

K3

Java i Python

Napisati klasu **Roba**, koja sadrži polja **marka** (String), **naziv** (String) i **tezinaKG** (ceo broj). U klasi **Roba** realizovati konstruktor sa parametrima sa podrazumevanim vrednostima i redefinisati metodu za ispis.

Napisati klasu **Vagon**, koja sadrži polja **opisTereta** (String) i **teret** (lista objekata klase **Roba**). U klasi **Vagon** realizovati konstruktor bez parametara, metod **getTezinaRobe** koja vraća težinu sve robe koja se nalazi unutar vagona (njihovu sumu), metodu **dodajRobu** koja dodaje prosleđeni objekat klase **Roba** na kraj liste objekata (u polju **teret**), kao i redefinisati metodu za ispis.

Prilikom kreiranja postaviti vrednost polja **opisTereta** na „PRAZAN“, a prilikom dodavanja nove robe vrednost ovog polja može imati jedno od dve vrednosti: „RAZNO“ – ukoliko se unutar vagona nalazi makar dve robe različitih marki; „<MARKA ROBE>“ – ukoliko je sva roba unutar vagona iste marke, onda je vrednost ovog polja naziv te marke.

Primeri:

1) *teret prazan* > *opisTereta* = „PRAZAN“

2) *teret (“Nike”, “AirMax patike”, 50)* > *opisTereta* = “Nike“

3) *teret (“Nike”, “AirMax patike”, 50), (“Nike”, “Lopta”, 14)* > *opisTereta* = “Nike“

4) *teret (“Nike”, “AirMax patike”, 50), (“Nike”, “Lopta”, 14), (“Loto”, “Majica”, 6)* > *opisTereta* = “RAZNO“

Napisati klasu **TeretniVoz**, koja sadrži polja **vagoni** (niz objekata klase **Vagon**) i **vagonMaxKG** (ceo broj) koji govori koliko tereta maksimalno može da stane unutar svakog od vagona. U klasi **TeretniVoz** realizovati:

- konstruktor sa parametrima: **brVagona** (ceo broj) koji govori koliko vagona se nalazi unutar teretnog voza i **maxDozvoljenoKG** (ceo broj) koji govori koliko tereta maksimalno može da stane unutar svakog od vagona.
- **getSlobodanProstor** koja za prosleđeni broj vagona vraća koliko je ostalo slobodnog prostora u kilogramima (ukoliko je prosleđeni broj vagona nepostojeći tj. van opsega metoda vraća -1),
- **utovariTeret** koja dodaje prosleđeni objekat klase **Roba** u prosleđeni broj vagona (metod vraća da li je došlo do uspešnog utovara; ukoliko je prosleđeni broj vagona nepostojeći tj. van opsega metoda vraća false; ukoliko u traženom vagonu nema dovoljno mesta za Robu koja se pokušava dodati metoda takođe vraća false),
- **getMarkaNajtezeRobe** koja vraća marku najteže robe unutar teretnog voza (ukoliko je voz prazan, metoda vraća „PRAZAN“; ukoliko se desi da nekoliko roba je iste maksimalne težine, vratiti marku bilo koje od njih),
- redefinisati metodu za ispis.

NAPOMENE

- Za **Java** rešenje: sva polja možete implementirati kao private ili protected (ne public) u zavisnosti od vašeg izbora načina implementacije konstruktora i metoda u zadatku.
- Sve što nije testirano u **main()** funkciji ne donosi bodove.

- Program se mora uspešno kompajlirati, u suprotnom možete dobiti najviše jedan bod.