

Objektno-orijentisano programiranje – prvi kolokvijum

Zadatak: Biblioteka – termin 1

Napraviti Java konzolnu aplikaciju za kreiranje knjiga i vođenje evidencije o njima unutar biblioteke. Knjige koje se nalaze u *Biblioteci* se dele na *AutorskuKnjizevnost* i *NarodnuKnjizevnost*. *NarodnaKnjizevnost* nema poznatog autora, ali ima *Sakupljaca*, a *AutorskaKnjizevnost* ima autora, tj. *Pisca*, koji može da bude *ProzniPisac* ili *Pesnik*. *ProzniPisac* i *Pesnik* se razlikuju po tome što *ProzniPisac* objavljuje samo *Epiku* kao *TipKnjizevnosti*, a *Pesnik* objavljuje samo *Liriku* kao *TipKnjizevnosti*.

Prvi deo – implementacija UML dijagrama klasa

1. (6p) Implementirati sve elemente na priloženom dijagramu uz sledeće napomene:

- za sva polja dodati get i set metode sa proverama koje se traže u zadatku,
- metode iz klase *Object*, equals i *toString* nadjačati prema proceni na mestima gde je to neophodno za realizaciju zahtevanih funkcionalnosti,
- proizvoljno se mogu dodati konstruktori, metode kojih nema na dijagramu i implementacije ugrađenih interfejsa (na primer *java.lang.Comparable*).

2. Apstraktna klasa *Knjizevnost* predstavlja knjigu koja ima naziv i *TipKnjizevnosti* (*LIRIKA*, *EPIKA*, ili *LIRSKO_EPSKI*).

- Dve instance klase *Knjizevnost* su iste ako su im naziv i *TipKnjizevnosti* isti.

- Klase *NarodnaKnjizevnost* i *AutorskaKnjizevnost* nasleđuju apstraktnu klasu *Knjizevnost*.

AutorskaKnjizevnost ima godinu izdavanja kao atribut (*godina*) i ima autora (*Pisac*). Kako *Pisac* može da bude *ProzniPisac* ili *Pesnik*, *AutorskaKnjizevnost* može za *TipKnjizevnosti* da ima ili *Liriku* ili *Epiku*, u zavisnosti od toga da li je njen autor *ProzniPisac* ili *Pesnik*. *NarodnaKnjizevnost* nema ni godinu izdavanja ni autora, ali ima *Sakupljaca* i može da ima za *TipKnjizevnosti* bilo koju od tri vrednosti.

(Ova ograničenja koja se tiču odgovarajućih tipova književnosti implementirati kroz metodu *objavljuje* iz interfejsa *Objavljivanje*)

3. Interfejs *Objavljivanje* obezbeđuje metodu pomoću koje se kreiraju knjige kao instance klase *Knjizevnost* i implementiraju ga klase *ProzniPisac*, *Pesnik* kao i *Sakupljac* i to na sledeći način:

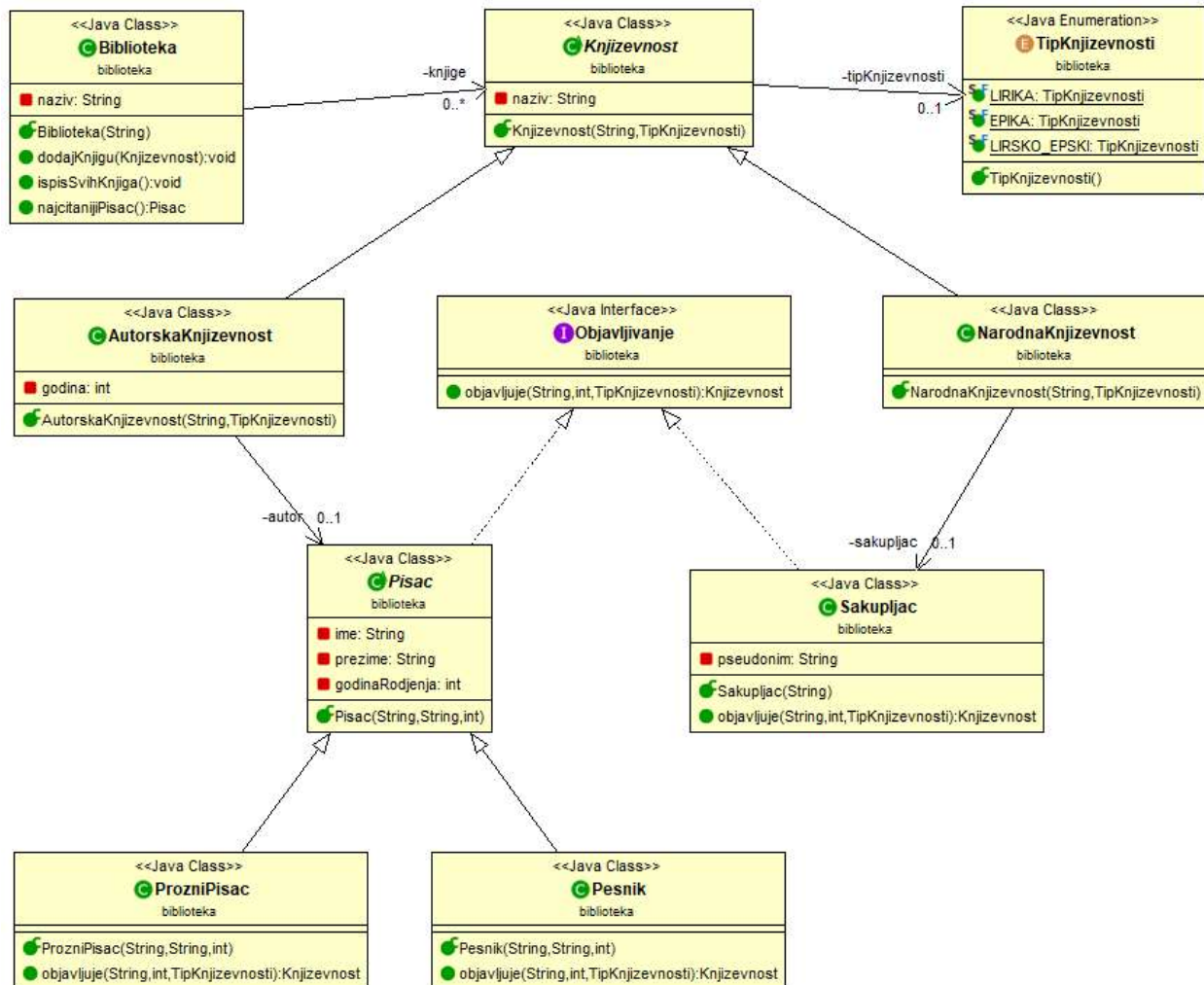
- **(1p)** metoda *objavljuje(naziv, godina, tipKnjizevnosti)* koju implementira *ProzniPisac* od argumenata prima naziv knjige, godinu objavljivanja i *tipKnjizevnosti* i na osnovu naziva, godine, kao i samog objekta nad kojim se poziva, kreira i vraća instancu klase *Knjizevnost* koja za *TipKnjizevnosti* ima *Epiku*.
- **(1p)** metoda *objavljuje(naziv, godina, tipKnjizevnosti)* koju implementira *Pesnik* funkcioniše isto kao i za klasu *ProzniPisac*, samo za *TipKnjizevnosti* umesto *Epike* ima *Liriku*.
- **(1p)** metoda *objavljuje(naziv, godina, tipKnjizevnosti)* koju implementira klasu *Sakupljac* ne uzima u obzir godinu koja se prosleđuje (nije bitno koja godina će da se prosledi), već samo na osnovu naziva, *tipaKnjizevnosti* i samog objekta nad kojim se poziva, kreira i vraća instancu klase *Knjizevnost* koja za *tipKnjizevnosti* može da ima bilo koju od tri vrednosti i postavlja se vrednost prosleđenog argumenta.

4. Apstraktna klasa *Pisac* predstavlja autora i od atributa ima ime, prezime i godinuRodjenja.

- Klase *ProzniPisac* i *Pesnik* nasleđuju apstraktnu klasu *Pisac* i razlikuju se samo po svojoj implementaciji metode *objavljuje*.
- Klasa *Sakupljac* ima jedan atribut, a to je *pseudonim*, i takođe implementira interfejs *Objavljivanje*.

5. Klasa *Biblioteka* ima sledeće metode:

- (1p) *dodajKnjigu(Knjizevnost)* u kolekciju *knjige* dodaje jednu knjigu (instancu *Knjizevnosti*).
- (3p) *ispisSvihKnjiga()* na konzolu ispisuje sve knjige sortirane po nazivu alfabetski rastuće. U prvom redu ispisuje string u formaru „Knjige na raspolaganju u biblioteci NAZIVBIBLIOTEKE: “, a u svakom sledećem redu pojedinačno ispisuje jednu po jednu knjigu na sledeći način: „NAZIVKNJIGE, Autor: IMEAUTORA PREZIMEAUTORA“ (npr. „Ana Karenjina, Autor: Lav Tolstoj“), ako je instanca klase *AutorskaKnjizevnost*, ili „NAZIVKNJIGE, Sakupljac: PSEUDONIMSAKUPLJACA“ (npr. „Srpske narodne pripovijetke, Sakupljac: Vuk Stefanovic Karadzic“), ako je instanca klase *NarodnaKnjizevnost*.
- (4p) *najcitanijiPisac()* vraća instancu *Pisaca* koji ima najveći broj knjiga u *Biblioteci*. Ako više *Pisaca* zadovoljava ovaj uslov, vraća se *Pisac* sa najranijom godinom rođenja.



Drugi deo – proširenje modela i dodatne funkcionalnosti

6. (8p) Dodati mogućnost da se u biblioteci čuvaju podaci o lektirama za osnovnu i srednju školu. Za lektiru se pamti kom razredu je namenjena i za koju školu (osnovna, gimnazija, stručna). Postoje dve vrste lektire, **tipska lektira** i **kombinovana lektira**. Tipska lektira treba da sadrži četiri različite knjige (*Knjizevnosti*) koje imaju isti *TipKnjizevnosti*, dok kombinovana lektira sadrži pet različitih knjiga od kojih su barem dve *Epike*, barem jedna *Lirika* i najviše jedna *Lirsko_epska*. Obezbediti operacije za dodavanje

knjiga (knjizevnosti) u lektiru koje proveravaju data ograničenja. U lektiru ne može da se doda knjiga koja ne postoji u biblioteci. Implementirati i operacije za ispis lektire u fajl, ispisuje se škola za koju je namenjena, razred i spisak knjiga i to ime i prezime autora ili pseudonim sakupljača, naslov knjige i tip književnosti, svaka knjiga u novom redu.

7. (3p) Napraviti klasu *Test* koja će da sadrži *main* metodu. U *main* metodi je potrebno kreirati tri *Pisca*, jednog *Sakupljaca* i jednu *Biblioteku*. Kerirati devet knjiga pozivima metoda *objavljuje* za svakog od *Pisaca* po tri puta, od čega je potrebno da budu barem po tri *Lirike* i tri *Epike*, i da barem dve, a najviše četiri, od devet knjiga budu iste (isti *naziv* i *TipKnjizevnosti*). Kreirati još dve knjige pozivima metoda *objavljuje* za *Sakupljaca*, tako da te knjige imaju *TipKnjizevnosti Lirsko_epski*. Dodati sve knjige u *Biblioteku* i pozvati metodu *ispisSvihKnjiga*, a zatim ispisati i najčitanijeg pisca pomoću poziva metode *najcitanijiPisac*.

8. (2p) Proširiti *main* metodu u klasi *Test* tako da se nakon kreiranja knjiga i njihovog dodavanja u *Biblioteku* formiraju dve *Lektire*, jedna *tipska* i jedna *kombinovana*. Nakon formiranja *Lektira* ispisati kombinovanu lektiru u fajl „kombinovanaLektira.txt“.