

OP K2

C++

Modelirati prevoz žičarom na nekoj planini. Žičara radi bez prekida i prevozi po nekoliko putnika odjednom u pravilnim razmacima od po nekoliko minuta. Koliko putnika se prevozi žičarom u jednoj turi zavisi od tipa žičare (dvosed, četvorosed, šestosed,...). Žičara se koristi u različitim periodima godine i svaki period može imati drugačiju vrstu karte.

Napisati klasu **Karta** koja ima sledeće polja:

- **brojKarte** (tipa **int**)
- **vrsta** (Nabrojivi tip podataka **VrstaKarte** ima sledeće vrednosti: **LETO**, **SKI**, **SKI_OPENING**)

Za klasu Karta realizovati sledeće metode:

- konstruktor bez parametara
- konstruktor sa parametrima
Karta(int, VrstaKarte);
prvi parametar za broj karte, drugi parametar za vrstu karte
- getere za polja brojKarte i vrsta
- preklopiti operator ispisa <<, tako da ispisuje sve informacije o objektu. Polja nabrojivog tipa ispisati rečima.

Napisati klasu **Putnik** ima polja:

- **Karta** (tipa **Karta**)
- **mb** (tipa **DinString**)
- **ime** (tipa **DinString**)

Za klasu Putnik realizovati sledeće metode:

- konstruktor bez parametara
- konstruktor sa parametrima
Putnik(int, VrstaKarte, const DinString&, const DinString&);
prvi parametar za broj karte, drugi za vrstu karte, treći za matični broj, četvrti za ime
- metod **getMb**
- metod **getVrstaKarte**
- virtualni metod **DinString toString const();** - kao povratnu vrednost vraća DinString koji nastaje spajanjem informacija o objektu (mb i ime) koji su razdvojeni razmakom.
- preklopiti operator ispisa << - ispisuje rezultat metode toString i ispisuje kartu

Iz klase **Putnik** izvesti klasu **PutnikDete** koja ima:

- dodatno polje **mbRoditelja** (tipa **DinString**)

Za klasu PutnikDete realizovati sledeće metode:

- konstruktor bez parametara
- konstruktor sa parametrima
PutnikDete(int, VrstaKart, const DinString&, const DinString&, const DinString&);

gde je prvi parametar broj karte, drugi vrsta karte, treći matični broj, četvrti ime i peti matični broj roditelja.

- konstruktor sa parametrima

PutnikDete(const Putnik&, const DinString&);

gde je prvi parametar objekat klase Putnik koji predstavlja roditeljski deo klase, a drugi parametar matični broj roditelja.

- redefinisani metod **toString** - tako da vraća DinString koji dodaje reč "Dete: ", a zatim informacije o imenu, matičnom broju i matičnom broju roditelja.

Napisati generičku klasu **Zicara<VrstaKarte SEZONA, int BROJ_SEDISTA, int INTERVAL>** koja je određena sezonom (nabrojivi tip vrsta karte), brojem sedišta koji se odnosi na vrstu žičare (koliko putnika se odjednom može prevesti) i intervalom izraženim u minutima koji se odnosi na vremenski period između puštanja dve ture putnika.

Klasa ima:

- **Putnici** (tipa **List<Putnik*>**)
- **stanje** (nabrojivi tip podataka **StanjeZicare** sa mogućim vrednostima: **OTVORENA, ZATVORENA**)
- **ukupnoVoznji** - (tipa **int**) broji ukupan broj voznji od otvaranja (pogledaj metod **preveziTuru**)
- **ukupnoMinuta** (tipa **int**) broji koliko dugo zicara radi u minutima od otvaranja (pogledaj metod **preveziTuru**)
- konstruktor bez parametara - postavlja stanje na **ZATVORENA**
- metod **void otvori();** - postavlja stanje na **OTVORENA** i ukupan broj voznji i minuta na 0
- metod **void zatvori();** - koji postavlja stanje na **ZATVORENA** (ukupan broj voznji i minuta se ne menja) i prazni listu putnika koji čekaju u redu za prevoz.
- metod **bool dodajPutnika(Putnik*);** - dodaje zadatog putnika na kraj reda ukoliko taj putnik već nije u listi i ukoliko ima ispravnu kartu (vrsta karte se ne poklapa sa sezonom). Da li se putnik nalazi u listi proveriti po matičnom broju. U slučaju da karta nije ispravna ili da se putnik već nalazio u listi metoda vraća false, dok je u suprotnom povratna vrednost informacija o uspešnosti dodavanja putnika u listu.
- metod **int brojPutnika() const;** - vraća koliko putnika čeka u redu.
- metod **bool preveziTuru()** - U slučaju da je žičara u stanju ZATVORENA ili da nema putnika u redu metoda vraća false. Ako je žičara OTVORENA prevozi prve putnike na redu. Ovom prilikom se prevozi onoliko putnika odjednom koliko žičara ima sedišta. Prilikom prevoženja putnici se uklanjaju iz reda čekanja. Ukoliko u redu ima manje putnika nego što je potrebno da se popuni jedna tura, prevozi se manje putnika. Polje **ukupnoVoznji** se uvećava za broj prevezenih putnika. Takođe, ukoliko je bilo prevoza, polje **ukupnoMinuta** se uvećava za zadati interval. U tom slučaju metoda vraća true.
- preklopiti operator ispisa **<<** - tako da ispisuje sezonu, broj sedišta na žičari, trajanje jedne vožnje, stanje, ukupan broj voznji, ukupno radno vreme žicare u satima, broj putnika u redu i u slučaju da je broj putnika u redu veći od nule, ispisati listu svih putnika koji čekaju u redu.

NAPOMENE

- Za listu putnika koristiti klasu List koje je data i koju treba uključiti u projekat.
- Za nabrojive tipove podataka se, takođe, može realizovati operator ispisa, ali nije obavezan.
- Pretpostaviti da je unos validan odnosno zanemariti slučajeve kada je interval ili broj sedišta negativna vrednost i slično.
- DinString klasa ima preklopljen operator + što znači da se stringovi mogu sabirati
- DinString klasa ima preklopljen operator == što znači da se stringovi mogu upoređivati
- Sve što nije testirano u **main()** funkciji (implicitno ili eksplicitno) ne donosi bodove.
- Polja nabrojivog tipa ispisivati rečima.
- Program se mora uspešno kompajlirati, u suprotnom možete dobiti najviše jedan bod.