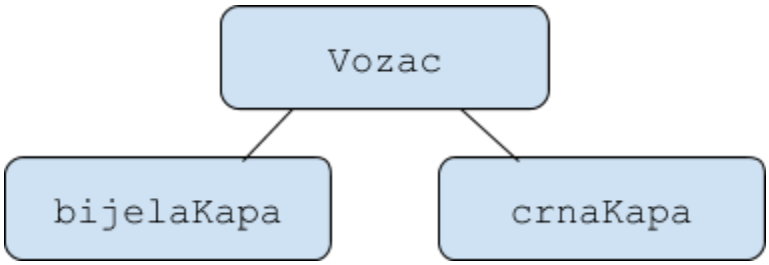
Zadatak

Napravite sljedeću hijerarhiju klasa koja opisuje utrku vozača na kružnoj stazi koja je predstavljena jediničnom kružnicom K(O,1). Svi vozači počinju utrku u točki (1,0) i pokušavaju izvesti 50 krugova u smjeru obratnom od kazaljke na satu. Trkaća staza je dovoljno široka da dva ili više vozača budu na istom položaju na stazi.

`crneKape` su sposobni izgurati drugog vozača sa staze, dok su `bijeleKape` su sposobni spriječiti izguravanje ili prestizanje.

Sučelje spremite u datoteku **vozac.h**, a implementaciju spremite u datoteku **vozac.cpp.** Sve funkcije i strukture smiju biti javne.



Vozac neka je apstraktna klasa. Implementirajte sljedeće funkcije članice:

- konstruktor(i) – prima(ju) ime (`string`) i brzinu (`int`) vozača. Brzina nam ovdje predstavlja kutnu brzinu (u stupnjevima) u jedinici vremena. To znači da vozač sa brzinom 20 obiđe puni krug (360 stupnjeva) za 18 vremenskih jedinica.
- `ime(), brzina(), polozajNaStazi(), predjeno()` - nemaju parametara, vraćaju odgovarajuće informacije o vozaču. `polozajNaStazi()` vraća int između 0(=početak staze) i 359, a `predjeno()` vraća int između 0 i 360*50-1+brzina - dakle u zadnjoj vremenskoj jedinici prije završetka vožnje, “u zaustavljanju”, može preći više od 360*50 stupnjeva.
- `Vozac& fokus()` - za bijeluKapu vraća referencu na aktivnog vozača koji je neposredno ispred na stazi, dok za crnuKapu vraća aktivnog vozača koji je neposredno iza na stazi. Neće biti test primjera gdje nema kandidata za fokus. (npr da postoji samo jedan aktivni vozač u utrci, ili poziva na vozačima koji su završili utrku).
- `prestizi(Vozac& B)` - vozač pokušava prestići vozača `B`. Ako vozač `B` nije bijelaKapa u obrambenom modu, te ako je brzina vozača `B` manja od dvostruke brzine vozača koji pokušava prestizanje, prestizanje je uspješno i prestizač se nalazi 1 kutni stupanj ispred prestizanog. U protivnom je neuspješno i nalazi se stupanj iza prestizanog. Prestizanje se radi u odnosu na polozaj na stazi, ne na

prijedjeni put. Neka je A presao 3700 stupnjeva, a B 10803 stupnja. Uspješni A.prestizi(B) postavlja A na 3904 stupnja.
Zamišljamo da se pokušaj prestizanja odigrava instantno, i samo objekt koji pokušava prestizanje mijenja položaj na stazi, svim ostalim vozačima položaj na stazi ostaje isti. Omogućite ulančavanje da isti vozač može pokušati više uzastopnih prestizanja.

- `bool izguraj()` – funkcija specifična za `crneKape`. Objekt koji je pozvao se vrati **unatrag** (unatrag=obratno od smjera utrke, smanjuje predjeni put) i pokušava izgurati prvog aktivnog vozača koji je na stazi iza njega. Uvjet za uspješno izguravanje je da vozač iza nije bijelaKapa u obrambenom modu.
  Nakon uspješnog izguravanja, objekt koji je pozvao funkciju se nalazi na stazi tamo gdje je bio izgurani, dok je izgurani van utrke (više ne sudjeluje. Naknadni pozivi funkcija za vozača koji je "ispao" trebaju ne promijeniti ništa. Pamtimo koliko je prešao do ispadanja).
  Nakon neuspješnog izguravanja je napadač na mjestu na stazi, gdje je prije bio napadani, dok je napadani jedan kutni stupanj ispred (pogledajte primjer maina).
  Ako crnaKapa izvede dva uspješna izguravanja, i sam je odmah isključen iz utrke.
  Pozivom `izguraj()` se promijene samo podaci za napadača i napadanog, i ni za kojeg trećeg vozača.
- `void brani()` – funkcija specifična za bijeleKape. Postaju imuni na jedan pokušaj prestizanja ili izguravanja. Nakon neuspješnog prestizanja ili izguravanja objekta koji je postavio brani(), on više nije imun. Imuniteti se ne mogu akumulirati, ali se nakon potrošenog imuniteta funkcija može ponovo pozvati.
- `void protokVremena(int k=1) - static` funkcija koja simulira protok k vremenskih jedinica. Za jednu vremensku jedinicu se svaki vozač pomakne za svoju brzinu u smjeru utrke. Ako pređe cilj (18000), zaustavlja se.
- `svi()` – `static` funkcija, vraća `list` pokazivača na vozače po vremenu nastanka, od starijeg do novijeg.

Možete staviti `public` pristup u svim klasama za sve varijable i funkcije. Više vozača se može nalaziti na istoj poziciji na stazi. Ako za funkciju nije napomenuto da je specifična za izvedenu klasu, znači da se odnosi na baznu klasu. Gore navedeni su **nužni** elementi strukture, smijete imati i druge dodatne. fokus, izguraj neće biti pozivane u situaciji gdje postoji samo jedan vozač.

Opće napomene

- Struktura, funkcije i datoteke moraju se zvati točno onako kako je zadano u zadatku. Pazite na mala i velika slova.
- Trebate poslati samo sučelje i implementaciju. U datoteci koju šaljete ne smije se nalaziti funkcija main()!
- Nijedna funkcija ne smije ništa učitavati s tipkovnice ili neke datoteke, niti išta ispisivati na ekran ili neku datoteku.
- Svaki od mainova kojim testiramo vaše programe neće pozivati sve navedene funkcije. Stoga, ako neku funkciju ne znate napisati, ipak možete skupiti koji bod na funkcije koje znate implementirati (u tom slučaju funkciju koju ne znate nemojte navesti u .h datoteci ili napravite neku trivijalnu implementaciju)

Ispravnost implementacija koje napišete bit će provjerena tako da ćemo mi napisati razne klijentske programe koji će deklarirati nekoliko varijabli zadane strukture, i na njima pozivati funkcije koje ste trebali napisati. Ako se poslani programi ne budu uspješno povezivali (linkali) s našim klijentskim programima, smatrat će se neispravnima.

Neki klijentski programi provjeravat će samo neke jednostavnije funkcije, dok će neki provjeravati sve funkcije koje trebate napisati. Provjera je potpuno automatska, tako da je od presudne važnosti da se pridržavate specifikacije. Nepridržavanje lako može uzrokovati osvojenih 0 bodova iz zadaće!

Upiti u vezi zadaće na mail *marijan.polic@math.hr*

Primjer maina:

```
#include<string>
#include<iostream>
#include<list>
#include”vozac.h”

using namespace std;
int main() {

	crnaKapa K("Skeletor", 23), D("Destro", 25), Z("Zod", 30);
	bijelaKapa She("Shera", 34), Bat("Batman", 37), m("GI", 12);
	list<Vozac*>::iterator i;
	list<Vozac*> S;

	K.protokVremena(20);

	cout<<K.ime()<<","<<K.brzina()<<","<< K.polozajNaStazi()<<","<<K.predjeno()<<endl;
//Skeletor,23,100,460

	cout<<D.fokus().ime()<<","<<She.fokus().ime()<<endl;    //Skeletor,Batman
	S=Vozac::svi();
	for(i= S.begin(); i!= S.end();++i)   cout<<(*i)->ime()<<" "<<(*i)->polozajNaStazi()<< "
"<< (*i)->predjeno()<< " "<<endl;
```

```
        cout<<endl;    //static listi ce test primjer pristupati samo preko funkcije svi().
//ispis je:
/*Skeletor 100 460
Destro 140 500
Zod 240 600
Shera 320 680
Batman 20 740
GI 240 240 */
        Bat.brani();   D.izguraj() ;                                  //uspjesno izgurao Skeletora
        cout<<"novi fokus:"<<D.fokus().ime()<<" "<<D.predjeno()<<endl;                  //novi
fokus:Batman , Destro presao 460
        D.izguraj() ;                                          //nije uspio izgurati Batmana
        cout<<"novi fokus:"<<D.fokus().ime()<<" "<<D.predjeno()<<endl;           //novi
fokus:Shera, Destro presao 380
        D.izguraj() ;                                                 //Uspio izgurati Sheru, ali i
sam diskvalificiran

        Bat.brani();

        cout<< Bat.polozajNaStazi()<< " "<< Bat.predjeno()<< " "<<endl ;  //21 741
        cout <<Z.polozajNaStazi()<< " "<< Z.predjeno()<< " "<<endl ;             //240 600
        cout<<endl;
      Z.prestizi(S).prestizi(D);     //ne desi se nista jer su prestizani vec ispali
        Z.prestizi(Bat);Bat.prestizi(Z);   //Z ne uspije u prestizanju ali B uspije. nece biti
test primjera sa prestizanjem iz izjednacenog polozaja
        cout <<Bat.polozajNaStazi()<< " "<< Bat.predjeno()<< " "<<endl ;         //21 1101
        cout << Z.polozajNaStazi()<< " "<< Z.predjeno()<< " "<<endl ;            //20 740
        cout<<endl;

        Vozac::protokVremena(1000);
        S=Vozac::svi();
        for(i=S.begin(); i!=S.end();++i)     cout<<(*i)->ime()<<" "<<(*i)->polozajNaStazi()<< "
"<< (*i)->predjeno()<< " "<<endl ;
/*
Skeletor 100 460      objasnjenje, ne ispis:  izbacen
Destro 320 320        izbacen
Zod 20 18020          izveo utrku i stao, zato nije predjeno 740+30000
Shera 320 680         izbacena
Batman 10 18010       izveo utrku i stao
GI 0 12240            jos vozi
*/
}
```