

Institut za matematiku i informatiku
OOP – I popravni kolokvijum
25.05.2019.

Napisati na programskom jeziku **JAVA** sledeće tipove, stanja i ponašanja:

Paket putnici

- Tip **Putnik** koji sadži sledeća stanja i ponašanja:
 - *Privatnu* statičku celobrojnu promenljivu *idGen* koja ima podrazumevanu vrednost 0
 - *Privatnu* celobrojnu promenljivu *id*
 - *Privatnu* string promenljivu *ime*
 - *Privatnu* promenljivu *karta* tipa **Karta**
 - *Javni* konstruktor koji prihvata ime i setuje odgovarajuće promenljive
 - *Javne* *gettere* za sve nestatičke promenljive
 - *Javni* *setter* za *kartu*
 - *Javnu* prepisanu metodu **toString()** koja vraća string koji sadrži *ime* i ukoliko putnik ima kartu, vrstu karte i njenu cenu, a ukoliko nema kartu sadrži "Nema kartu!"
- Apstraktni tip **IPopust** kojim se garantuje postojanje metode **dajPopust()** koja vraća realnu vrednost
- Apstraktni tip **Karta** koja ima sledeća stanja i ponašanja:
 - *Privatnu* realnu promenljivu *osnovnaCena*
 - *Javni* konstruktor koji prihvata osnovnu cenu i setuje je
 - *Javni* *getter* za osnovnu cenu
 - *Javnu* apstraktnu metodu **dajCenuKarte()** koja vraća realnu vrednost
- Tip **BiznisKlasa** je *karta* sa sledećim stanjima i ponašanjima:
 - *Privatnu* logičku promenljivu *dodatniWifi*
 - *Privatnu* realnu promenljivu *cenaZaWifi*
 - *Javni* konstruktor koji prima sve promenljive i setuje ih
 - *Javnu* prepisanu metodu **dajCenuKarte()** koja vraća cenu sa uračunatim dodatnim wifi-em ako se koristi
- Tip **EkonomskaKlasa** je *karta* kod koje se **garantuje postojanje popusta** i koja ima sledeća stanja i ponašanja:
 - *Privatnu* realnu promenljivu *tezinaPrtljaga*
 - *Privatnu* realnu promenljivu *cenaPrtljagaPoKg*
 - *Javni* konstruktor koji prima sve promenljive i setuje ih
 - *Javnu* prepisanu metodu **dajPopust()** koja vraća vrednost 20.0 ukoliko je težina prtljaga veća od 10kg, u suprotnom vraća 30.0
 - *Javnu* prepisanu metodu **dajCenuKarte()** koja vraća cenu sa uračunatom težinom prtljaga i popustom

Paket luka.brodovi

- Apstraktni tip **Brod** koji ima sledeća stanja i ponašanja:
 - *Privatnu* string promenljivu *naziv*
 - *Privatnu* string promenljivu *imeKapetanaBroda*
 - *Javni* konstruktor koji prima obe promenljive i setuje ih
 - *Javni* *getter* za naziv broda
 - *Javnu* apstraktnu metodu **dajZaradu()** koja ne prima argumente i vraća realan broj
 - *Prepisanu* metodu **toString()** koja vraća string koji sadrži naziv broda i ime kapetana broda
- Apstraktni tip **IPrevozni** kojim se garantuje postojanje sledećih metoda:
 - **ukrcajPutnika(...)** koja nema povratnu vrednost, a prihvata putnika koga je potrebno ukrcati na brod. Metoda baca izuzetak ukoliko nije moguće ukrcati putnika
 - **iskrcajPutnika(...)** koja nema povratnu vrednost, a prihvata putnika koga je potrebno iskrcati sa broda. Metoda baca izuzetak ukoliko dati putnik nije na brodu
- Tip **PutnickiBrod** je *brod* kojim se **garantuje postojanje popusta i prevoz putnika** i koji ima sledeća stanja i ponašanja:
 - *Privatnu* celobrojnu promenljivu *brojPutnikaBiznisKlase*
 - *Privatnu* celobrojnu promenljivu *brojPutnikaEkonomskeKlase*

- *Privatni* niz **putnik**-a. U ovom nizu na početku se nalaze putnici biznis klase, a nakon njih putnici ekonomske klase
- *Javni* konstruktor koji prima naziv broda, ime kapetana broda, kao i broj putnika biznis klase i ekonomske klase i setuje odgovarajuće promenljive
- *Javnu* prepisanu metodu **dajZaradu()** koja računa ukupnu zaradu koju je putnički brod ostvario
- *Javnu* prepisanu metodu **dajPopust()** koja vraća ukupni popust koji su putnici ostvarili
- *Javnu* prepisanu metodu **ukrcajPutnika(...)** koja ukrcava putnika na brod, vodeći računa o rasporedu putnika na osnovu tipa karata. Greška je ukoliko putnik nema kartu i u tom slučaju baciti izuzetak sa porukom "Putnik <ime putnika> nema kartu!". Takođe, baciti izuzetak ukoliko nema mesta za putnika u određenoj klasi (biznis ili ekonomskoj) sa porukom "Nema dovoljno mesta u <tip karte> za putnika <ime putnika>."
- *Javnu* prepisanu metodu **iskrcajPutnika(...)** koja zadatog putnika izbacuje sa broda. Greška je ukoliko putnik koji se izbacuje ne postoji na brodu i u tom slučaju baciti izuzetak sa porukom "Putnik <ime putnika> ne postoji na brodu!"
- *Javnu* prepisanu metodu **toString()** koja vraća string koji sadrži naziv broda i ime kapetana broda i podatke o svim putnicima na brodu
- Tip **TeretniBrod** je **brod** koji ima sledeća stanja i ponašanja:
 - *Privatnu* celobrojnu promenljivu *brojKontejnera*
 - *Privatnu* celobrojnu promenljivu *zapreminaKontejnera* koja je ista za svaki kontejner na brodu
 - *Privatnu* celobrojnu promenljivu *cenaPoMetruKubnom* koja predstavlja cenu prevoženja kontejnera brodom
 - *Javni* konstruktor koji prima sve promenljive i setuje ih
 - *Javnu* metodu **dajUkupnuZapreminuKontejnera()** koja vraća realan broj koji predstavlja ukupnu zapreminu svi kontejnera na brodu
 - *Javnu* prepisanu metodu **dajZaradu()** koja vraća realnu promenljivu koja predstavlja cenu prevoženja svih kontejnera
 - *Javnu* prepisanu metodu **toString()** koja vraća string koji sadrži naziv broda i ime kapetana broda i podatak o ukupnoj zapremini svih kontejnera na brodu

Paket luka

- Tip **Skladiste** koji ima sledeća stanja i ponašanja:
 - *Privatnu* realnu promenljivu *kapacitet*
 - *Privatnu* realnu promenljivu *popunjenost* čija je podrazumevana vrednost 0.0
 - *Privatnu* celobrojnu promenljivu *brojBrodova* čija je podrazumevana vrednosti 0. Broj brodova predstavlja informaciju o tome koliko brodova je tu skladištilo sve svoje kontejnere
 - *Javni* konstruktor koji prihvata kapacitet i setuje ga
 - *Javne* *gettere* za sve promenljive
 - *Javne* *settere* za popunjenost i broj brodova
 - *Javnu* prepisanu metodu **toString()** koja vraća string koji sadrži informacije o sve tri promenljive
- Tip **Luka** koji ima sledeća stanja i ponašanja:
 - *Privatni* niz brodova koji se nalaze u luci
 - *Privatni* niz skladišta u koja se smeštaju kontejneri sa teretnih brodova
 - *Javni* konstruktor koji prihvata broj brodova koji mogu da stanu u luku i niz skladišta i setuje odgovarajuće promenljive
 - *Javnu* metodu **prodajKartu(...)** koja nema povratnu vrednost, a prihvata putnika i kartu koju daje datom putniku
 - *Javnu* metodu **usidriBrod(...)** koja nema povratnu vrednosti, a prihvata brod koji se smešta u luku. U slučaju da se radi o teretnom brodu, dodatno **izbacuje kontejnere u skladišta**. Greška je ukoliko nema mesta da brod pristane u luku. U tom slučaju baciti izuzetak sa porukom "Nema mesta u luci!"
 - *Privatnu* metodu **izbaciKontejnereUSkladista(...)** koja prihvata teretni brod i pronalazi skladište u koje može da izbaci **sve** svoje kontejnere. Ako je moguće izbaciti sve kontejnere u neko skladište, ažurirati odgovarajuće informacije o skladištu. Greška je ukoliko ne postoji takvo skladište. U tom slučaju baciti izuzetak sa porukom "Nema mesta u skladištima!"
 - *Javnu* metodu **izlistajUsidreneBrodove()** koja nema povratnu vrednosti i štampa na ekran podatke o svim brodovima u luci

- Javnu metodu **izlistajSkladista()** koja nema povratnu vrednosti i štampa na ekran podatke o svim skladištima

default paket

U tesnoj klasi je potrebno kreirati sledeće statičke metode:

- **ispisiPodatkeOPutnicima(...)** koja prihvata niz putnika i ispisuje podatke o svim putnicima. Metoda nema povratnu vrednost
- **ukrcajPutnikeNaBrodove(...)** koja prihvata niz putnika i niz brodova i date putnike redom ukrcava na brodove kojim putnici mogu da se prevoze. Metoda nema povratnu vrednost
- **usidriSveBrodove(...)** koja prihvata luku i niz brodova i usidrava sve brodove u luku
- **iskrcajPutnikeIzBroda(...)** koja prihvata niz putnika i **prevozni** brod i pokušava da iskrca svakog drugog putnika iz niza putnika sa datog broda
- **dajUkupanPopustZaSveBrodove(...)** koja prihvata niz brodova i vraća realan broj koji predstavlja ukupan **popust** na svim brodovima

U fajlu **Test.java** se nalazi **main** funkcija koju je potrebno pozvati. Potrebno je dopuniti main funkciju kako bi ona bila ispravna.