

Zadání

Vaším úkolem je rozšířit projekt s evidencí studentů, který jsme vyvíjeli během cvičení.

Implementovali jsme třídu **Register** s několika metodami, které umožňují vkládat nové studenty do evidence, vyhledávat je a mazat. V rámci tohoto úkolu byste do třídy měli přidat další sadu metod, jejichž popis je uvedený dále. Kromě implementace metod ve třídě **Register** je třeba připravit i odpovídající testy ve třídě **RegisterTest**.

Metody:

✓ **public Set<String> getEmailsByFaculty(Faculty faculty)**

- Metoda vrátí množinu emailových adres všech studentů z evidence studujících na fakultě předané v parametru. Metoda vždy vrátí množinu (*kteřá může být prázdná*), nikdy null.
- Příklady volání (*parametr metody, obsah evidence studentů v okamžiku volání metody, očekávaný výsledek*):

Parametr	Obsah evidence studentů (včetně důležitých atributů)	Výsledek metody
Faculty.FIS	<ul style="list-style-type: none"> student(id: 1, fakulta: FIS, email: 1@vse.cz, ...) student(id: 2, fakulta: FIS, email: 2@vse.cz, ...) student(id: 3, fakulta: NF, email: 3@vse.cz, ...) 	<ul style="list-style-type: none"> „1@vse.cz“ „2@vse.cz“
Faculty.NF	<ul style="list-style-type: none"> student(id: 1, fakulta: FIS, email: 1@vse.cz, ...) student(id: 2, fakulta: FIS, email: 2@vse.cz, ...) 	(prázdná množina)
null	<ul style="list-style-type: none"> student(id: 1, fakulta: FIS, email: 1@vse.cz, ...) student(id: 2, fakulta: FIS, email: 2@vse.cz, ...) 	(prázdná množina)

✓ **public Set<Student> getStudentsByCity(String city)**

- Metoda vrátí množinu všech studentů z evidence, kteří bydlí ve městě předaném v parametru. Metoda vždy vrátí množinu (*kteřá může být prázdná*), nikdy null.
- Student může mít zadané 2 adresy – adresu trvalého pobytu a kontaktní adresu. Metoda bude při vyhodnocování preferovat město z kontaktní adresy. Pokud nebude vyplněná, vyhodnotí se město z adresy trvalého pobytu. Pokud nebude vyplněná žádná adresa, daný student ve výsledné množině vůbec nebude.
- Příklady volání (*parametr metody, obsah evidence studentů v okamžiku volání metody, očekávaný výsledek*):

Parametr	Obsah evidence studentů (včetně důležitých atributů)	Výsledek metody
„Praha“	<ul style="list-style-type: none"> student(id: 1, trvalý pobyt: Praha, ...) student(id: 2, kont. adresa: Praha, ...) student(id: 3, trvalý pobyt: Praha, kont. adresa: Brno, ...) student(id: 4, kont. adresa: Ostrava, ...) student(id: 5, ...) student(id: 6, trvalý pobyt: Brno, kont. adresa: Praha, ...) 	<ul style="list-style-type: none"> student(id: 1, ...) student(id: 2, ...) student(id: 6, ...)
„Brno“	<ul style="list-style-type: none"> student(id: 1, trvalý pobyt: Praha, ...) student(id: 2, kont. adresa: Ostrava, ...) 	(prázdná množina)
null	<ul style="list-style-type: none"> student(id: 1, trvalý pobyt: Praha, ...) student(id: 2, kont. adresa: Ostrava, ...) 	(prázdná množina)

✓ **public Student getOldestStudent()**

- Metoda vrátí nejstaršího studenta z evidence. Pokud je evidence prázdná, vrátí null.
- Pokud je v evidenci více ‚nejstarších‘ studentů (*kteří mají shodný věk, aniž by existoval nějaký starší student*), metoda libovolným způsobem vybere a vrátí jednoho z nich.
- Příklady volání (*obsah evidence studentů v okamžiku volání metody, očekávaný výsledek*):

Obsah evidence studentů (včetně důležitých atributů)	Výsledek metody
<ul style="list-style-type: none"> student(id: 1, věk: 25, ...) student(id: 2, věk: 32, ...) student(id: 3, věk: 27, ...) 	student(id: 2, ...)
<ul style="list-style-type: none"> student(id: 1, věk: 31, ...) student(id: 2, věk: 28, ...) student(id: 3, věk: 31, ...) 	student(id: 1, ...) nebo student(id: 3, ...)
(prázdná množina)	null

✓ **public Student getSecondOldestStudent()**

- Metoda vrátí druhého nejstaršího studenta z evidence (*studenta s druhým nejvyšším věkem*). Pokud je evidence prázdná, vrátí null. Pokud evidence obsahuje pouze jednoho studenta, vrátí ho.
- Pokud je v evidenci více „nejstarších“ studentů (*kteří mají shodný věk, aniž by existoval nějaký starší student*), metoda libovolným způsobem vybere a vrátí jednoho z nich.
- Pokud je v evidenci jeden nejstarší student a více studentů s druhým nejvyšším věkem, metoda libovolným způsobem vybere a vrátí jednoho z nich.
- Příklady volání (*obsah evidence studentů v okamžiku volání metody, očekávaný výsledek*):

Obsah evidence studentů (včetně důležitých atributů)	Výsledek metody
<ul style="list-style-type: none"> student(id: 1, věk: 25, ...) student(id: 2, věk: 32, ...) student(id: 3, věk: 27, ...) student(id: 4, věk: 36, ...) 	student(id: 2, ...)
<ul style="list-style-type: none"> student(id: 1, věk: 31, ...) student(id: 2, věk: 28, ...) student(id: 3, věk: 31, ...) student(id: 4, věk: 30, ...) 	student(id: 1, ...) nebo student(id: 3, ...)
<ul style="list-style-type: none"> student(id: 1, věk: 31, ...) student(id: 2, věk: 28, ...) student(id: 3, věk: 30, ...) student(id: 4, věk: 30, ...) 	student(id: 3, ...) nebo student(id: 4, ...)
<ul style="list-style-type: none"> student(id: 1, věk: 28, ...) 	student(id: 1, ...)
(prázdná množina)	null

✓ **public boolean[] addStudents(List<Student> newStudents)**

- Metoda přidá do evidence všechny studenty předané v parametru (*seznamu*).
- Pro přidávání studentů platí stejné podmínky, jako v metodě **addStudent()**, kterou jsme implementovali během cvičení – tzn. student musí mít nastavenou fakultu a student ještě nesmí být v evidenci.
- Návratovou hodnotou metody je pole logických hodnot – jeho velikost odpovídá velikosti seznamu v parametru a pro každého studenta bude obsahovat hodnotu true, nebo false, dle toho, zda daný student byl do evidence přidán, nebo ne.
- Příklady volání (*parametr metody, obsah evidence studentů v okamžiku volání metody, očekávaný výsledek*):

Parametr	Obsah evidence studentů (včetně důležitých atributů)	Výsledek metody
<ul style="list-style-type: none"> student(id: 1, fakulta: FIS) student(id: 3, fakulta: NF) student(id: 4) student(id: 5, fakulta: FMJH) 	<ul style="list-style-type: none"> student(id: 1, fakulta: FIS) student(id: 2, fakulta: NF) 	<ul style="list-style-type: none"> false true false true
(prázdný seznam)	<ul style="list-style-type: none"> student(id: 1, fakulta: FIS) student(id: 2, fakulta: NF) 	(pole s nulovou velikostí)
null	<ul style="list-style-type: none"> student(id: 1, fakulta: FIS) student(id: 2, fakulta: NF) 	(pole s nulovou velikostí)

Termín a hodnocení

- Je potřeba přidat nové metody a odpovídající automatizované testy.
- Termín pro splnění úkolu je **6. 12. 2020**.
- Za úkol můžete získat **6 bodů**.

Podklady

Podkladový projekt najdete v archivu se zadáním v adresáři **du-03-projekt**. Jedná se o BlueJ projekt s hotovou implementací metod ze cvičení, oproti cvičení jsou doplněné testy pro všechny metody.

Odevzdání

Po vypracování úkolu ho odevzdejte do odevzdávnice [D. ú. 3 : Evidence studentů](#).

Před odevzdáním vymažte v adresáři projektu všechny soubory s příponami **.ctxt** a **.class** a případnou vygenerovanou dokumentaci (*podadresář doc*). Celý adresář s projektem poté zabalte do ZIP archivu.

Doplňte své jméno a datum do dokumentačních komentářů tříd, které jste v rámci úkolu vytvořili nebo upravili, viz následující příklad:

```
/**
 * Třída MojeTrida představuje ...
 *
 * @author: Antonín Kostěj
 * @version: 2020-12-06
 */
public class MojeTrida
{
    ...
}
```