

OOP – I kolokvijum 2023/2024

Napisati u jeziku Java sledeća stanja i ponašanja:

Paket prodavnica

- Apstraktni tip **IUporediv** koji ima sledeća ponašanja:
 - Javni metod **uporedi** koji prima jedan parametar tipa Object i vraća true/false. Eventualno baca izuzetak **NeuporedivException**.
- Apstraktni tip **Proizvod** koji je **uporediv** ima sledeća stanja i ponašanja:
 - Javnu String promenljivu **naziv**
 - Javnu double promenljivu **cena**
 - Konstruktor koji prima naziv proizvoda i cenu
 - Javnu apstraktnu metodu **dajCenu()**
 - Prepisanu **toString** metodu koja vraća string oblika [naziv - cena]
 - Prepisan metod **uporedi** koji prima koji poredi da li su dva proizvoda jednaka. Proizvodi su jednaki ukoliko imaju isti naziv i cenu. Proizvod se može porediti samo sa proizvodom, u suprotnom se baca izuzetak.
- Tip **Namirnica** koji je **proizvod** i ima sledeća stanja i ponašanja:
 - Nasledjena promenljiva cena predstavlja cenu po komadu
 - Celobrojnu promenljivu **brKomada**
 - Getter i setter za brKomada
 - Konstruktor koji prima naziv i cenu (brKomada je podrazumevano 0)
 - Konstruktor koji prima naziv, cenu i brKomada
 - Konstruktor koji prima Proizvod i brKomada i pravi kopiju objekta Namirnica koji je prosledjen i setuje brKomada
 - Prepisanu metodu **dajCenu()** koja računa ukupnu cenu namirnica
 - Prepisanu **toString** metodu koja vraća string oblika [Namirnica [naziv - cena] – brKomada]
- Tip **Rinfuz** koji je **proizvod** i ima sledeća stanja i ponašanja:
 - Nasledjena promenljiva cena predstavlja cenu po kilogramu
 - Celobrojnu promenljivu **tezina** koja označava težinu u gramima
 - Getter i setter za tezinu
 - Konstruktor koji prima naziv i cenu (tezina je podrazumevano 0)
 - Konstruktor koji prima naziv, cenu i tezinu
 - Konstruktor koji prima Proizvod i tezinu i pravi kopiju objekta Rinfuz koji je prosledjen i setuje tezinu
 - Prepisan metod **dajCenu()** koji računa ukupnu cenu rinfuza
 - Prepisanu **toString** metodu koja vraća string oblika [Rinfuz [naziv - cena] – tezina]
- Tip **AkcijskiProizvod** koji je **namirnica** na akciji
 - Privatnu double promenljivu **akcija**
 - Javni konstruktor koji prima jedan proizvod(proizvod koji se stavlja na akciju) i jednu double promenljivu (popust na proizvod u procentima). Setuje promenljive naziv, cena po komadu i brKomada iz nadklase (cena po komadu je cena sa popustom). Može baciti izuzetak **VecNaAkcijiException**(ukoliko je prosledjen akcijski proizvod) ili **RinfuzNaAkcijiException**(ukoliko je prosledjen rinfuz proizvod).

- Javni konstruktor koji prima **AkcijskiProizvod** i broj komada i pravi kopiju objekta **AkcijskiProizvod** koji je prosledjen i setuje broj komada i akciju
- Javnu metodu **dajCenuBezPopusta()** koja ne prima ništa od parametara i vraća cenu bez popusta
- Javni statički metod **skiniSaAkcije** koji prima jedan akcijski proizvod i vraća Namirnicu kojoj je skinuta akcija
- Prepisanu **toString** metodu koja vraća string oblika [Akcija [naziv - cena]%akcija – brKomada]
- Tip **Prodavnica** ima sledeća stanja i ponašanja:
 - Javni niz proizvoda koji su dostupni u prodavnici
 - Konstruktor koji prima jednu celobrojnu vrednost koja određuje koliko najviše različitih proizvoda može da stane u prodavnicu
 - Javni metod **dodajProizvod** koji prima proizvod i dodaje ga u prodavnicu. Ukoliko nema mesta baca izuzetak **NemaMestaException**
 - Javni metod **dopuniRafove()** koji ne prima nijedan parametar, vec samo dopunjuje količine svih proizvoda na 20 (ukoliko je u pitanju broj komada) ili 2000 (ukoliko je u pitanju težina). Ukoliko je postojeća količina proizvoda veća od količine na koju se dopunjuje, ostaviti već postojeću količinu.
 - Javni metod **dajKupcu** koji prima proizvod koji kupac hoće da uzme i količinu koju kupac želi i vraća odgovarajući objekat proizvoda. Može baciti izuzetak **NemaNaStanjuException** ukoliko nema dovoljno proizvoda. Ukoliko kupac može da dobije proizvod, neophodno je ažurirati stanje proizvoda u prodavnici tj. smanjiti količinu.
 - Javni metod **vратиProizvod** koji prima proizvod i količinu i dodaje ga na stanje u prodavnici ukoliko postoji.
 - Prepisan metod **toString()** koji vraća string reprezentaciju svih proizvoda koje prodavnica nudi

Paket kupci

- Tip **IPopust** koji ima sledeća ponašanja:
 - Promenljivu **popust** koja predstavlja popust u procentima i setovana je na 20
 - Javni metod **dajPopust()** koji vraća double vrednost ostvarenog popusta
- Tip **Kupac** koji ima sledeća stanja i ponašanja:
 - String promenljivu **ime**
 - double promenljivu **budzet**
 - niz proizvoda proizvodi koje korisnik želi da kupi
 - Konstruktor koji prima ime, budzet i maksimalan broj proizvoda koje korisnik može da kupi
 - Javni metod **staviProizvodUKorpu** koji prima prodavnicu u kojoj želi da kupuje, proizvod koji želi da kupi i količinu koju želi da kupi. U niz proizvoda dodati odgovarajući proizvod odgovarajuće težine. Ukoliko u nizu proizvoda postoji isti proizvod ne dodavati novi proizvod nego ažurirati količinu tog proizvoda. Može baciti izuzetak **NemogućeDodatiUKorpuException** ukoliko kupac nema novca da kupi proizvod ili ukoliko nema više mesta i u tom slučaju vraća proizvod u prodavnicu.
 - Javni metod **dajVrednostKorpe()** koji vraća ukupnu cenu korpe kupca
 - Javni metod **kupi** koji ažurira budžet kupca i prazni niz proizvoda. Ukoliko kupac ima pravo na popust, potrebno je uračunati ga prilikom kupovine.
 - Prepisan metod **toString()** koji vraća string oblika Kupac{ime}

- Tip **Student** koji je **kupac** sa **popustom** i ima sledeća stanja i ponašanja:
 - String promenljivu **brIndeksa**
 - Prepisan metod **dajPopust()** koji vraća koliki popust je student ostvario. Popust se ostvaruje na sve proizvode koji nisu na akciji.
 - Prepisan metod **toString()** koji vraća string oblika Kupac{ime} – Student

Paket izuzetci:

- Tip **VecNaAkcijiException** koji je izuzetak
- Tip **RinfuzNaAkcijiException** koji je izuzetak
- Tip **NemaMestaException** koji je izuzetak
- Tip **NeuporedivException** koji je izuzetak
- Tip **NemaNaStanjuException** koji je izuzetak
- Tip **NemogućeDodatiUKorpuException** koji je izuzetak

Default-ni paket:

- Klasu **Test** u kojoj je potrebno
 - Napraviti prodavnicu koja najviše može da ima 6 proizvoda
 - Napraviti 3 rinfuz proizvoda, 2 proizvoda tipa namirnica i jedan akcijski proizvod i dodati u prodavnicu
 - Ispisati sve proizvode koji se nalaze u prodavnici
 - Napraviti 3 kupca od toga jednog studenta
 - Svaki kupac neka kupi 3 različita proizvoda u prodavnici (student i jedan kupac neka kupi određenu količinu proizvoda na akciji)
 - Jedan kupac neka pokuša da kupi proizvod koga nema dovoljno na stanju
 - Dopuniti rafove u prodavnici
 - Student kupuje još jedan proizvod na akciji
 - Kupce staviti u niz
 - Svaki kupac neka kupi proizvode koje je dodao u korpu
 - Za svakog kupca ispisati koliki je njegov račun i popust koji je ostvario (ukoliko je ostvario popust)

Napomena:

Uhvatiti i obraditi sve neophodne izuzetke