# Lab: Java Auto Mapping Objects

This document defines the lab assignments for the ["Spring Data" course @ Software University.](https://softuni.bg/courses/spring-data)

## Simple Mapping

Create class **Employee** that has properties **first name**, **last name**, **salary**, **birthday** and **address**. Create **EmployeeDto** class that will keep synthesized information about instances of the Employee class (only **first name**, **last name** and **salary**). Create an **instance of employee object** and **use the ModelMapper** **to map** the **newly created Employee** to **object of type EmployeeDto**.

## Advanced Mapping

Create class **Employee** with properties **first name**, **last name**, **birthday**, **salary**, information about whether the employee **is on holiday**, **address**, **manager** (another employee) and **list of employees** that he/she is in charge of.

Create **2 types** of data transfer objects – **employee** data transfer object and **manager** data transfer object.

* **EmployeeDto** – first name, last name, salary
* **ManagerDto** – first name, last name, list of EmployeeDtos that he/she is in charge of and their count

**Create a list** of several **employees**, then **transform it to list of ManagerDtos** and **print it on the console** in the format provided below:

{ManagerFirstName} {ManagerLastName} | Employees: {EmployeesCount}

- {EmployeeFirstName} {EmployeeLastName} {EmployeeSalary}

### Example

|  |
| --- |
| **Sample output** |
| Steve Jobbsen | Employees: 2  - Stephen Bjorn 4300.00  - Kirilyc Lefi 4400.00  Carl Kormac | Employees: 14  - Jurgen Straus 1000.45  - Moni Kozinac 2030.99  - Kopp Spidok 2000.21  - … |

## Projection

Use Hibernate to create and insert into a database a **list of employees**, for which each employee has **first name**, **last name**, **salary**, **birthday**, **address** and **manager** (another employee). Transform to **EmployeeDto** (first name, last name, salary, manager's last name) those employees who are **born before 1990**. Order them **by salary descending** and **print them on the console**.

### Example

|  |
| --- |
| **Sample output** |
| Steve Jobbsen 6000.20 – Manager: [no manager]  Kirilyc Lefi 4400.00 – Manager: Jobbsen  Stephen Bjorn 4300.00 – Manager: Jobbsen |