

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 1a (7,5 poena):

U kompjuterskoj grafici veoma se često javlja potreba za radom sa matricama, pri čemu te matrice nikada nemaju format veći od 4×4 (mogući su i manji formati). Stoga je, radi lakšeg rada sa ovakvim matricama, veoma praktično razviti posebnu klasu, koju možemo nazvati recimo “GMatrix”. Vaš zadatak je da razvijete upravo takvu klasu, u skladu sa opisom koji slijedi. S obzirom da su maksimalne dimenzije takvih matrica ograničene na 4×4 , nema potrebe za korištenjem dinamičke alokacije memorije, niti fleksibilnih tipova podataka kao što su vektori (odnosno vektori vektôra). Stoga elemente matrice čuvajte u atributu koji je obični dvodimenzionalni niz realnih brojeva formata 4×4 . Međutim, kako format matrice može biti i manji od 4×4 , predvidite i dva atributa koji čuvaju stvarni broj redova i kolona matrice (naravno, u slučaju kada je format manji od 4×4 , neki elementi dvodimenzionalnog niza pohranjenog unutar odgovarajućeg atributa biće neiskorišteni). Pored navedenih atributa, klasa “GMatrix” treba da sadrži i sljedeće:

- Konstruktor sa tri parametra, koji redom predstavljaju broj redova, broj kolona i vrijednost na koju se inicijaliziraju svi elementi matrice. Ovaj posljednji parametar ima podrazumijevanu vrijednost 0 koja se koristi ukoliko se on izostavi. U slučaju da broj redova ili broj kolona nisu legalni (u skladu sa postavkom zadatka), treba baciti izuzetak.
- Konstruktor sa jednim parametrom koji je dvodimenzionalni niz realnih brojeva formata 4×4 . Ovaj konstruktor prosto kreira objekat tipa “GMatrix” koji modelira matricu formata 4×4 i inicijalizira je odgovarajućim vrijednostima iz dvodimenzionalnog niza koji je zadan kao parametar. Glavna uloga ovog konstruktora je da omogući automatsku konverziju dvodimenzionalnih nizova realnih brojeva formata 4×4 u objekte tipa “GMatrix”.
- Metode koje daju broj redova odnosno broj kolona matrice.
- Preklopljeni operator “+” koji daje kao rezultat zbir matrica predstavljenih operandima. Matrice se mogu sabirati samo ukoliko su istog formata, u suprotnom treba baciti izuzetak. Pri tome, odgovarajuća operatorska funkcija *ne treba da bude funkcija članica*. Napomena: da uštedite na pisanju, uputno je ovaj operator implementirati preko operatora “+=”.
- Preklopljeni operator “*” koji daje kao rezultat proizvod matrica predstavljenih operandima, odnosno proizvod matrice sa realnim brojem, pri čemu je matrica prvi operand a realni broj drugi, ili obrnuto. Matrice se mogu množiti samo ukoliko je broj kolona prve matrice jednak broju redova druge matrice. U suprotnom, treba baciti izuzetak. Odgovarajuće operatorske funkcije *ne trebaju biti funkcije članice*. Napomena: za one koji ne znaju, element u i -tom redu i j -toj koloni matrice koja je proizvod matrica A i B dobija se kao zbir proizvoda svih elemenata i -tog reda matrice A sa odgovarajućim elementima j -te kolone matrice B.
- Preklopljeni operator “+=” koji obezbjeđuje da izraz oblika “ $x += y$ ” uvijek ima isto značenje kao i izraz “ $x = x + y$ ”. Odgovarajuća operatorska funkcija *treba da bude funkcija članica*.
- Preklopljene operatore “==” i “!=” koji testiraju da li su dvije matrice jednake odnosno različite.
- Preklopljeni operator “()” koji omogućava pristup elementu u i -tom redu i j -toj koloni matrice navođenjem i i j razdvojenih zarezom unutar zagrada (npr. “ $a(2, 3)$ ” predstavlja element matrice “a” u drugom redu i trećoj koloni). Indeksi redova i kolona treba da počinju od jedinice, kako je to uobičajeno u matematici notaciji. Ukoliko indeksi nisu legalni, treba baciti izuzetak. Operator treba da vrati *referencu* na odgovarajući element da bi se omogućilo korištenje ove notacije sa lijeve strane znaka jednakosti, osim u slučaju da se ovaj operator primijeni na konstantnu matricu (u tom slučaju, ne treba vraćati referencu na element, nego njegovu kopiju).
- Preklopljeni operator “<<” koji omogućava ispis matrice na izlazni tok. Elementi matrice se ispisuju red po red, pri čemu se za ispis svakog elementa rezervira prostor od 8 znakova.
- Metode koje snimaju sadržaj matrice u binarnu datoteku čije se ime navodi kao parametar, odnosno obnavljaju sadržaj matrice iz datoteke čije se ime navodi kao parametar.

Zadatak 1b (1,5 poen):

Opišite šta bi se tačno izmijenilo u realizaciji prethodnog zadatka ukoliko bi se tražilo da klasa "GMatrixa" bude generička, tako da se tip elemenata matrice može zadavati (npr. da se podrže matrice sa cjelobrojnim ili kompleksnim elementima). Posebno istaknite kako bi se izveo preklapljeni operator "+" i operatori slični njemu.

Zadatak 2a (9,5 poena):

Za vođenje evidencije podataka o robi u nekom skladištu potrebno je razviti kontejnersku klasu nazvanu "Skladiste". U skladištu se roba nalazi pohranjena u sanducima (za čvrste predmete) i u buradima (za tečnosti). Sanduci i burad se modeliraju redom pomoću klasa "Sanduk" odnosno "Bure". Sanduk je opisan svojom težinom, nazivom predmeta koji se u njemu čuvaju (pretpostavlja se da jedan sanduk čuva samo istovrsne predmete), brojem predmeta koji se u njemu čuvaju i težinom predmeta koji se u njemu čuvaju (svi su iste težine). Bure je opisano svojom težinom, nazivom tečnosti koja se u njemu čuva, te težinom tečnosti koja se u njemu čuva. Informacijama o robi pohranjenoj u skladištu pristupa se pomoću dinamički alociranog niza pokazivača koji pokazuju na objekte tipa "Sanduk" ili tipa "Bure" (za tu svrhu, obje klase "Sanduk" i "Bure" će morati biti naslijeđene iz neke zajedničke apstraktne bazne klase). Tom nizu pokazivača se pristupa preko nekog od atributa pohranjenog unutar klase "Skladiste". Interfejs klasa "Sanduk" i "Bure" omogućava da se saznaju težine sanduka odnosno bureta, sa i bez onoga što je u njima, kao i da se ispišu podaci o sanduku odnosno buretu i onome što je u njima (format ispisa odaberite po volji). Interfejs klase "Skladiste" treba sadržavati sljedeće elemente:

- Konstruktor sa jednim parametrom koji predstavlja maksimalnu količinu objekata (sanduka odnosno buradi) koji se mogu pohraniti u skladištu. Ovaj konstruktor se ne smije koristiti za automatsku pretvorbu cijelih brojeva u objekte tipa "Skladiste".
- Destruktor, koji oslobađa svu memoriju koji su objekti tipa "Skladiste" zauzeli tokom svog života.
- Konstruktor kopije i preklapljeni operator dodjele koji omogućavaju bezbjedno kopiranje i međusobno dodjeljivanje objekata tipa "Skladiste" zasnovano na strategiji dubokog kopiranja (nemojte zaboraviti da se radi o polimorfnoj kolekciji objekata).
- Metode koje kreiraju novi objekat tipa "Sanduk" odnosno "Bure" i pohranjuju ga u skladište. Parametri ovih metoda trebaju biti u skladu sa informacijama koje su potrebne da se opiše sanduk odnosno bure. Ukoliko je skladište već popunjeno, treba baciti izuzetak.
- Metode koje vraćaju reference na najlakši odnosno najteži objekat (bure ili sanduk) u skladištu, ne računajući ono što je pohranjeno u tom objektu.
- Metodu koja vraća broj objekata u skladištu čija je ukupna težina (tj. vlastita težina zajedno sa težinom onoga što se u njima nalazi) veća od iznosa koji se zadaje kao parametar.
- Metode koje ispisuju spisak svega što se nalazi u skladištu, sortiran u opadajući poredak po ukupnoj težini.

Napisane klase demonstrirajte u testnom programu koji čita podatke o robi iz tekstualne datoteke, smješta robu u skladište i na kraju ispisuje sortirani spisak svega što se nalazi u skladištu. Svaki objekat opisan je sa dva reda u datoteci. U prvom redu se nalazi početno slovo "S" ili "B" (za sanduk odnosno za bure) iza kojeg nakon jednog razmaka slijedi naziv predmeta ili tečnosti koje su pohranjene u sanduku odnosno buretu (npr. "S Tepsije" ili "B Suncokretovo ulje"). U drugom redu se za slučaj sanduka nalazi težina sanduka, broj predmeta i težina svakog od njih, razdvojeno po jednim razmakom (npr. "1.2 50 0.35") dok se za slučaj bureta nalazi težina bureta i težina tečnosti (npr. "0.8 16.5"). 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.

Zadatak 2b (1,5 poen):

Opišite šta bi se tačno izmijenilo u realizaciji prethodnog zadatka ukoliko bi se za pristup informacijama o robi pohranjenoj u skladištu umjesto dinamički alociranog niza pokazivača koji pokazuju na objekte tipa "Sanduk" ili tipa "Bure" koristio vektor pokazivača koje pokazuju na iste takve objekte.