

## PRVI ZADATAK - 1. grupa (20 bodova)

Napravite generičku strukturu (koristiti template) `Vozilo` koja opisuje kretanje vozila po cjelobrojnoj kvadratnoj mreži. Vozilo se na početku nalazi na koordinatama (0, 0). Struktura `Vozilo` ima sljedeće funkcije članice:

- konstruktor koji prima brzinu vozila tipa  $T$  (neki numerički tip, npr. `int` ili `float`)
- funkciju `promijeniBrzinu` koja mijenja brzinu vozila. Funkcija prima jedan parametar tipa  $T$ .
- funkciju `pomak` koja prima `string`. Funkcija pomiče vozilo za jedno mjesto lijevo, desno, gore ili dolje ovisno primi li `string` "lijevo", "desno", "gore" ili "dolje". Svaki pomak vozila je jednak putu od 1 km. Možete pretpostaviti da će postojati najviše 10 pomaka za jedno vozilo.
- funkciju `ispisiNaEkran` koja ispisuje koordinate i rutu vozila. Ako su redom proslijeđeni pomaci `gore gore desno lijevo lijevo dolje gore` tada se ispisuje:  
(-2, 2)  
2 gore 1 desno 3 lijevo 1 dolje 1 gore
- funkciju `prosjecnaBrzina` koja vraća prosječnu brzinu vozila (ukupno pređeni put / ukupno vrijeme). **Napomena:** ako se vozilo pomakne jedno mjesto gore, a zatim jedno mjesto dolje to se računa kao pređeni put od 2 km iako je ukupni pomak 0 km.
- funkcija `izbjegniSudar` koja prima neko drugo vozilo  $V$ . Ako se vozilo koje je pozvalo funkciju i vozilo  $V$  nalaze na istom mjestu tada se vozilo koje je pozvalo funkciju pomiče jedno mjesto prema gore, a vozilo  $V$  jedno mjesto prema desno. Funkcija u tom slučaju vraća 1. Ako se vozila ne nalaze na istom mjestu funkcija vraća 0. Možete pretpostaviti da će oba vozila biti parametrizirana s istim tipom.
- destruktork koji ispisuje: "DESTRUKTOR: unisteno vozilo na koordinatama  $(x, y)$ "

U ovom zadatku **smijete** (ali ne morate) koristiti *containere* iz STL-a (`vector`, `list`, `set` itd.), kao i `string`.

Cjelokupni program spremite pod imenom `zadatak1.cpp`.

### Napomena:

Smijete se koristiti pisanim materijalima, web-stranicom kolegija i linkovima koji vode s nje. Ne smijete se koristiti nikakvim sredstvima komunikacije (papirići, mobitel, mail i sl.). Svaki pokušaj prepisivanja i dogovaranja rezultirat će trenutnim udaljavanjem sa kolokvija. Strogo je zabranjeno fotografiranje ili pretipkavanje teksta zadataka. Smijete koristiti samo svoj korisnički račun na računalu **student!**

**PREZIME I IME:** \_\_\_\_\_

**MATIČNI BROJ STUDENTA:** \_\_\_\_\_

## PRVI ZADATAK - 2. grupa (20 bodova)

Napravite generičku strukturu (koristiti template) `Skladiste` koja opisuje skladište koje sadrži različite predmete. Pretpostavite da svako skladište sadrži najviše 10 predmeta. Svaki predmet ima svoje ime i cijenu. Može postojati više istih predmeta s različitim cijenama. Struktura `Skladiste` ima sljedeće funkcije članice:

- konstruktor koji prima dva polja i cijeli broj. Prvo polje je niz stringova koje odgovara imenima predmeta, a drugo polje je niz cijena tipa  $T$  (neki numerički tip, npr. `int` ili `float`) za te predmete. Treći parametar je cijeli broj koji odgovara veličini proslijeđenih polja. U skladište se predmeti dodaju po istom redosljedu koji imaju u proslijeđenom nizu.
- funkciju `ispisiNaEkran` koja ispisuje sadržaj skladišta na način da ispiše ime predmeta koji se nalazi u skladištu i njegovu prosječnu cijenu.
- funkciju `dodaj` koja prima ime predmeta (`string`) i cijenu predmeta (tip  $T$ ). Funkcija dodaje novi predmet ako u skladištu ima mjesta i vraća 1. U slučaju da je skladište puno vraća 0.
- funkciju `najskuplji` koja vraća ime najskupljeg predmeta u skladištu.
- funkcija `prebaci` koja prima neko drugo skladište  $S$ . Funkcija prebacuje predmete iz skladišta  $S$  (kreće od zadnjeg dodanog predmeta prema prvom) u skladište koje je pozvalo funkciju. Funkcija staje kada se skladište  $S$  isprazni ili kada se skladište koje je pozvalo funkciju popuni te vraća ukupan broj prebačenih predmeta. Možete pretpostaviti da su oba skladišta parametrizirana istim tipom.
- destruktor koji ispisuje "DESTRUKTOR: unisteno  $n$  predmeta", gdje je  $n$  broj predmeta koji se nalazio u skladištu.

U ovom zadatku **smijete** (ali ne morate) koristiti *containere* iz STL-a (`vector`, `list`, `set` itd.), kao i `string`.

Cjelokupni program spremite pod imenom `zadatak1.cpp`.

### Napomena:

Smijete se koristiti pisanim materijalima, web-stranicom kolegija i linkovima koji vode s nje. Ne smijete se koristiti nikakvim sredstvima komunikacije (papirići, mobitel, mail i sl.). Svaki pokušaj prepisivanja i dogovaranja rezultat će trenutnim udaljavanjem sa kolokvija. Strogo je zabranjeno fotografiranje ili pretipkavanje teksta zadataka. Smijete koristiti samo svoj korisnički račun na računalu **student!**

**PREZIME I IME:** \_\_\_\_\_

**MATIČNI BROJ STUDENTA:** \_\_\_\_\_

**PRVI ZADATAK - 3. grupa (20 bodova)**

Napravite generičku strukturu (koristiti template) `ZOO` koja opisuje zoološki vrt. Pretpostavite da svaki zoološki vrt ima najviše 5 kaveza (numeriranih od 1 do 5). Svaki kavez može sadržavati više životinja iste vrste dok različiti kavezi sadrže različite životinje. Struktura `ZOO` ima sljedeće funkcije članice:

- konstruktor koji prima polje stringova i dva broja. Polje je niz različitih životinja. Drugi parametar je veličina polja (`int`). Treći parametar je broj koji govori koliko je dnevno kg hrane potrebno za jednu životinju i tipa je  $T$  (neki numerički tip, npr. `int` ili `float`). Konstruktor redom dodaje u kaveze po 1 životinju iz proslijeđenog polja.
- funkciju `ispisiNaEkran` koja redom ispisuje ispisuje sadržaje kaveza (ime životinje i broj životinja u kavezu).
- funkciju `dodaj` koja prima ime životinje (`string`) i broj životinja (`int`). Funkcija dodaje proslijeđeni broj životinja u kavez koji već sadrži te životinje i vraća 1. U slučaju da ne postoji takav kavez životinje dodaje u sljedeći slobodni kavez i vraća 1. Ukoliko ne postoji slobodnih kaveza funkcija vraća 0.
- funkciju `najviseHrane` koja nalazi kavez kojem je potrebno najviše hrane dnevno te vraća tu količinu hrane (tip  $T$ ).
- funkciju `prebaci` koja prima neki drugi zoološki vrt  $V$ . Funkcija prebacuje životinje iz  $V$  (kreće od zadnjeg nepraznog kaveza prema prvom) u zoološki vrt koji je pozvao funkciju. Funkcija staje kada se  $V$  isprazni ili kada se zoološki vrt koji je pozvao funkciju popuni te vraća ukupan broj prebačenih životinja. Možete pretpostaviti da su oba zoološka vrta parametrizirana s istim tipom.
- funkciju `napraviNovi` koja kreira i vraća novi zoološki vrt. U novo kreirani zoološki vrt se prebacuju životinje iz posljednjeg nepraznog kaveza. Novi zoološki vrt je parametriziran s istim tipom kao i zoološki vrt koji je pozvao funkciju.
- destruktor koji ispisuje "DESTRUKTOR: unisteno  $n$  kaveza", gdje je  $n$  broj nepraznih kaveza.

U ovom zadatku **smijete** (ali ne morate) koristiti *containere* iz STL-a (`vector`, `list`, `set` itd.), kao i `string`.

Cjelokupni program spremite pod imenom `zadatak1.cpp`.

**Napomena:**

Smijete se koristiti pisanim materijalima, web-stranicom kolegija i linkovima koji vode s nje. Ne smijete se koristiti nikakvim sredstvima komunikacije (papirići, mobitel, mail i sl.). Svaki pokušaj prepisivanja i dogovaranja rezultat će trenutnim udaljavanjem sa kolokvija. Strogo je zabranjeno fotografiranje ili pretipkavanje teksta zadataka. Smijete koristiti samo svoj korisnički račun na računalu **student!**

**PREZIME I IME:** \_\_\_\_\_

**MATIČNI BROJ STUDENTA:** \_\_\_\_\_