbies Vaccine" price="10.0"/>  </animal-aids>  </procedure>  …  <procedure animal-passport="jessiii355">  <owner>0887446123</owner>  <animal-aids>  <animal-aid name="Lepto 4" price="15.9"/>  <animal-aid name="Lyme Test" price="15.0"/>  <animal-aid name="Bordetella" price="7.55"/>  <animal-aid name="Fecal Test" price="7.5"/>  </animal-aids>  </procedure>  </procedures> |