

II PARCIJALNI ISPIT IZ PREDMETA “TEHNIKE PROGRAMIRANJA”

NAPOMENA: Sve funkcije čija je implementacija duža od dvije naredbe obavezno implementirajte izvan klase. Također, sve metode koje su inspektori obavezno deklarirajte kao takve.

Zadatak 1 (8 poena)

U raznim oblastima nauke i tehnike često se javlja potreba rada sa promjenljivim veličinama koje se mijenjaju po sinusoidalnom zakonu oblika $A \sin(\omega t + \varphi)$, pri čemu se parametri A , ω i φ nazivaju redom amplituda, frekvencija i faza. (primjer su recimo struje i naponi u kolima naizmjenične struje). U načelu, A , ω i φ mogu biti ma kakvi realni brojevi. Međutim, bez obzira na njihove vrijednosti, uvijek je moguće dobiti ekvivalentan oblik (tzv. *normirani oblik*) u kojem su A i ω nenegativni, dok φ ima vrijednost u opsegu od $-\pi$ do π . Zaista, ukoliko je A negativno, možemo ga zamijeniti suprotnom pozitivnom vrijednošću a povećati φ za π , jer je $-A \sin(\omega t + \varphi) = A \sin(\omega t + \varphi + \pi)$. Slično, ukoliko je ω negativan, možemo ga zamijeniti suprotnom pozitivnom vrijednošću a oduzeti φ od π , s obzirom da vrijedi $A \sin(-\omega t + \varphi) = A \sin(\omega t + \pi - \varphi)$. Konačno, nakon što učinimo A i ω nenegativnim, možemo φ svesti na opseg od $-\pi$ do π tako što ćemo zamijeniti φ sa $\varphi - 2\pi \lfloor (\varphi + \pi) / (2\pi) \rfloor$ gdje $\lfloor \dots \rfloor$ označava “cijeli dio od”. Korektnost ove zamjene slijedi direktno iz periodičnosti funkcije sinus.

Interesantna osobina sinusoidalnih veličina je da zbir odnosno razlika dvije sinusoidalne veličine sa istom frekvencijom ponovo predstavlja sinusoidalnu veličinu sa istom frekvencijom, tj. vrijedi $A_1 \sin(\omega t + \varphi_1) \pm A_2 \sin(\omega t + \varphi_2) = A_3 \sin(\omega t + \varphi_3)$, pri čemu vrijednosti A_3 i φ_3 zavise na od veličina A_1 , A_2 , φ_1 i φ_2 . Ova zavisnost se najlakše može dobiti koristeći tzv. fazorski prikaz sinusoidalnih veličina, iz kojeg direktno slijedi da je $A_3 e^{i\varphi_3} = A_1 e^{i\varphi_1} \pm A_2 e^{i\varphi_2}$.

Vaš zadatak je da napravite klasu koja će služiti za podršku radu sa sinusoidalnim veličinama. Interfejs klase treba sadržavati sljedeće elemente:

- Konstruktor sa tri parametra, koji kreira objekat tipa “Sinusoida” na osnovu parametara koji predstavljaju vrijednosti amplitude, frekvencije i faze respektivno. Faza se može izostaviti, pri čemu se podrazumijeva vrijednost 0. Dozvoljavaju se bilo kakve vrijednosti za amplitudu, frekvenciju i fazu, ali pri kreiranju objekta zadanu sinusoidalnu veličinu uvijek treba svesti na ekvivalentni normirani oblik. Savjet: prije nego što počnete ovo raditi, pročitajte i dio zadatka pod c).
- Trivijalne pristupne metode koje vraćaju vrijednosti amplitude, frekvencije odnosno faze.
- Metode koje omogućavaju naknadnu izmjenu amplitude, frekvencije i faze. Dozvoljeno je zadati bilo kakve vrijednosti, ali nakon bilo koje izmjene objekat uvijek mora ostati normiran. Recimo, ukoliko se zada negativna vrijednost amplitude, objekat nakon obavljene izmjene treba svesti na ekvivalentni objekat u kojem je amplituda nenegativna. Savjet: kako u ovim metodama ima mnogo toga zajedničkog sa onim što radi konstruktor, nije loša ideja sve što je zajedničko izdvojiti u privatnu metodu (ili više njih) koja (koje) će se prosto pozivati iz konstruktora i ovih metoda.
- Preklopljene operatore “+” i “-” koji primijenjeni na dva operanda koji su sinusoidalne veličine daju kao rezultat novu sinusoidalnu veličinu koja predstavlja zbir odnosno razliku sinusoidalnih veličina koje su zadane operandima. Pri tome se pretpostavlja da obje sinusoidalne veličine imaju istu frekvenciju (u suprotnom treba baciti izuzetak). Rezultati moraju biti normirani (ako budete pametno radili, ovaj zahtjev ništa ne otežava zadatak). Savjet: za realizaciju se poslužite tipovima i funkcijama iz biblioteke “complex”. Kompleksan broj sa modulom r i argumentom (fazom) φ (tj. broj $re^{i\varphi}$) možete dobiti pomoću funkcije “polar(r , φ)”, dok modul odnosno argument kompleksnog broja z možete dobiti pomoću funkcija “abs(z)” i “arg(z)”.
- Preklopljene operatore “*” i “/” koji omogućavaju množenje sinusoidalne veličine realnim brojem (kao i ekvivalentnu operaciju množenja realnog broja sinusoidalnom veličinom), odnosno dijeljenje sinusoidalne veličine realnim brojem. Ovi operatori kreiraju nove sinusoidalne veličine dobijene kao rezultat izvršene operacije. Pri tome, sinusoidalna veličina se množi odnosno dijeli sa realnim brojem tako što se prosto amplituda pomnoži odnosno podijeli zadanim brojem, dok frekvencija i faza ostaju iste. U slučaju množenja, realan broj može biti bilo prvi bilo drugi operand, dok kod dijeljenja realni broj može biti samo drugi operand. Rezultat svakako mora biti normiran.

- f) Preklopljene operatore “+=”, “-=”, “*=” i “/=” koji omogućavaju da se kad god izrazi poput “ $X = X + Y$ ”, “ $X = X - Y$ ”, “ $X = X * Y$ ” i “ $X = X / Y$ ” imaju smisla, isti izrazi mogu skraćeno pisati kao “ $X += Y$ ”, “ $X -= Y$ ”, “ $X *= Y$ ” i “ $X /= Y$ ”.
- g) Preklopljeni operator “()” koji daje kao rezultat vrijednost sinusoidalne veličine u trenutku t , pri čemu se vrijednost t zadaje kao parametar. Uvođenje ovog operatora omogućit će da se sinusoidalne veličine ponašaju kao funkcije sa jednim argumentom t (što oni, logički gledano, zapravo i jesu).
- h) Preklopljeni operator “<<” koji omogućava ispis sinusoidalnih veličina na izlazni tok. Sinusoidalna veličina sa parametrima A , ω i ϕ ispisuje se u obliku “<A=A, w= ω , phi= ϕ >” ako je $\phi \neq 0$, odnosno u obliku “<A=A, w= ω >” ako je $\phi = 0$.

Zadatak 2 (12 poena)

Potrebno je napraviti kontejnersku klasu koja čuva kolekciju objekata, pri čemu svaki objekat može biti oblika *kvadra*, *valjka* ili *lopte*, koji se modeliraju redom pomoću klasa *Kvadar*, *Lopta* odnosno *Valjak*. Sva tri tipa objekata imaju specifičnu težinu (gustinu). Pored specifične težine, kvadar je opisan dužinom svojih stranica (a , b i c), lopta je opisana dužinom poluprečnika (r), dok je valjak opisan dužinom poluprečnika baze (r) i visinom (h). Kontejnerska klasa informacije o pohranjenim objektima čuva u *vektoru* koji sadrži pokazivače koji pokazuju na objekte tipa *Kvadar*, *Lopta* ili *Valjak* (za tu svrhu, sve ove klase će morati biti naslijeđene iz neke zajedničke apstraktne bazne klase) i taj vektor je *jedini atribut* kolekcije. Interfejs klase *Kvadar*, *Lopta* i *Valjak* omogućava da se saznaju osnovni atributi pohranjeni u njima, kao i njihova zapremina i težina (tj. specifična težina pomnožena sa zapreminom). Zapremine kvadra, lopte i valjka računaju se redom kao $V = abc$, $V = 4r^3\pi/3$ i $V = hr^2\pi$. Također trebaju postojati odgovarajući konstruktori, kao i po jedna metoda za svaku vrstu objekta koja ispisuje sve interesantne podatke o objektu nad kojim je pozvana (format ispisa odaberite po volji). Interfejs kontejnerske klase treba sadržavati sljedeće elemente:

- Konstruktor bez parametara koji kreira praznu kolekciju.
- Metode koje kreiraju novi objekat tipa *Kvadar*, *Lopta* odnosno *Valjak* i pohranjuju ga u kolekciju. Parametri ovih metoda trebaju biti u skladu sa informacijama koje su potrebne da se opiše kvadar, lopta odnosno valjak.
- Metodu koja daje broj objekata pohranjen u kolekciji.
- Destruktor, koji oslobađa svu memoriju koji je kolekcija zauzela tokom svog života.
- Konstruktor kopije i preklopljeni operator dodjele koji omogućavaju bezbjedno kopiranje i međusobno dodjeljivanje ovakvih kolekcija zasnovano na strategiji dubokog kopiranja. Napomena: razmislite dobro šta ovdje tačno treba uraditi, s obzirom sa se pokazivači na objekte čuvaju u *vektoru*, a ne u dinamički alociranom nizu. Također, nemojte zaboraviti da se radi o polimorfnoj kolekciji objekata.
- Metode koje vraćaju reference na najlakši odnosno najteži objekat u kolekciji.
- Preklopljeni operator “[]” koji vraća referencu na i -ti objekat pohranjen u kolekciji, pri čemu se indeks i zadaje unutar uglastih zagrada. Ukoliko je indeks izvan opsega, treba baciti izuzetak.
- Metodu koja vraća broj objekata u kolekciji čija je težina veća od iznosa koji se zadaje kao parametar.
- Metodu koja sortira kolekciju u rastući poredak prema težini objekata.

Napisane klase demonstrirajte u testnom programu koji čita podatke o objektima iz tekstualne datoteke, smješta ih u kolekciju i na kraju ispisuje sortirani spisak svega što se nalazi u kolekciji. Svaki objekat opisan je sa jednim redom u datoteci. Na početku reda se nalazi početno slovo “K”, “L” ili “V” (za kvadar, loptu odnosno valjak) iza kojeg slijedi prvo podatak o specifičnoj težini, a zatim podaci o dimenzijama objekta (dužine stranica za slučaj kvadra, dužina poluprečnika za slučaj lopte, odnosno dužina poluprečnika baze i visina za slučaj valjka). Podaci su razdvojeni razmakom. Radi jednostavnosti, pretpostavite da datoteka sadrži isključivo ispravne podatke.

NAPOMENA: U privatne dijelove klasa imate pravo stavljati šta god Vam treba da bi tražene klase funkcionirale u skladu sa postavkom zadatka.