Univerzitet u Sarajevu
Elektrotehnički fakultet Sarajevo
Predmet: Razvoj programskih rješenja 2020/2021

Završni ispit - Parcijalni ispit

Ukupno bodova: 20 (Bodovi će se dodijeliti proporcionalno broju uspješnih testova.)

Na repozitoriju se nalazi gotov projekat koji sadrži samo praznu Main klasu i testove. Vaš zadatak je da napravite kompletan Java program koji zadovoljava postavku zadatka i prolazi testove.

Zadatak 1.

Kreirajte klasu **Proizvod** koja opisuje jedan proizvod koji se prodaje u prodavnici. Proizvod je opisan svojim **bar kodom**, **nazivom i cijenom**. Bar kod je oznaka od 13 karaktera od kojih su svi isključivo brojevi. Ukoliko se pokuša postaviti bar kod koji ne odgovara ovoj specifikaciji iz tih metoda treba baciti izuzetak **NeispravanFormatException** sa porukom "Bar kod treba imati 13 cifara". Klasa treba slijediti Java Bean konvenciju.

Zadatak 2.

Kreirajte klasu **Racun** koja modelira fiskalni račun u prodavnici. Račun je opisan svojim **id-**em (Long), **datumom izdavanja (LocalDateTime),** kolekcijom proizvoda (**proizvodi)** koji se nalaze na tom računu sa njihovim količinama te oznakom **zatvoren** koja govori da li je račun zaključen ili je trenutno otvoren. Ukoliko je račun zatvoren, niti jedna metoda ne smije napraviti promjene nad računom. Ukoliko se to pokuša, bacite izuzetak tipa **NedozvoljenaAkcijaException** sa porukom "Račun #574 je već zatvoren". Implementirajte

- Konstruktor sa jednim parametrom (id) koji kreira račun sa poslanim id-em i postavlja datum na trenutni datum te flag zatvoren na false
- Konstruktor sa dva parametra: id i datum izdavanja (ovaj konstruktor je potreban radi testiranja)
- Gettere za sve attribute
- Metodu **dodajProizvod** koja prima dva parametra: proizvod i količinu i dodaje tu količinu tog proizvoda na fiskalni račun. Ukoliko je poslana količina nije veća od 0 baciti izuzetak tipa **IllegalArgumentException** sa porukom "Količina mora biti veća od 0 ".
- Metodu **obrisiProizvod** koja prima ista dva parametra kao i prethodna metoda i briše sa fiskalnog računa toliko jedinica poslanog proizvoda (a ako je poslano više jedinica nego što stoji na računu, obrisati proizvod u potpunosti). Ako proizvoda nema na računu ova metoda ne treba da uradi ništa. Količina također treba da bude veća od 0. Ako nije postupiti kao u prethodnoj metodi.
- Verziju prethodne metode koja prima samo proizvod i bez obzira na količinu proizvoda na računu, briše ga u potpunosti sa računa. Ako proizvoda nema na računu ne treba uraditi ništa.

- Metodu **zatvoriRačun** koja zatvara račun
- Metodu **dajUkupnuCijenu** koja vraća ukupnu cijenu svih proizvoda na računu
- Metodu **toString** koja vraća račun u obliku stringastrignovnom formatu

“Račun #574 (23.02.2021. 11:58):
Kafa, 2 kom, 3.00 KM
Hljeb, 1 kom, 1.00 KM
Ukupno: 4.00 KM”

(Cijena pored prozivoda je cijena za tu količinu proizvoda, ne jedinična cijena tog proizvoda)

Zadatak 3.

Kreirajte klasu **Kasa** koja modelira fiskalnu kasu u prodavnici sa funckionalnostima vođenja evidencije o stanju prodavnice kao i obavljanja kupovine. Kasa je opisana **stanjem** (proizvodi koji se nalaze trenutno u prodavnici uz količinu tih proizvoda na stanju). Konačno kasa vodi evidenciju o **računima** koji su zatvoreni na toj kasi. Implementirajte:

- Konstruktor bez parametara
- Gettere za oba atributa
- Metodu **dodajProizvod** koja prima proizvod i količinu i na stanje dodaje tu količinu tog proizvoda. Količina mora biti veća od nule. Ukoliko nije baciti izuzetak tipa **IllegalArgumentException** sa porukom “Količina mora biti veća od 0”.
- Metodu **kreirajKupovinu** koja otvara jedan račun u prodavnici. Za id postavlja prvi slobodan id računa, a id-evi počinju od 100. Metoda prima mapu proizvoda i njihovih količina koje treba dodati na račun. Drugi argument metode je datum (LocalDateTime) na koji treba postaviti datum izdavanja računa (radi testiranja). Ukoliko se pošalje proizvod koji nije u prodavnici baciti izuzetak tipa **IllegalArgumentException** sa porukom “Proizvod ne postoji u prodavnici” (Proizvodi su jednaki ako su im jednaki bar kodovi). Ukoliko se pokuša dodati više nekog proizvoda nego što je dostupno na stanju, baciti izuzetak istog tipa sa porukom “Na stanju je 10 proizvoda Kafa, a vi pokušavate dodati 15”. Ukoliko se pokuša dodati negativna količina (ili nula) baciti izuzetak istog ripa sa porukom “Količina mora biti veća od 0”. Ova metoda automatski zatvara račun i on se vraća iz metode.
- Metoda **otvoriKupovinu** koja radi na isti način kao prethodna metoda, samo se račun na kraju ne zatvori
- Metoda **dopuniKupovinu** koja prima id računa i mapu proizvoda i količina. Ova metoda će račun s tim id-em dopuniti proizvodima iz drugog parametra. Ukoliko kupovina s tim računom ne postoji treba baciti izuzetak tipa **IllegalArgumentException** sa porukom “Račun #578 nije otvoren”. Količine u ovoj mapi smiju biti negativne čime će se sa računa ukloniti proizvodi. Ukoliko je količina nula ne raditi ništa.

(Sve metode koje dodaju proizvode na račun, naravno mijenjaju stanje u prodavnici. Ukoliko količina nekog proizvoda padne na 0, ne treba ga obrisati iz kolekcije)

- Metodu **zakljuciKupovinu** koja prima id računa i zatvara tu kupovinu ukoliko je otvorena. Ukoliko ista ne postoji ili je već zatvorena ova metoda vraća vrijednost false. U suprotnom vraća vrijednost true.
- Metodu **filtrirajRacune** koja prima funkciju kriterija i vraća listu svih računa koji zadovoljavaju taj kriterij. Ovo treba uraditi preko streamova.
- Medoda **dajRacuneZaDatum** koja prima datum (LocalDate) i vraća listu svih računa koji su zaključeni tog dana. Ovu metodu uraditi pozivanjem prethodne metode.
- Metoda **dajRacunePreko** koja vraća listu svih računa čija je ukupna cijena veća od iznosa koji se šalje kao parametar ovoj metodi . Ovu metodu također treba uraditi pozivanjem metode filtrirajRacune.
- Metodu **dajSortiranePoDatumu** koja vraća skup (Set) svih računa sortiranih po datumu izdavanja od najstarijih prema najnovijim.
- Metodu **presjekDana** koja prima datum (LocalDate) i vraća string koji predstavlja sve račune izdate u tom danu:

“Računi za 23.02.2021:

Račun #574 (23.02.2021. 11:58):
Kafa, 2 kom, 3.00 KM
Hljeb, 1 kom, 1.00 KM
Ukupno: 4.00 KM

Račun #575 (23.02.2021. 12:05):
Mlijeko, 3 kom, 3.50 KM
Hljeb, 1 kom, 1.00 KM
Ukupno: 4.50 KM

Ukupno: 8.50 KM”

Sarajevo, 23. 2. 2021