

Напомена: На *Desktop*-у направити директоријум **EclipseWorkspace** уколико не постоји и одабрати га за *workspace* при покретању Еклипсе. Направити пројекат са именом `oop_Ime_Prezime_Indeks_Asistent` (нпр. `oop_Pera_Peric_mi12082_NeM`). У направљеном пројекту, пакет такође назвати исто тако. Код **не сме** имати синтаксних грешака нити избацавања `NullPointerException`-а. Време за рад: **2.5 сата**.
Иницијали асистента: **BS** (Биљана Стојановић), **NiM** (Никола Милев) и **NeM** (Немања Мићовић).

1. Направити апстрактну класу `Softver` коју карактеришу атрибути `ime (String)`, `brojLinija (int`, који означава број линија кода) и програмски језик у којем је програм написан (`String`). Имплементирати потребне `get` методе.
2. Направити класу `Editor` која наслеђује класу `Softver` и представља едитор текста. `Editor` се додатно карактерише пољем `najcesciJezik (String)` које означава за који програмски језик се едитор најчешће користи. Имплементирати метод `toString` као у тест примеру.

Пример за метод `toString` (име, број линија, имплементациони језик, језик за који се најчешће користи)

```
[editor]: Vim, 300000, c, c++
```

3. Направити класу `Biblioteka` која наслеђује класу `Softver` и представља библиотеку за програмски језик. Класа се карактерише њеним кратким описом (`String`). Додати неопходне `get` методе као и метод `toString`. Пример за метод `toString` (име, број линија, имплементациони језик, кратак опис)

```
[biblioteka]: boost spirit, 650 000, c++, Biblioteka za kreiranje parsera.
```

4. Направити класу `SoftverskaMreza` која представља мрежу која корисницима омогућава да деле и заједнички развијају софтвер. Класа садржи атрибуте `ime (String)` и `podaci (Map<String, List<Softver>>)`. Мапа пресликава имплементациони језик (нпр. `c++`) у листу софтвера који су већински имплементирани у том језику.

Имплементирати конструктор који не прима аргументе и инстанцира празну мапу (дозвољено је користити било коју имплементацију мапе која имплементира интерфејс `Map`).

Имплементирати метод `toString` тако да враћа списак софтвера као што је приказано на слици 1.

Имплементирати метод `boolean ucitajPodatke(String putanja)` који из датотеке која се налази на прослеђеној путањи учитава податке и смешта их у мапу `podaci`. Уколико су подаци успешно учитани, метод враћа `true`, а иначе враћа `false`.

Датотека садржи прво индикатор о податку (1 ако је едитор, 2 ако је библиотека), потом име софтвера, број линија кода и програмски језик у којем је софтвер већински написан. Након тога следи кратак опис уколико је у питању библиотека, а уколико је едитор онда следи програмски језик за који се едитор најчешће користи.

Пример датотеке:

```
1, Vim, 300 000, c, c++
2, tensorflow, 1 000 000, c++, Biblioteka posvecena masinskom ucenju.
1, Emacs, 25000, c, c
1, Sublime Editor, 320000, c++, JavaScript
2, javafx, 700 000, java, Biblioteka za graficki korisnicki interfejs.
2, qt, 900 000, c++, Biblioteka i radni okvir za programski jezik c++.
2, boost spirit, 650 000, c++, Biblioteka za kreiranje parsera.
2, jquery, 600 000, JavaScript, Biblioteka za jezik javascript.
```

5. У класу `SoftverskaMreza` додати методе:

- `String najpopularnijiJezik()` који проналази програмски језик који има највише у њему написаног софтвера. Уколико их је више, вратити први пронађени.
- `List<Softver> softverZaJezik(String jezik)` који враћа листу програма који су имплементирани у прослеђеном језику. Уколико не постоји ниједан такав програма, вратити `null`.
- `int ukupnoLinijaKodaZaJezik(String jezik)` који враћа укупан број линија кода за софтвер имплементиран у прослеђеном језику. Уколико не постоји ниједан програм, вратити `0`.

- `int ukupnoLinijaKoda()` koji vraća ukupno linija koda koje sadrži softverska mreža.

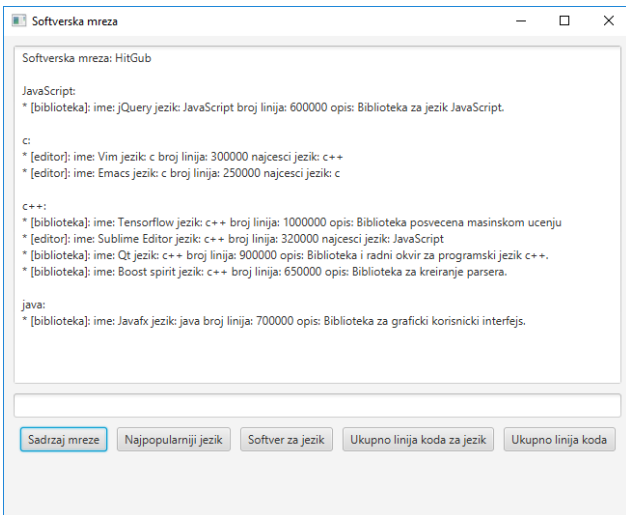
У методима претпоставити да се у мапи налази **барем један** податак са иницијализованом листом која садржи **барем један** софтвер уколико је читање датотеке протекло успешно.

- Направити графички кориснички интерфејс као што је приказано на слици. При покретању програма, инстанцирати објекат класе `SoftverskaMreza`, позвати метод `ucitajPodatke()` и у `TextArea` елемент исписати поруку да ли је читавање било успешно или не. За имплементацију догађаја је дозвољено користити претходно имплементирани методе у класи `SoftverskaMreza`.

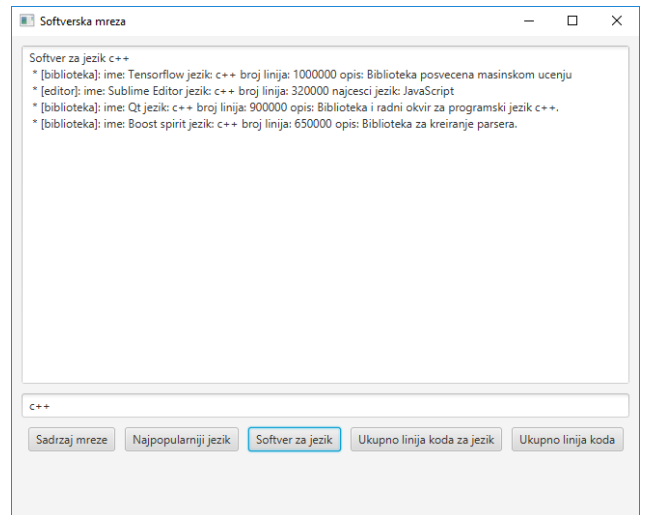
На клик дугмета:

- **Sadržaj mreze** у `TextArea` елемент се уписују подаци о целокупном софтверу који мрежа поседује.
- **Najpopularniji jezik** у `TextArea` елемент се уписује програмски језик који има највише програма у којима је он коришћен за главни део имплементације.
- **Softver za jezik** у `TextArea` елемент исписати целокупан софтвер који је написан у језику који се уписује у `TextField` елемент.
- **Ukupno linija koda za jezik** у `TextArea` елемент исписати укупан број линија кода који софтверска мрежа садржи за језик који је уписан у `TextField` елемент.
- **Ukupno linija koda** у `TextArea` елемент исписати укупан број линија кода који софтверска мрежа садржи.

Уколико је `TextField` елемент празан при обрађивању догађаја које га користи, у `TextArea` елемент исписати поруку "Molimo unesite programski jezik."



Слика 1: Приказивање садржаја мреже



Слика 2: Приказивање софтвера за изабрани језик