

Semestrální práce z předmětu TS1

1. Popis aplikace

CRUD-app je aplikace, která reprezentovaná jako rozhraní pro existující SQL databázi.

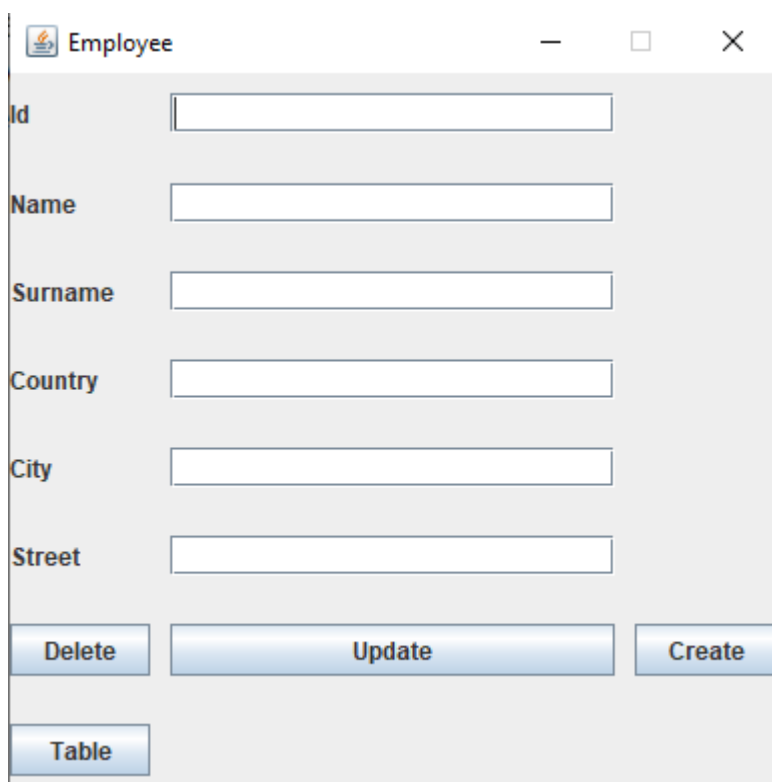
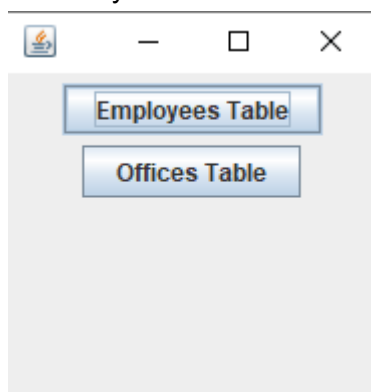
Hlavní feature je možnost momentálně vidět změny v databázi pomocí rozhraní.

Implementace vztahů je realizovaná pomocí knihovny Hibernate.

Databáze obsahuje 2 tabulky: Employees a Office (který obsahuje zaměstnance z Employees). Uživatel bude schopen upravovat a vytvářet zaměstnance a přidávat je do nějakého departmentu v firmě.

Aplikace má dvě části: Front-end je napsán v Swingu, který reprezentuje data z DB a umožňuje pohodlně manipulovat s těmi daty (vytvořit, smazat, obnovit). Back-end, napsaný v Javě, který se stará o bezpečný přenos a validaci těch dat. Komunikuje s ČVUT FEL serverem PostgreSQL.

Elementy GUI:

A screenshot of a Java Swing window titled 'Employee'. It has a standard title bar. The main area contains a form with six labels on the left and corresponding text input fields on the right: 'Id', 'Name', 'Surname', 'Country', 'City', and 'Street'. Below the input fields, there are three buttons: 'Delete', 'Update', and 'Create'. At the bottom left, there is a button labeled 'Table'.

2.Návrh testovací strategie

2.1.Části aplikace

Aplikace je členěna v klasické MVC architektuře, takže má základní dělení na části Model, View a Controller. Ty se pak dále dělí

2.1.1.Controller

EmployeesDAO/EmployeeService - chrání v sobě metody pro připojení k entitě Employees, nastavení hodnot.

OfficeDAO/OfficeService - chrání v sobě metody pro připojení k entitě Office, nastavení hodnot.

2.1.2.Model

Employees - základní entita databázi, která obsahuje data, která se týče zaměstnanců

Office - základní entita databázi, která obsahuje data, která se týče firem, v kterých pracují zaměstnanci.

2.1.3.View

Window - zobrazuje formuláře a tabulku Employees

Window2 - zobrazuje formuláře a tabulku Office

3.Prioritizace částí aplikace

Část	Možné poškození	Vysvětlení možného poškození	Pravděpodobnost selhání	Vysvětlení pravděpodobnosti selhání
Parsing dat z tabulek Office a Employees	H	Data v DB nebude správná	M	Část systému je závislá na mechanismu ošetření nevalidních vstupů
Reprezentace dat	L	Nekorektní práce celého systému	M	Může selhat připojení k internetu/serveru s DB nebo

				data může být nevalidní/nekorrektní.
Správné mazání, ukládání, obnovení dat v DB	M	Klient bude muset přidat data znovu	H	Může selhat připojení k internetu/serveru s DB

4. Test levels

Quality characteristic Část systému/funkce	Třída rizika	Test levels					Unit test	Procesní test
		Revize	Vývojářské	Systémové	UAT	Test v produkci		
Bezpečnost								
Správné mazání, ukládání, obnovení dat v DB	A	Ano		Vysoká			Ano	Ano
Bezchybná funkcionality								
Parsing dat z tabulek	A	Ano	Střední	Vysoká	Vysoká	Ano	Ano	
Reprezentace dat	C			Nizká			Ano	

5. Testovací scénáře

5.1. Testy vstupů - třídy ekvivalence

- Employees id

<0	0<=
----	-----

Nevalidní z technického a business pohledu. Validní z technického a business pohledu.

- Office id

<0	0<=
----	-----

Nevalidní z technického a business pohledu. Validní z technického a business pohledu.

- Jméno a příjmení

1234%:?	Dmitrij Shevchenko
---------	--------------------

Nevalidní z technického a business pohledu. Validní

- Chair number

<0	0-1000	>1000
----	--------	-------

Nevalidní z technického a business pohledu.

Validní

Nevalidní z technického a business pohledu.

- Cabinet number

<0	0-60	>60
----	------	-----

Nevalidní z technického a business pohledu.

Validní

Nevalidní z technického a business pohledu.

- Adresa

Czech republic, Bratislava, Ulice 23	Czech republic, Praha, Ulice 23
--------------------------------------	---------------------------------

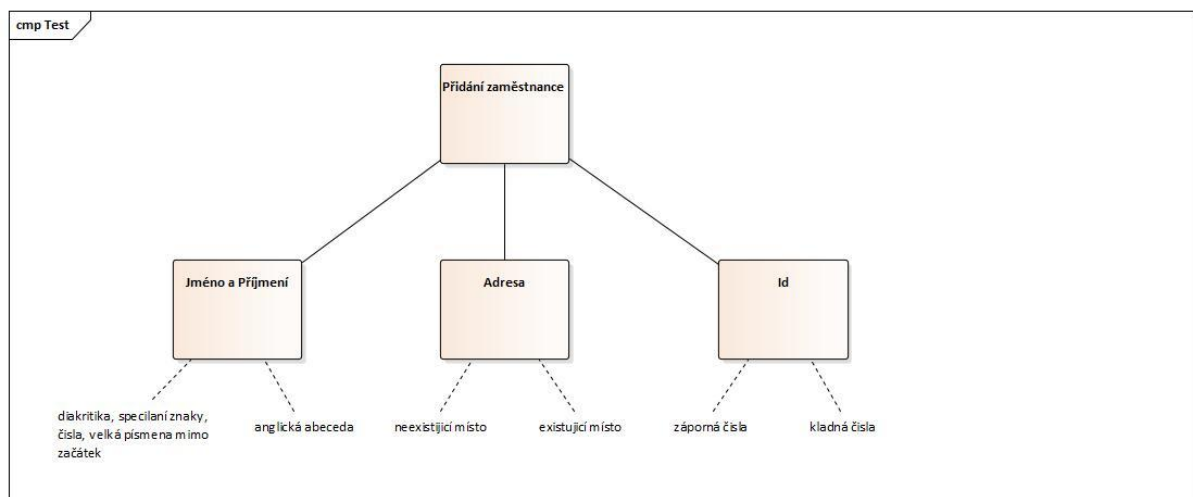
Nevalidní z business pohledu

Validní z business pohledu

5.2. Testy vstupů - kombinace dat

-Přidání zaměstnance do tabulky Employees.

Vstupy podle EC pro formulář:

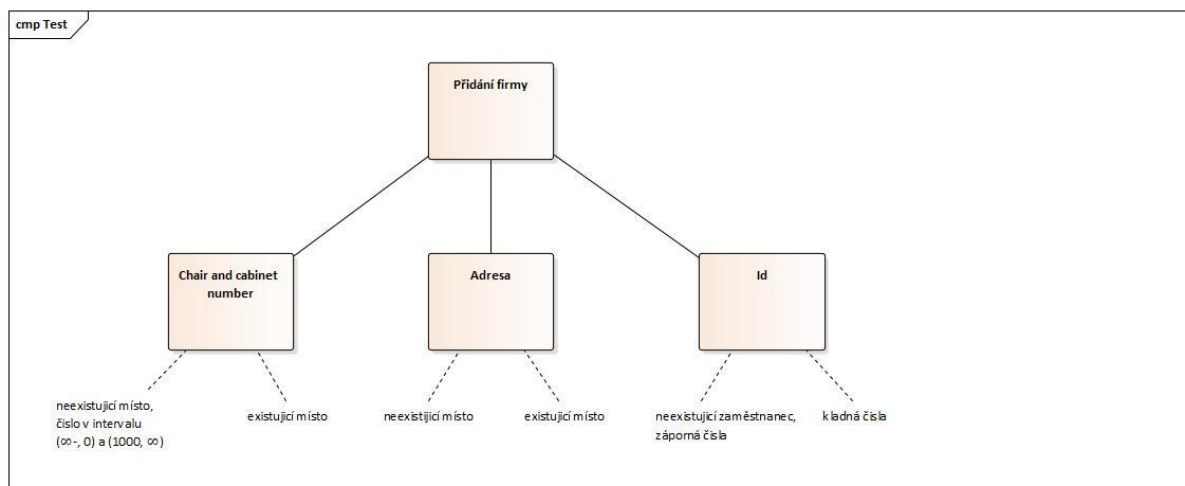


Kompletní pokrytí všemi možnými kombinacemi – 16 variant k otestování:

Test Case	Id	Name_and_Surname	Adresa			
1	-1	Dmitrij_Shevchenko	CzechRepublic_Praha_Ulice23			
2	-1	Dmitri21_Sheebv	CzechRepublic_Bratislava_Ulice23			
3	-1	E23213@@#	CzechRepublic_Praha_Ulice23			
4	-1	_23#@#	CzechRepublic_Bratislava_Ulice23			
5	-10	Dmitrij_Shevchenko	CzechRepublic_Bratislava_Ulice23			
6	-10	Dmitri21_Sheebv	CzechRepublic_Praha_Ulice23			
7	-10	E23213@@#	CzechRepublic_Bratislava_Ulice23			
8	-10	_23#@#	CzechRepublic_Praha_Ulice23			
9	0	Dmitrij_Shevchenko	CzechRepublic_Bratislava_Ulice23			
10	0	Dmitri21_Sheebv	CzechRepublic_Praha_Ulice23			
11	0	E23213@@#	CzechRepublic_Praha_Ulice23			
12	0	_23#@#	CzechRepublic_Praha_Ulice23			
13	23	Dmitrij_Shevchenko	CzechRepublic_Bratislava_Ulice23			
14	23	Dmitri21_Sheebv	CzechRepublic_Praha_Ulice23			
15	23	E23213@@#	CzechRepublic_Bratislava_Ulice23			
16	23	_23#@#	CzechRepublic_Praha_Ulice23			

-Přidání zaměstnanci do tabulky Office.

Vstupy podle EC pro formulář:

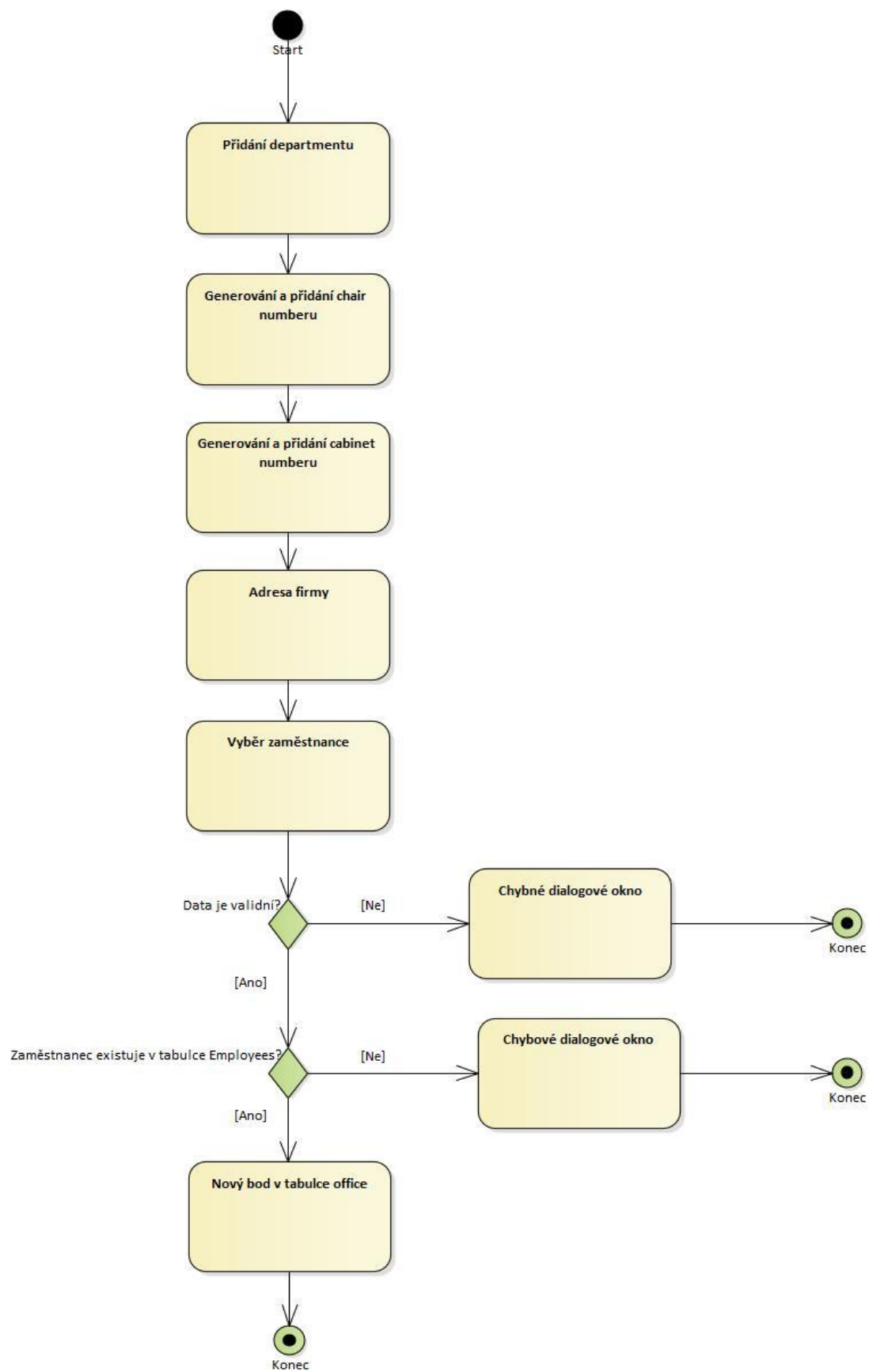


Kompletní pokrytí všemi možnými kombinacemi – 30 variant k otestování:

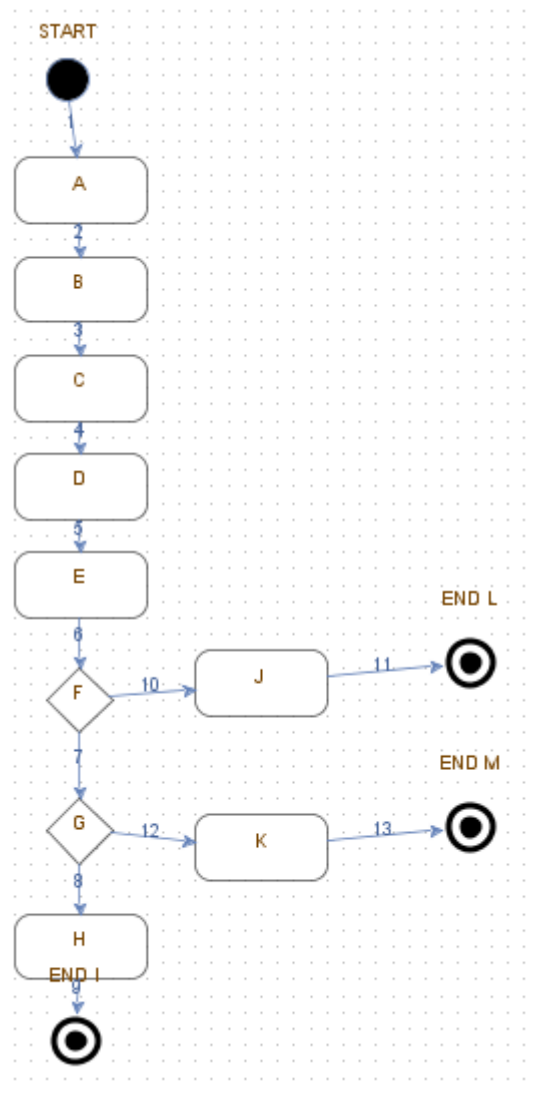
Test Case	Departmer	Chair_num	Cabinet_nu	Adresa			
1	-3	-123	-2	CzechRepublic_Praha_Ulice23			
2	0	-123	0	CzechRepublic_Bratislava_Ulice23			
3	20	-123	50	CzechRepublic_Praha_Ulice23			
4	-10	-123	60	CzechRepublic_Bratislava_Ulice23			
5	-3	-123	61	CzechRepublic_Bratislava_Ulice23			
6	0	-5	-2	CzechRepublic_Praha_Ulice23			
7	20	-5	0	CzechRepublic_Bratislava_Ulice23			
8	-10	-5	50	CzechRepublic_Praha_Ulice23			
9	-3	-5	60	CzechRepublic_Praha_Ulice23			
10	0	-5	61	CzechRepublic_Praha_Ulice23			
11	20	0	-2	CzechRepublic_Bratislava_Ulice23			
12	-10	0	0	CzechRepublic_Praha_Ulice23			
13	-3	0	50	CzechRepublic_Bratislava_Ulice23			
14	0	0	60	CzechRepublic_Praha_Ulice23			
15	20	0	61	CzechRepublic_Praha_Ulice23			
16	-10	10	-2	CzechRepublic_Bratislava_Ulice23			
17	-3	10	0	CzechRepublic_Praha_Ulice23			
18	0	10	50	CzechRepublic_Bratislava_Ulice23			
19	20	10	60	CzechRepublic_Praha_Ulice23			
20	-10	10	61	CzechRepublic_Praha_Ulice23			
21	-3	1000	-2	CzechRepublic_Bratislava_Ulice23			
22	0	1000	0	CzechRepublic_Praha_Ulice23			
23	20	1000	50	CzechRepublic_Bratislava_Ulice23			
24	-10	1000	60	CzechRepublic_Praha_Ulice23			
25	-10	1000	61	CzechRepublic_Praha_Ulice23			
26	-3	1001	-2	CzechRepublic_Bratislava_Ulice23			
27	0	1001	0	CzechRepublic_Praha_Ulice23			
28	20	1001	50	CzechRepublic_Praha_Ulice23			
29	-10	1001	60	CzechRepublic_Bratislava_Ulice23			
30	-3	1001	61	CzechRepublic_Bratislava_Ulice23			

6. Testy průchodů

-Přidání zaměstnance do nového departementu ve firmě.



Testovací graf




Testovací kombinace pro TDL=2

 Test situations 1, TDL= 2, ALG= PCT. ✕

Sub-combinations of edges	Test situations
Node	Sub-combinations of edges
A	1 - 2
E	5 - 6
H	8 - 9
J	10 - 11
B	2 - 3
D	4 - 5
F	6 - 7 6 - 10
K	12 - 13
G	7 - 8 7 - 12
C	3 - 4

Průchody

 Test situations 1, TDL= 2, ALG= PCT. ✕

Sub-combinations of edges	Test situations
No.	Test sequence
1	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9
2	1 - 2 - 3 - 4 - 5 - 6 - 10 - 11
3	1 - 2 - 3 - 4 - 5 - 6 - 7 - 12 - 13

7.Detailní testovací scénáře

Parametr	Obsah
Id testu	<code>officeCreateAndDeleteTest</code>
Název testu	Přidání a odstranění zaměstnance z firmy
Hloubka detailu	Střední
Shrnutí testu	Zadání platných údajů, pozitivní průchod

Popis testu	Přidáváme nového zaměstnanci do firmy pomocí ho Jména, Příjmení, Adresy a Personal ID a zároveň ho Cabinet/Chair numberu, adresy firmy. System kontroluje jestli toto ID už existuje nebo naopak tento zaměstnanec neexistuje v tabulce Employees. Pak smažeme z firmy tohoto zaměstnanci.
Vstupní podmínky	Existující employee
Testovací data	Employee: id = employees.size() + 1; name = "Dmitrij"; surname = "Shevchenko"; country = "Cz"; city = "Praha"; street = "Ulice2334"; employee = employees.size() + 1; department = offices.size() + 1; chair_number = "454"; cabinet_number = "324"; country = "Cz"; city = "Praha"; street = "Ulice2323";
Očekávaný výsledek	V Employees tabulce se objeví zaměstnanec "Dmitrij Shevchenko" a v tabulce Office se nic neobjeví
Autor	Shevchenko Dmitriy
Priorita	Vysoká
Testovací prostředí	Database PostgreSQL

Parametr	Obsah
Název kroku testu	Přidání zaměstnanci do firmy
Očekávaný krok testu	Nastavení setterů entity office
Očekávaný výsledek kroku testu	Vytvoření nového řádku s existujícím employee
Název kroku testu	Odstranění zaměstnance z firmy
Očekávaný krok testu	Výběr stejného řádku z 1.kroku

Očekávaný výsledek kroku testu	Odstranění řádku s existujícím employee
--------------------------------	---

8. Popis testů

-Employees

`createTest()` - vytváří testovou entitu pomocí Mockito a ověří jestli metod funguje.

`incorrectFindEmployee()` - vytváříme entitu s `Id = 10000`, ale budeme hledat v tabulce zaměstnanci s `Id = 9999` (který neexistuje) zase pomocí Mockito. Ověřujeme správnost metody hledání.

`employeeCreateAndUpdateAndDelete()` - vytváří novou entitu `employee`, uloží do databáze, obnoví data a potom prostě odstraní odtud ji.

`employee2CreateAndDelete()` - vytváří 2krát nové entity `employee` potom odstraní jen tu první z databáze.

`employee2CreateException()` - snaží se vytvořit 2krát nové entity se stejným `Id`.

-DaoTest

`testDaoMethodFindById()` - pomocí Mockito testujeme správnost fungování metody `findById()` knihovny Hibernate.

`testDaoMethodFindAll()` - pomocí Mockito testujeme správnost fungování metody `findAll()` knihovny Hibernate.

-Office

`testSave()` - pomocí Mockito testujeme správnost fungování metody `save()` pro tabulku `Office` knihovny Hibernate.

`officeCreateAndDeleteTest()` - vytváří novou entitu `office`, uloží do databáze a odstraní odtud ji.

`officeCreateAndUpdateAndDelete()` - vytváří novou entitu `office`, uloží do databázi, obnoví data a potom prostě odstraní odtud ji.

`officeCreateException()` - snaží se nastavit zaměstnanci, který neexistuje v tabulce `Employees`.

`officeDeleteException()` - snaží se smazat zaměstnanci, který neexistuje v tabulce `Employees`.

`employeesLinkWithDepartmentExceptionAndDelete()` - snaží se přidat dvou zaměstnance do stejného departmentu.