

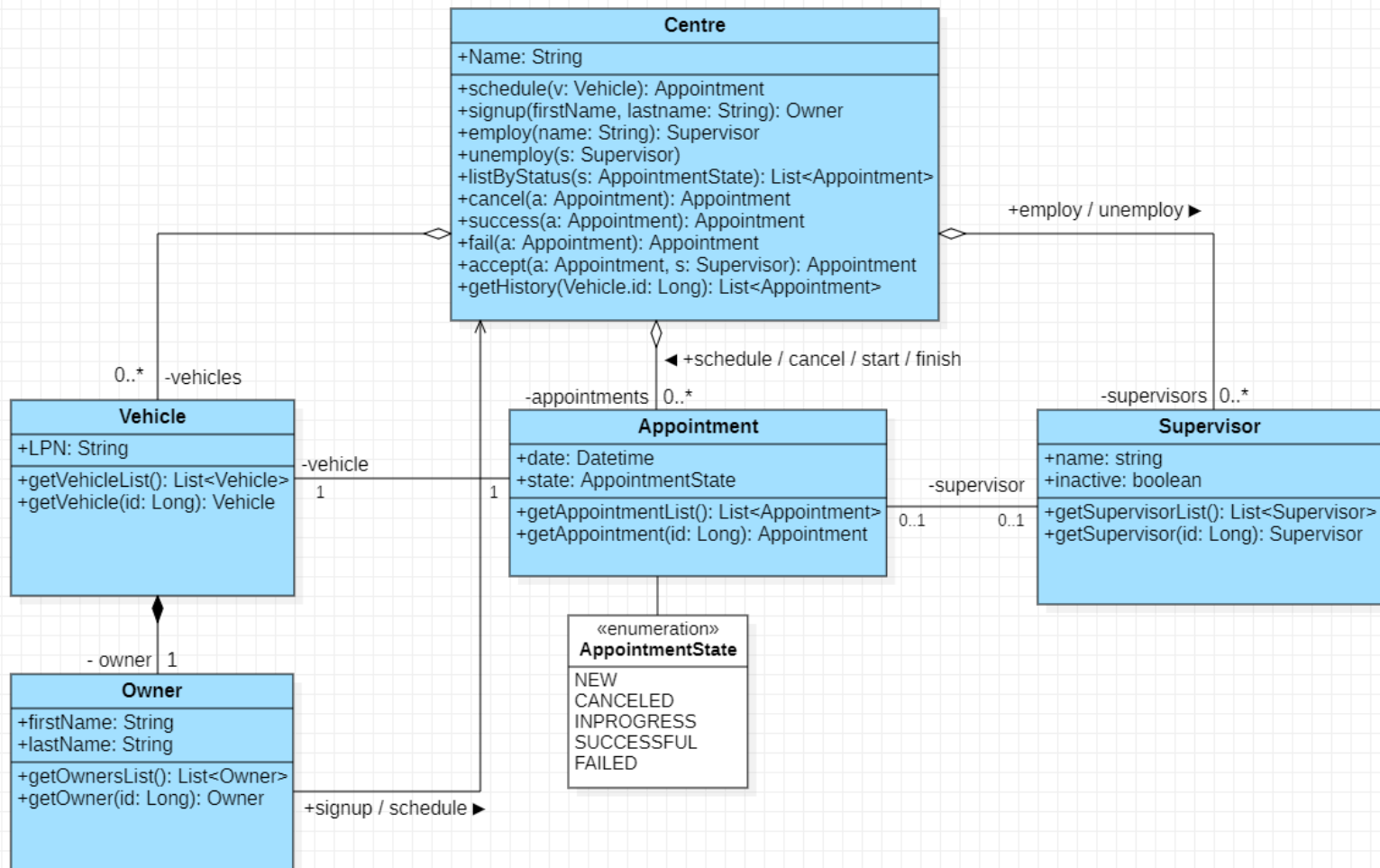
Tribe feladat: műszaki vizsga „M”

Kallai Péter

Tervezés

- Szempontok:
 - Egyszerű, átlátható struktúra
 - OOP megközelítés
 - WEB alkalmazás
 - REST API
 - Adatbázis-integráció

Osztálydiagram



Választott technológia

- Java nyelv
- REST API Spring boot segítségével
 - Spring WEB starter: beépülő Tomcat szerver, REST mapping
- JPA (Java Persistence API)
- H2 in-memory database (default, leállításig perzisztens)
- Dokumentáció/ „frontend”: SwaggerUI

Buzzwords

- REST mapping

```
27 // POST /signup
28 @PostMapping("/signup")
29 Owner signupOwner(@RequestBody Owner o) {
30     Centre centre = new Centre();
31     return centre.signup(o.getFirstName(), o.getLastName());
32 }
33
34 // POST /employ
35 @PostMapping("/employ")
36 Supervisor employSupervisor(@RequestBody Supervisor s) {
37     Centre centre = new Centre();
38     return centre.employ(s.getName());
39 }
40
41 // POST /employ
42 @PostMapping("/unemploy")
43 void unemploySupervisor(@RequestBody Supervisor s) {
44     Centre centre = new Centre();
45     centre.unemploy(s);
46 }
47
48 // GET /appointments/new
49 @GetMapping("/appointments/new")
50 public List<Appointment> getNewAppointments() {
51     Centre centre = new Centre();
52     return centre.listByStatus(Appointment.AppointmentState.NEW);
53 }
54
55
```

Buzzwords

- RESTful, HATEOAS

```
▼ {  
  "id": 2,  
  "firstName": "Jane",  
  "lastName": "Doe",  
  ▼ "_links": {  
    ▼ "all-owners": {  
      "href": "http://localhost:8080/owners"  
    }  
  }  
}
```

Buzzwords

- Java Persistence API

```
10 @Entity
11 public class Vehicle {
12     //=====
13
14     @Id
15     @GeneratedValue
16     private Long id;
17     private String LPN;
18     private String brand;
19     @ManyToOne
20     private Owner owner;
21
22     //=====
23 }
```

Buzzwords

- H2 database

The screenshot shows the H2 database console interface. On the left is a tree view of the database structure for 'jdbc:h2:mem:testdb'. It includes tables like APPOINTMENT, OWNER, SUPERVISOR, and VEHICLE, as well as sequences and users. The 'APPOINTMENT' table is expanded, showing columns: ID, DATE, STATE, SUPERVISOR_ID, and VEHICLE_ID. The main area on the right contains a toolbar with buttons like 'Run', 'Run Selected', 'Auto complete', and 'Clear'. Below the toolbar, the SQL statement 'SELECT * FROM APPOINTMENT' is entered. The results of the query are displayed in a table with 2 rows and 5 columns. The first row shows ID 9, DATE 2022-07-25 20:23:55.661, STATE 0, SUPERVISOR_ID 7, and VEHICLE_ID 3. The second row shows ID 10, DATE 2022-07-25 20:23:55.669, STATE 0, SUPERVISOR_ID 8, and VEHICLE_ID 5. Below the table, it indicates '(2 rows, 11 ms)'. An 'Edit' button is located at the bottom left of the results area.

jdbc:h2:mem:testdb

- APPOINTMENT
 - ID
 - DATE
 - STATE
 - SUPERVISOR_ID
 - VEHICLE_ID
 - Indexes
- OWNER
- SUPERVISOR
- VEHICLE
- INFORMATION_SCHEMA
- Sequences
 - HIBERNATE_SEQUENCE
 - Current value: 11
- Users
- H2 2.1.214 (2022-06-13)

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM APPOINTMENT

SELECT * FROM APPOINTMENT;

ID	DATE	STATE	SUPERVISOR_ID	VEHICLE_ID
9	2022-07-25 20:23:55.661	0	7	3
10	2022-07-25 20:23:55.669	0	8	5

(2 rows, 11 ms)

Edit

Válaszok

Miért ezeket a technológiákat és feladatot választottad?

> A munkámhoz ez is kell, és jó alkalom volt, hogy ezzel gyakoroljak

Mivel volt gond, elakadás?

> A JPA sajátos feltételes lekérdezései nem annyira jöttek át, custom query-t használtam

Mivel tudtál jól haladni?

> Nagyon könnyű volt megcsinálni az Entity-ket (adatosztályokat), a JPA nagyon kényelmes felületet biztosít

Mit tanultál a feladat megoldása során?

> Megtanultam a Spring segítségével RESTful alkalmazásszervert készíteni

Ha holnap kellene újrakezdeni, mit csinálnál másképp?

> Talán kicsit máshogy tervezném meg az objektum relációkat

Mivel tudnád folytatni a fejlesztést?

> Authentikáció

> Owner, Supervisor és Admin felhasználó származtatása őssosztályból

Köszönöm a figyelmet!