1. Uvod

Cilj vježbe je kreirati dvostruko povezanu listu i odgovarajuće metode za rad sa listom. Za kreirane klase i metode potrebno je napisati korisničku dokumentaciju pomoću JAVADoc komentara.

2. Opis vježbe

Od studenta se zahtjeva da kreira klasu koja implementira strukturu podataka - dvostruko povezanu listu i metode koje omogućavaju da se sa tom strukturom može raditi kao sa stogom (stack) i redom (queue).

Da bi se sa podacima moglo raditi kao sa stogom potrebne su slijedeće metode:

- peek – vraća slijedeći element sa stoga ali element ostaje u stogu
- push - stavlja novi element na stog
- pop – dohvaća element sa stoga

Za rad kao sa redom:

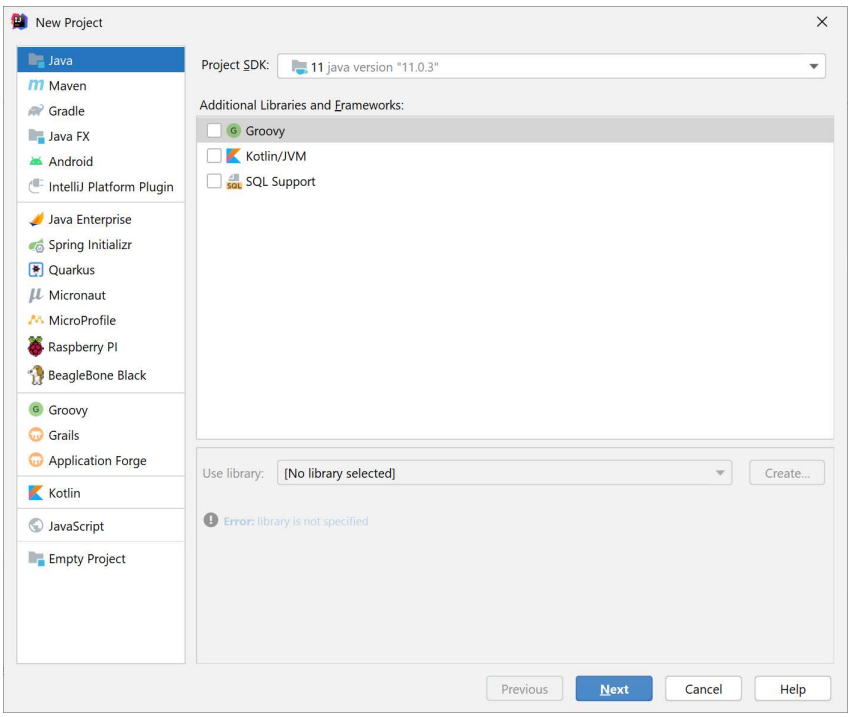
- offer – stavlja element na kraj reda
- poll – dohvaća element sa početka reda
- element – dohvaća element sa početka reda ali element ostaje u redu

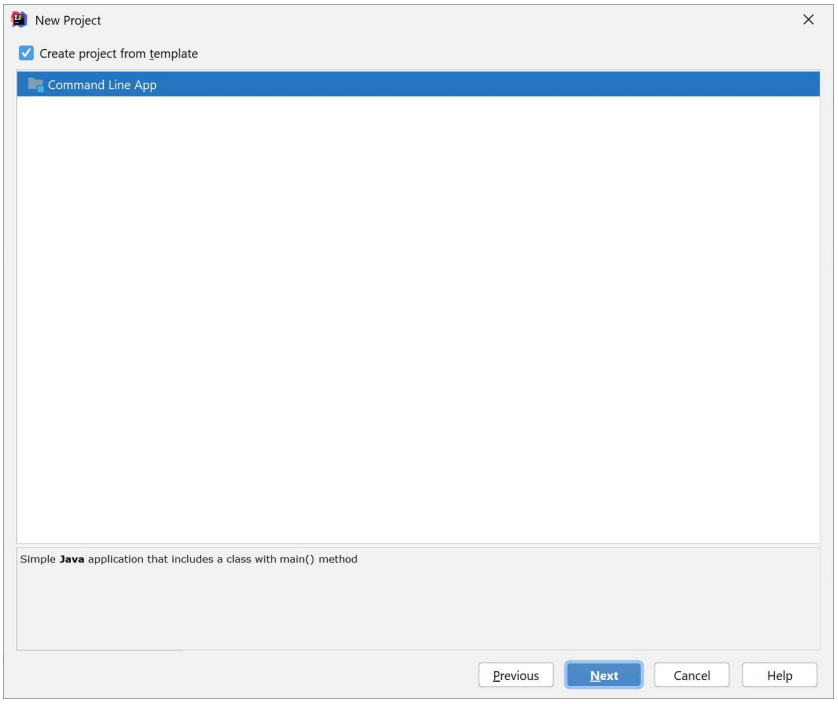
Od ostalih metoda potrebno je implementirati:

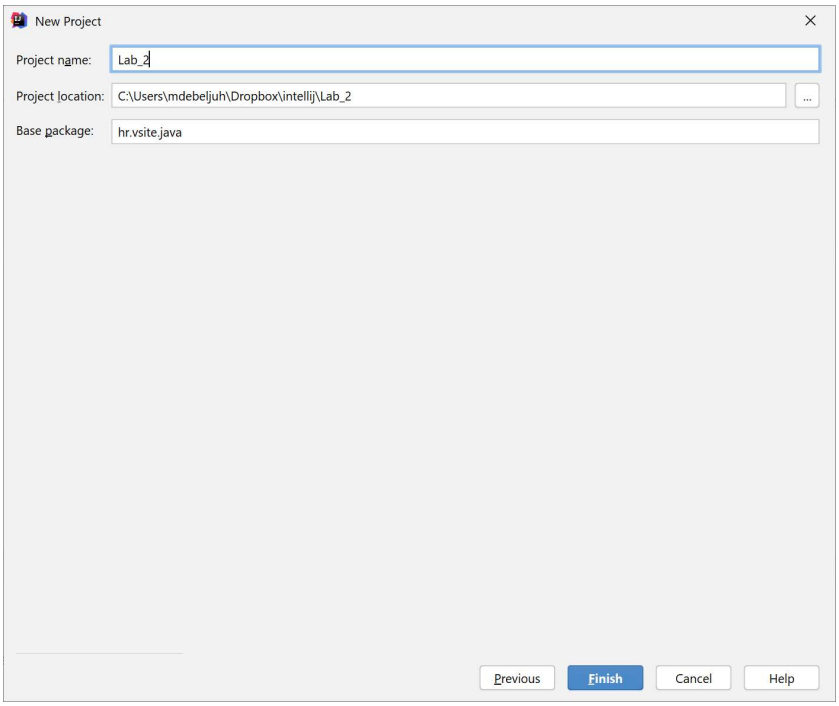
- size – broj elemenata u listi
- isEmpty – da li je lista prazna
- contains - da li lista sadržava neki element
- toString – formatira string sa vrijednostima elemenata u listi

3. Dodatne upute

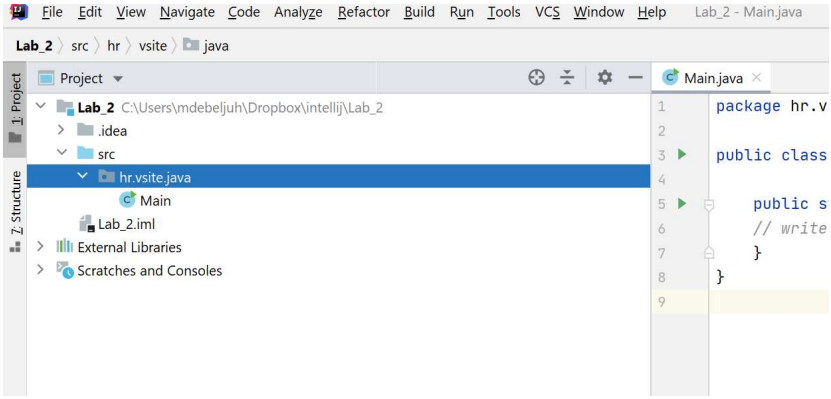
Potrebno je u intellij okruženju napraviti novi Java project File->New->Project.



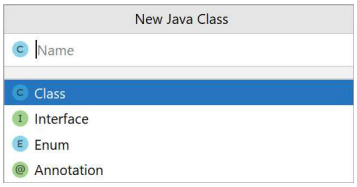




Nakon toga potrebno je kreirati klase za strukturu podataka. Izbornik za kreiranje klase će biti vidljiv ako je u strukturi projekta izabran neki direktorij za izvorne datoteke:



Nove klase se kreiraju putem opcije File->New->Java Class:



Na ekranu za kreiranje klase potrebno je samo definirati naziv klase (počinje sa velikim slovom kao i svaka nova riječ u nazivu npr. BankovniRacun). Paket u kojem će se nalaziti nova klasa ovisi o (polje package) izabranom direktoriju. Naziv paketa se može sastojati od više dijelova odvojenih točkom i svaki dio naziva počinje sa malim slovom (hr.vsite.java).
Kod pisanja implementacije klasa treba voditi računa od pravima pristupa svojstvima (varijablama) i ponašanju (metodama). Prava pristupa se definiraju modifikatora pristupa: private, protected, public i *bez deklaracije* (package private):

Modifikator	klasa	podklasa	package	svijet
private	X			
protected	X	X	X	
public	X	X	X	X
Bez modifikatora	X		X	

Modifikatori pristupa se navode prije deklaracije varijable ili metode:
private int iznos;
private boolean uplata(int iznos){ ...}

4. Dodatak: Pisanje korisničke dokumentacije

U Javi se može direktno u kodu pisati dokumentacija pomoću posebnih JavaDoc komentara. JavaDoc komentari počinju sa /** i završavaju sa */. Takvim komentarima se može dodati opis klasama, sučeljima, konstruktorima, varijablama, metodama i navode se prije elementa koji se opisuje:

```
/**
 * Returns an Image object that can then be painted on the screen.
 * The url argument must specify an absolute URL. The name
 * argument is a specifier that is relative to the url argument.
 * <p>
```

```
 * This method always returns immediately, whether or not the
 * image exists. When this applet attempts to draw the image on
 * the screen, the data will be loaded. The graphics primitives
 * that draw the image will incrementally paint on the screen.
 *
 * @param  url  an absolute URL giving the base location of the image
 * @param  name the location of the image, relative to the url argument
 * @return      the image at the specified URL
 * @see         Image
 */
public Image getImage(URL url, String name) {
       try {
           return getImage(new URL(url, name));
       } catch (MalformedURLException e) {
           return null;
       }
}
```

Najprije se navodi opis elementa i cijeli opis se mora navesti prije ostalih @ elemenata. Prva rečenica treba biti kratki opis elementa (koji se kopira u sažetak paketa ili klase), a nakon toga se navodi duži opis. Osim opisa elementa (klase, metode, varijable, …) posebnim tagovima (počinju sa @) može se definirati opis pojedinog dijela elementa:

- @author (samo za klase i interface, obavezni) – dodaje se naziv autora
- @version (samo za klase i interface, obavezni) – verzija klase ili interface-a
- @param (samo za metode i konstruktore) – opis parametra
- @return (samo za metode) – opis povratne vrijednosti
- @exception (@throws je sinonim uveden u Javadoc 1.2) – opis iznimke koja se može baciti
- @see – definira se referenca na neku drugu klasu, metodu, varijablu
- @since – od koje verzije postoji element
- @deprecated – označava da je element zastario, ne bi se trebao više koristiti

U Intellij se takvi komentari dodaju tako da se započne javadoc komentar (/**) i nakon prelaska u novi redak intellij će generirati prazan javadoc komentar sa osnovnim tagovima za taj element:

```
/**
 * @param args
 */
public static void main(String[] args)
```

Zadatak: Potrebno je dodati opise za sve public članove u vašim klasama.

Više o javadoc komentarima na
http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html

Generiranje javadoc dokumentacije u Intellij alatu:

https://www.jetbrains.com/help/idea/working-with-code-documentation.html#generate-javadoc