

U programskom jeziku JAVA potrebno je kreirati sledeće tipove:

Paket Prevoz

Apstraktni tip **PrevoznoSredstvo** koje ima sledeća stanja i ponašanja:

- Privatnu *double* promenljivu **brzinaPrevoza**
- Niz **karte** tipa **Karta**
- Javni konstruktor koji prihvata brzinu i setuje je
- Javnu apstraktnu metodu **cekirajKartu** koja nema povratni tip i prihvata tip **Karta** kao ulazni parametar
- Javnu apstraktnu metodu **dajKartu** koja ima povratni tip **Karta**, a kao ulazne parametre prihvata **double** promenljivu **razdaljina** (razdaljina za koju se karta izdaje) i tip **Putnik** (putnik za koga se izdaje karta)
- Javni metod **vremePutovanja** koji ima povratni tip **double** i za prosleđeni parametar **double razdaljina** računa vreme putovanja prevoznim sredstvom
- Javni apstraktni metod **dajBrojSlobodnihMesta** koji ima povratni tip **int**
- Javni getter za niz karata

Avion koji je **PrevoznoSredstvo** i ima sledeća stanja i ponašanja:

- Privatnu *celobrojnu* promenljivu **brojSlobodnihMestaUBiznisKlasi**
- Privatnu *celobrojnu* promenljivu **brojSlobodnihMestaUEkonomskojKlasi**
- Staticku *double* promenljivu **cenaKartePoKilometruUBiznisKlasi** (setovati vrednost na 20)
- Staticku *double* **cenaKartePoKilometruUEkonomskoj** (setovati vrednost na 10)
- Staticku *double* **cenaPrtljagaPoKilogramu** (setovati vrednost na 50)
- Javni konstruktor koji prihvata promenljive **int brojMestaUBiznisKlasi**, **int brojMestaUEkonomskojKlasi**, **double brzinaPrevoza** i setuje ih
- Implementaciju metode **dajKartu** iz nasleđene klase. U zavisnosti od budžeta putnika prvo pokušati kreiranje karte **BiznisKlasa**, ukoliko putnik nema dovoljno novca, pokušati sa kreiranjem karte **EkonomskaKlasa**. Ukoliko putnik nema novca ni za ekonomsku klasu vratiti **null**.
- Implementaciju metoda **dajBrojSlobodnihMesta** iz nasleđene klase
- Implementaciju metoda **cekirajKartu** iz nasleđene klase. Referencu prosleđene karte je potrebno sačuvati u niz karata, a zatim umanjiti odgovarajući broj slobodnih mesta

Autobus koji je **PrevoznoSredstvo** i ima sledeća stanja i ponašanja:

- Staticku *double* promenljivu **cenaPoKilometru** (setovati vrednost na 5)
- Staticku *double* **cenaPoKilogramuPrtljaga** (setovati vrednost na 2)
- Javni konstruktor koji prihvata promenljive **int brojSlobodnihMesta**, **int brzinaPrevoza** i setuje ih
- Implementaciju metode **dajKartu** iz nasleđene klase. Potrebno je kreirati kartu tipa **EkonomskaKlasa**. Ukoliko putnik nema novca vratiti **null**.
- Implementaciju metoda **dajBrojSlobodnihMesta** iz nasleđene klase
- Implementaciju metoda **cekirajKartu** iz nasleđene klase. Referencu prosleđene karte je potrebno sačuvati u niz karata, a zatim umanjiti odgovarajući broj slobodnih mesta

Paket Karte

Apstraktni tip **Karta** koji ima sledeća stanja i ponašanja:

- Promenljivu *putnik* tipa **Putnik**
- Promenljivu *double cenaPoKilometru*
- Promenljivu *double cenaPoKilogramuPrtljaga*
- Promenljivu *double razdaljina*
- Javni konstruktor koji prihvata prethodno opisane promenljive i setuje ih
- Javnu apstraktnu metodu **dajCenuKarte**
- Pregaženi metod *toString* koji vraća ime putnika koji je kupio kartu i tip karte (Biznis/Ekonomska)

Tip **EkonomskaKlasa** koji je **Karta** i ima sledeća stanja i ponašanja:

- Javni konstruktor koji prihvata vrednosti za sve promenljive iz nadklase i setuje ih
- Implementaciju metoda **dajCenuKarte** koja računa cenu karte na osnovu razdaljine, težine prtljaga i njihovih cena. Ukoliko putnik ima popust on se uračunava u cenu koja se vraća

Tip **BiznisKlasa** koji je **Karta** i ima sledeća stanja i ponašanja:

- Javni konstruktor koji prihvata vrednosti za sve promenljive iz nadklase i setuje ih
- Implementaciju metoda **dajCenuKarte** koja računa cenu karte na osnovu razdaljine, težine prtljaga i njihovih cena.

Paket Putnici

Apstraktni tip **Ipopust** koji ima sledeću metode:

- *double dajpopust()*

Tip **Putnik** koji ima sledeća stanja i ponašanja:

- *String* promenljivu **ime**
- *Double* promenljivu **budzet**
- *Double* promenljivu **tezinaPrtljaga**
- Promenljivu **karta** tipa **Karta** (setovati je na null)
- Javni konstruktor koji prihvata sledeće promenljive *String ime*, *double budzet*, *double tezinaPrtljaga*
- Gettere za tezinu, budzet i ime
- Setter za kartu

Tip **Radnik** koji je **Putnik** i ima sledeća stanja i ponašanja:

- Promenljivu **nazivZanimanja** tipa *String*
- Javni konstruktor koji prihvata sledeće promenljive *String ime*, *String nazivZanimanja*, *double budzet*, *double tezinaPrtljaga* i setuje ih

Tip **Student** koji je **Putnik** sa **POPUSTOM** i ima sledeća stanja i ponašanja:

- Staticku promenljivu *double* **popust** koja treba setovati na vrednost 10
- Konstruktor koji prihvata *String ime*, *String brojIndeksa*, *double budzet*, *double tezinaPrtljaga*

Paket Izuzeci

Tip **NedovoljnoNovca** koji je **Izuzetak** i ima sledeće osobine:

- Promenljivu **poruka** tipa *String*
- Javni konstruktor koji setuje pomenutu promenljivu
- Javni metod **getPoruka** koji ima povratni tip *String* i vraća **poruku**

Tip **NepostojecaDestinacija** koji je **Izuzetak** i ima sledeće osobine:

- Promenljivu **poruka** tipa *String*
- Javni konstruktor koji setuje pomenutu promenljivu
- Javni metod **getPoruka** koji ima povratni tip *String* i vraća **poruku**

Default paket

Tip **Destinacija** koji ima sledeća stanja i ponašanja:

- Privatni niz **prevoznaSredstva** tipa *PrevoznaSredstvo*
- Privatnu *String* promenljivu **odMesta**
- Privatnu *String* promenljivu **doMesta**
- Privatnu *double* promenljivu **razdaljina**
- Javni konstruktor koji prihvata *String odMesta, String doMesta, double razdaljina, int maxBrojPrevoznihSredstava*
- Javni metod **dodajPrevoznaSredstvo** koji prihvata referencu na **PrevoznaSredstvo** i dodaje je u niz
- Javni metod **equals** koji ima povratni tip **boolean** i prihvata promenljive *String odMesta, String doMesta*. Ukoliko se prosleđene vrednosti poklapaju sa vrednostima u destinaciji, metod vraća **true**, u suprotnom vraća **false**.
- Javni metod **kupiKartu** koji ima povratni tip **Karta** i prihvata referencu na **putnika** za kojeg se karta kupuje. Za budzet koji putnik ima potrebno je kupiti kartu kojom će najbrže stići na željenu destinaciju nekim od prevoznih sredstava koja su raspoloživa za istu. Ukoliko putnik ima novca potrebno je čekirati kartu kod prevoznog sredstva za koji je karta kupljena, u suprotnom podići izuzetak **NedovoljnoNovca**.

Tip **TuristickaAgencija** koji ima sledeća stanja i ponašanja:

- Privatni niz destinacije tipa *Destinacije*
- Privatnu *String* promenljivu **naziv**
- Javni konstruktor koji setuje članice klase
- Javni metod **dodajDestinaciju** koji prihvata referencu na *Destinaciju* i setuje je u **niz destinacija**
- Javni metod **kupiKartu** koji ima povratni tip **Karta** i prihvata sledeće promenljive *String odMesta, String doMesta, Putnik putnik*. Ukoliko prosleđena destinacija ne postoji, podiže se izuzetak **NepostojecaDestinacija**. Ukoliko destinacija postoji, kupiti kartu za putnika. Ako je kupovina prošla vratiti referencu na kartu.

Opis Testne klase:

U nastavku su prikazane klase koje je potrebno instancirati i potrebno je dodati sledeće stvari:

1. Za svakog putnika iz niza `putniciZaPariz` potrebno je kupiti kartu za destinaciju Kragujevac-Pariz. Ukoliko je karta uspešno kupljena setovati je kod putnika
2. Za svakog putnika iz niza `putniciZaLondon` potrebno je kupiti kartu za destinaciju Kragujevac-London. Ukoliko je karta uspešno kupljena setovati je kod putnika
3. Ispisati na standardni izlaz imena putnika koji su kupili avionsku biznis klasu
4. Ispisati na standardni izlaz koliko su putnici ukupno potrošili novca na kupovinu karata

```
TuristickaAgencija turistickaAgencija = new TuristickaAgencija("IMI Travel", 2);
```

```
Destinacija d1 = new Destinacija("Kragujevac", "Pariz", 1900, 5);
```

```
PrevoznoSredstvo a1 = new Avion(1,5, 900);
```

```
d1.dodajPrevoznoSredstvo(a1);
```

```
PrevoznoSredstvo v1 = new Autobus(5,80);
```

```
d1.dodajPrevoznoSredstvo(v1);
```

```
turistickaAgencija.dodajDestinaciju(d1);
```

```
Destinacija d2 = new Destinacija("Kragujevac", "London", 2225, 10);
```

```
PrevoznoSredstvo a2 = new Avion(1,5, 900);
```

```
d2.dodajPrevoznoSredstvo(a2);
```

```
PrevoznoSredstvo v2 = new Autobus(4,80);
```

```
d2.dodajPrevoznoSredstvo(v2);
```

```
turistickaAgencija.dodajDestinaciju(d2);
```

```
Putnik putniciZaPariz[] = new Putnik[5];
```

```
putniciZaPariz[0] = new Student("Maja", "54/08", 60000, 10);
```

```
putniciZaPariz[1] = new Radnik("Mila", "Informaticar", 20000, 15);
```

```
putniciZaPariz[2] = new Radnik("Ana", "Informaticar", 15000, 20);
```

```
putniciZaPariz[3] = new Student("Milica", "60/08", 11000, 20);
```

```
putniciZaPariz[4] = new Student("Marina", "59/08", 14000, 10);
```

```
//TO DO 1
```

```
// Za svakog putnika iz niza "putniciZaPariz" pokusaj da kupis kartu od  
turisticke agencije za destinacija "Kragujevac", "Pariz"
```

```
Putnik putniciZaLondon[] = new Putnik[5];
```

```
putniciZaLondon[0] = new Student("Pera", "590/08", 54000, 10);
```

```
putniciZaLondon[1] = new Radnik("Zika", "Informaticar", 20000, 15);
```

```
putniciZaLondon[2] = new Radnik("Zika", "Informaticar", 20000, 20);
```

```
putniciZaLondon[3] = new Student("Pavle", "60/09", 20000, 20);
```

```
putniciZaLondon[4] = new Student("Milivoje", "59/07", 20000, 10);
```

```
//TO DO 2 - Za svakog putnika iz niza "putniciZaLondon" pokusaj da kupis  
kartu od turisticke agencije za destinacija "Kragujevac", "London"
```

```
//TO DO 3 - Ispisati na standardni izlaz imena putnika koji su kupili  
avionsku biznis klasu
```

```
//TO DO 4 - Ispisati na standardni izlaz koliko su putnici ukupno potrosili  
novca na kupovinu karata
```