

Napraviti konzolnu C# aplikaciju i snimiti je u folderu **/Rad/ImePrezimeBrIndeksa_kolokvijum/**.
Aplikacija treba da sadrži sledeće:

Javni interfejs **IPopust** koji sadrži javni properti **Popust** kojim se može dohvatiti **double** vrednost.

Javni apstraktni tip **Karta** koji sadrži:

- Zaštićenu string promenljivu **id**.
- Zaštićenu string promenljivu **pocetnaDestinacija**.
- Zaštićenu string promenljivu **krajnjaDestinacija**.
- Zaštićenu double promenljivu **razdaljinaUKilometrima**.
- Zaštićenu double promenljivu **cenaKartePoKilometru**.
- Javni konstruktor koji prihvata vrednosti za sve prethodno navedene promenljive i setuje ih.
- Javni properti **PocetnaDestinacija** kojim se može dohvatiti vrednost promenljive **pocetnaDestinacija**.
- Javni properti **KrajnjaDestinacija** kojim se može dohvatiti vrednost promenljive **krajnjaDestinacija**.
- Apstraktni metod **dajCenuKarte**.
- Prepisan metod string reprezentacije koji vraća string u formatu „**id – pocetnaDestinacija -> krajnjaDestinacija**“.

Javni tip **AutobuskaKarta** koji je **Karta** i sadrži:

- Javni konstruktor koji prihvata sve vrednosti za kartu i setuje ih.
- Cena karte se računa tako što se razdaljina u kilometrima pomnoži cenom karte po kilometru.

Javni tip **VoznaKarta** koji je **Karta** i obezbediti da **VoznaKarta** ima popust.

- Cena karte se računa tako što se razdaljina u kilometrima pomnoži cenom karte po kilometru.
- Popust se računa tako što se za svaki kilometar preko 100 kilometra cena umanjuje za 10%.

Javni tip **AvionskaKarta** koji je **Karta** i obezbediti da **AvionskaKarta** ima popust. **AvionskaKarta** sadrži sledeće:

- Zaštićenu logičku promenljivu **prtljag**, koja služi kao indikator da li putnik ima prtljag sa sobom.
- Zaštićenu logičku promenljivu **lastMinute**.
- Cena karte se računa tako što se razdaljina u kilometrima pomnoži cenom karte po kilometru. Ukoliko putnik poseduje prtljag potrebno je dodati 100,0 na cenu.
- Ukoliko je ponuda **lastMinute** putnik ima popust od 50% na cenu karte, u suprotnom plaća pun iznos.
- Javni konstruktor koji prihvata sve podatke za **AvionskuKartu** i setuje ih.

Javni tip **GreskaAranzmana** koji je **Exception** i sadrži:

- Zaštićenu promenljivu **karta** tipa **Karta**.
- Javni konstruktor koji prihvata kartu i string promenljivu koja predstavlja poruku koju je potrebno dodeliti svojstvu **Message** klase **Exception**.
- Prepisan metod string reprezentacije koji vraća string u formatu „**Tip karte - poruka**“. Tip karte može imati vrednosti: avionska karta, autobuska karta ili vozna karta, u zavisnosti od tipa karte.

Javni tip **Aranzman** koji ima popust. Tip **Aranzman** sadrži sledeće:

- Privatnu string promenljivu **pocetnaDestinacija**.
- Privatnu string promenljivu **krajnjaDestinacija**.
- Privatnu logičku promenljivu **lastMinute**.
- Privatnu celobrojnu promenljivu **brojNoćenja**, koja označava koliko noći će putnik prespavati u hotelu koji je deo aranžmana.
- Privatnu **double** promenljivu **cenaNoćenja**.
- Zaštićenu listu tipa **Karta** koja predstavlja karte kojima putnik stiže od početne do krajnje destinacije aranžmana.
- Javni svojstvo **PocetnaDestinacija** kojim se može dohvatiti vrednost za **pocetnaDestinacija**.
- Javni svojstvo **KrajnjaDestinacija** kojim se može dohvatiti vrednost za **krajnjaDestinaciju**.
- Javni konstruktor koji prihvata sve vrednosti koje poseduje aranžman i setuje ih.
- Javni metod **dodajKarte** koji prihvata listu karata. Ukoliko se početna destinacija aranžmana ne poklapa sa početnom destinacijom prve karte u listi, baciti izuzetak **GreskaAranzmana** za problematičnu kartu sa odgovarajućom porukom koja opisuje nastali problem. Ukoliko se krajnja destinacija poslednje karte u listi ne poklapa sa krajnjom destinacijom aranžmana baciti izuzetak **GreskaAranzmana** za problematičnu kartu sa odgovarajućom porukom koja opisuje nastali problem. Ako se početne i krajnje destinacije poklapaju setovati listu karata.
- Javni svojstvo **CenaAranzmana** kojim se može dohvatiti cena aranžmana. Cena se računa tako što se broj noćenja u hotelu pomnoži sa cenom jednog noćenja i na sve to dodaju cene svih karata aranžmana.
- Ukoliko je ponuda **lastMinute** onda putnik ima **popust** od 30% na cenu noćenja u hotelu. U popust je potrebno uračunati i popuste svih karata.
- Prepisan metod string reprezentacije koji vraća string u formatu
 „**Aranžman: pocetna destinacija – krajnja destinacija**
id prve karte – pocetna destinacija prve karte -> krajnja destinacija prve karte
id druge karte - pocetna destinacija druge karte -> krajnja destinacija druge karte
 ...“

Javni tip **GreskaPutovanja** koji je **Exception** i sadrži:

- Zaštićenu promenljivu aranžman tipa **Aranzman**.
- Javni konstruktor koji prihvata aranžman i string promenljivu koja predstavlja poruku koju je potrebno dodeliti svojstvu **Message** klase **Exception**.

- Prepisan metod string reprezentacije koji vraća string u formatu „**string reprezentacija aranžmana – poruka greške**“

Javni tip **Putovanje** koji ima popust. Tip **Putovanje** sadrži:

- Privatnu string promenljivu **pocetnaDestinacija**.
- Privatnu string promenljivu **krajnjaDestinacija**.
- Zaštićenu listu tipa **Aranžman** koja predstavlja sve aranžmane u okviru jednog putovanja.
- Javni metod **dodajAranzmane** koji prihvata listu aranzmana. Ukoliko se početna destinacija putovanja ne poklapa sa prvim aranžmanom u listi baciti izuzetak **GreskaPutovanja** sa odgovarajućom porukom. Ukoliko se krajnja destinacija putovanja ne poklapa sa krajnjom destinacijom poslednjeg aranžmana u listi baciti izuzetak **GreskaPutovanja** sa odgovarajućom porukom. Ako se destinacije poklapaju setovati listu aranžmana.
- Javni properti **CenaPutovanja** kojim se može dohvatiti cena putovanja. Cena putovanja predstavlja sumu cena svih karata i nocenja u hotelima.
- Javni properti **Popust** koji predstavlja sumu svih popusta na karte i aranžmane.
- Prepisan metod string reprezentacije koji vraća string u formatu:

“

Putovanje : početna destinacija -> krajnja destinacija

Prvi aranaranžman: početna destinacija -> krajnja destinacija

id prve karte – pocetna destinacija prve karte -> krajnja destinacija prve karte

id druge karte - pocetna destinacija druge karte -> krajnja destinacija druge karte

...

Drugi aranaranžman: početna destinacija -> krajnja destinacija

id prve karte – pocetna destinacija prve karte -> krajnja destinacija prve karte

id druge karte - pocetna destinacija druge karte -> krajnja destinacija druge karte

...

...

”

U **Main** metodu potrebno je kreirati putovanje, aranžmane i karte kao u kodu ispod. Karte1 dodati u a1, karte 2 u a2 i karte3 u a3. Sve aranžmane dodati u kreirano putovanje. Voditi računa o hvatanju izizetaka prilikom dodavanja karata i aranžmana. Izuzetake je potrebno uhvatiti ispisati njihovu string reprezentaciju. Na standardnom izlazu je potrebno ispisati ukupnu cenu svih aranžmana (noćenja i karata) koji su deo putovanja, ukupan popust svih aranžmana (noćenja i karata) i cenu putovanja sa uračunatim popustom.

```
Putovanje p = new Putovanje("Kragujevac", "Prag");
List<Aranzman> aranzmani = new List<Aranzman>();
```

```

Aranzman a1 = new Aranzman("Kragujevac", "Budimpesta", 3, 20, false);
List<Karta> karte1 = new List<Karta>();
karte1.Add(new AutobuskaKarta("AUTOBUS_101", "Kragujevac", "Beograd", 150, 0.2));
karte1.Add(new AvionskaKarta("AVION_102", "Beograd", "Budimpesta", 320, 1, true, true));

Aranzman a2 = new Aranzman("Budimpesta", "Prag", 5, 20, true);
List<Karta> karte2 = new List<Karta>();
karte2.Add(new AvionskaKarta("AVION_201", "Budimpesta", "Prag", 442, 0.05, false, true));

Aranzman a3 = new Aranzman("Prag", "Kragujevac", 0, 0, true);
List<Karta> karte3 = new List<Karta>();
karte3.Add(new AvionskaKarta("AVION_301", "Prag", "Budimpesta", 442, 0.1, false, true));
karte3.Add(new VoznaKarta("Voz_301", "Budimpesta", "Kragujevac", 470, 0.15));

```

