



А К А Д Е М И Ј А
ТЕХНИЧКО-ВАСПИТАЧКИХ
СТРУКОВНИХ СТУДИЈА



PROGRAMSKI ALATI ZA RAZVOJ SOFTVERA

Prvi kolokvijum

Sistem za upravljanje inventarom knjižare

Implementirajte sistem za praćenje i upravljanje inventarom u knjižari. Sistem treba da obuhvata klase za knjige, sa specifičnim vrstama knjiga kao što su **udžbenici, slikovnice i romani**, kao i mogućnost ažuriranja inventara, upravljanje kupcima i narudžbinama. Cilj je da napravite osnovnu strukturu koja omogućava praćenje informacija o knjigama, upravljanje narudžbinama kupaca i testiranje sistema.

1. Dizajn klasa

Glavne klase koje treba implementirati:

- **Book** - Apstraktna klasa koja predstavlja osnovne karakteristike knjiga.
- **Textbook, PictureBook, Novel** - Podklase klase Book koje nasleđuju zajedničke atribute ponašanja, a takođe imaju i svoje specifične atribute.
- **Inventory** - Klasa koja sadrži kolekciju Book objekata i pruža metode za dodavanje, uklanjanje i pretragu knjiga u inventaru.
- **Order** - Klasa koja predstavlja narudžbinu i sadrži listu knjiga koje je kupac naručio.
- **Customer** - Klasa koja predstavlja kupca sa njegovim kontakt podacima i listom narudžbina.

Osnovna struktura klasa:

Book (apstraktna klasa):

- **Atributi:**
 - title - naziv knjige.
 - author - autor knjige.
 - price - cena knjige.
 - quantity - količina na zalihama.
- **Metode:**
 - get_details() - apstraktna metoda koja vraća osnovne informacije o knjizi.

Textbook (nasleđuje Book):

- **Dodatni atributi:**
 - subject - predmet (npr. matematika, fizika).
 - academic_level - nivo obrazovanja (npr. osnovna, srednja škola, fakultet).
- **Metode:**

- `get_details()` - vraća detalje specifične za udžbenik.

PictureBook (nasleđuje Book):

- **Dodatni atributi:**
 - `illustrator` - ime ilustratora knjige.
 - `recommended_age` - preporučeni uzrast za decu.
- **Metode:**
 - `get_details()` - vraća detalje specifične za slikovnicu.

Novel (nasleđuje Book):

- **Dodatni atributi:**
 - `genre` - žanr romana (npr. drama, triler).
 - `is_bestseller` - boolean vrednost koja označava da li je roman bestseler.
- **Metode:**
 - `get_details()` - vraća detalje specifične za roman.

Inventory:

- **Atributi:**
 - Lista objekata klase Book.
- **Metode:**
 - `add_book(book)` - dodaje knjigu u inventar.
 - `remove_book(book)` - uklanja knjigu iz inventara.
 - `search_book(title)` - pretražuje inventar po naslovu knjige.
 - `display_inventory()` - prikazuje sve knjige u inventaru.

Order:

- **Atributi:**
 - `order_id` - jedinstveni identifikator narudžbine.
 - Lista knjiga koje je kupac naručio (lista objekata klase Book).
 - `total_price` - ukupna cena narudžbine, izračunava se kao zbir cena svih knjiga u narudžbini.
- **Metode:**
 - `add_book_to_order(book)` - dodaje knjigu u narudžbinu.
 - `calculate_total_price()` - računa ukupnu cenu knjiga u narudžbini.

Customer:

- **Atributi:**
 - `customer_id` - jedinstveni identifikator kupca.
 - `name` - ime kupca.
 - `phone` - telefon kupca,
 - `address` - adresa kupca,
 - `email` - email kupca
 - Lista narudžbina kupca (lista objekata klase Order).
- **Metode:**
 - `place_order(order)` - dodaje narudžbinu u listu narudžbina kupca.

- `get_order_history()` - prikazuje sve narudžbine kupca.

2. Use-Case dijagram

U ovom dijagramu, neophodno je prikazati osnovne akcije koje korisnici mogu obavljati u sistemu za upravljanje inventarom knjiga. Dijagram treba da prikazuje interakcije korisnika sa različitim funkcijama sistema. Ključni akteri u ovom sistemu su **Kupac** i **Administrator**. Predlog operacija je dat u nastavku:

Kupac može:

- Prikazati listu dostupnih knjiga.
- Pretražiti knjigu po naslovu ili autoru.
- Kreirati narudžbinu i dodati knjige u korpu.
- Pregledati istoriju svojih narudžbina.

Administrator može:

- Dodavati nove knjige u inventar.
- Uklanjati knjige iz inventara.
- Ažurirati podatke o knjigama (naslov, autor, cena, količina).
- Prikazati i upravljati svim narudžbinama kupaca.

3. UML klasni dijagram

UML klasni dijagram treba da prikaže sledeće:

- Klasa Book kao apstraktna klasa sa osnovnim atributima i apstraktnom metodom `get_details()`.
- Nasledene klase Textbook, PictureBook, i Novel, sa svojim osnovnim i dodatnim atributima.
- Klasa Inventory, prikazana kao kolekcija objekata klase Book.
- Klasa Order, koja sadrži listu objekata klase Book.
- Klasa Customer, koja sadrži listu objekata klase Order.
- U dijagramu obavezno povežite sve entitete odgovarajućim strelicama. Neke od veza koje možete koristiti su **Generalizacija**, **Kompozicija** i **Agregacija**. Obavezno opišite ove veze zašto baš njih koristite i koje su njihove prednosti u odnosu na neke druge (par rečenica).

4. Zahtevi za testove

Napišite 5 testova za testiranje funkcionalnosti sistema koristeći JUnit (Java) ili pytest (Python), sa različitim assert izrazima. Testovi treba da pokriju sledeće funkcionalnosti:

1. **Dodavanje knjige u inventar:** Testirajte da li se knjiga uspešno dodaje u listu knjiga u klasi Inventory. Koristite assert da proverite da li se povećava broj knjiga u inventaru.
2. **Kreiranje narudžbine sa knjigama:** Kreirajte Order objekat sa listom knjiga i testirajte da li se ukupan iznos narudžbine računa ispravno pozivanjem `calculate_total_price()` metode.
3. **Dodavanje narudžbine za kupca:** Proverite da li se narudžbina uspešno dodaje u listu svih narudžbina kupca u klasi Customer. Assert proverava da li broj narudžbina raste pri dodavanju.
4. **Pretraga knjige po naslovu u inventaru:** Testirajte pretragu u klasi Inventory da proverite da li je knjiga sa određenim naslovom uspešno pronađena ili ne.

5. **Prikaz istorije narudžbina kupca:** Testirajte metodu `get_order_history()` u klasi `Customer` i proverite da li metoda vraća ispravan broj narudžbina za datog kupca.