

## Zadaća 5.

*Ova zadaća nosi ukupno 3 poena, pri čemu prvi zadatak nosi 1 poen, drugi zadatak 0.8 poena, a treći zadatak 1.2 poena. Sva tri zadatka se mogu uraditi na osnovu gradiva sa prvih 11 predavanja i pretpostavljenog predznanja iz predmeta "Osnove računarstva". Rok za predaju ove zadaće je četvrtak, 6. VI 2013. (do kraja dana) i ne može se produžiti. Zadaće se predaju putem Zamgera.*

1. U kompjuterskoj grafici veoma se često javlja potreba za radom sa matricama, pri čemu te matrice nikada nemaju format veći od  $4 \times 4$  (mogući su i manji formati). Stoga je, radi lakšeg rada sa ovakvim matricama, veoma praktično razviti posebnu klasu, koju možemo nazvati recimo "GMatrixa". Vaš zadatak je da razvijete upravo takvu klasu, u skladu sa opisom koji slijedi. S obzirom da su maksimalne dimenzije takvih matrica ograničene na  $4 \times 4$ , 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  $4 \times 4$ . Međutim, kako format matrice može biti i manji od  $4 \times 4$ , predvidite i dva atributa koji čuvaju stvarni broj redova i kolona matrice (naravno, u slučaju kada je format manji od  $4 \times 4$ , neki elementi dvodimenzionalnog niza pohranjenog unutar odgovarajućeg atributa biće neiskorišteni). Pored navedenih atributa, klasa "GMatrixa" treba da sadrži i sljedeće:
  - a) 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.
  - b) Konstruktor sa jednim parametrom koji je dvodimenzionalni niz realnih brojeva formata  $4 \times 4$ . Ovaj konstruktor prosto kreira objekat tipa "GMatrixa" koji modelira matricu formata  $4 \times 4$  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  $4 \times 4$  u objekte tipa "GMatrixa".
  - c) Metode "DajBrojRedova" i "DajBrojKolona" koje daju broj redova odnosno broj kolona matrice.
  - d) Preklopljene binarne operatore "+" i "-" koji daju kao rezultat zbir odnosno razliku matrica predstavljenih operandima. Matrice se mogu sabirati odnosno oduzimati samo ukoliko su istog formata, u suprotnom treba baciti izuzetak. Pri tome, odgovarajuće operatorske funkcijae *ne smiju da budu funkcije članice*.
  - e) Preklopljeni binarni 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 smiju biti funkcije članice*.
  - f) Preklopljene operatore "+=" koji obezbjeđuju da izrazi oblika "X += Y", "X -= Y" i "X \*= Y" uvijek imaju isto značenje kao i izrazi "X = X + Y", "X = X - Y" i "X = X \* Y", kad god ovi posljednji slučajevi imaju smisla. Odgovarajuće operatorske funkcije *treba da budu funkcije članice*.
  - g) Preklopljeni unarni operator "-" koji daje kao rezultat matricu čiji su elementi negirani u odnosu na elemente matrice na koju je primijenjen.
  - h) Preklopljeni unarni operator "!" koji daje kao rezultat logičku vrijednost "tačno" ukoliko je matrica na koju je primijenjen nul-matrica (tj. matrica čiji su svi elementi jednaki nula), dok u suprotnom vraća logičku vrijednost "netačno".
  - i) 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).

- j) Preklopljene operatore “==” i “!=” koji testiraju da li su dvije matrice jednake odnosno različite.
- k) Preklopljeni operator “<<” koji omogućava ispis matrice na izlazni tok. Elementi matrice se ispisuju red po red, pri čemu se za ispis svakog elementa onoliko prostor koliki je postavljen pomoću poziva funkcije “cout.width” ili pomoću manipulatora “setw”. Naprimjer, ukoliko je “a” objekat tipa “GMatrica”, prilikom ispisa “cout << setw(10) << a” za ispis elemenata matrice treba rezervirati prostor od 10 mjesta. Ukoliko je zadata širina manja od 6 mjesta, treba koristiti podrazumijevanu minimalnu širinu od 6 mjesta. Uputa: da biste unutar operatorske funkcije za operator “<<” saznali koja je prethodno postavljena širina, na početku funkcije pozovite funkciju “cout.width”, ali *bez parametara*. Njen rezultat biće tekuća postavljena širina ispisa.
- l) Preklopljeni operator “>>” koji omogućava unos matrice sa ulaznog toka. Elementi matrice se nalaze unutar uglastih zagrada a međusobno se razdvajaju zarezima, dok tačka-zarez označava prelazak u sljedeću kolonu. Naprimjer, matrica formata  $2 \times 3$  čiji su elementi u prvom redu 2, 5 i 4, a elementi u drugom redu 3, 0 i -2 unosi se sa tastature kao “[2,5,4;3,0,-2]”. U slučaju bilo kakvih grešaka prilikom unosa (što uključuje i neusklađen broj unijetih elemenata sa dimenzijama matrice), ulazni tok treba da se postavi u neispravno stanje.

Sve metode implementirajte izvan klase, osim trivijalnih metoda čija implementacija može stati u jedan red ekrana. Sve metode koje su po prirodi inspektori obavezno treba deklarirati kao takve. Obavezno napišite i mali testni program u kojem će se testirati *svi zahtijevani elementi* ove klase.

- 2. Razvijte klasu “TabelarnaFunkcija” koja predstavlja tabelarno zadanu funkciju, tj. funkciju koja nije zadana analitički, nego kao skup parova realnih brojeva oblika  $(x_i, y_i)$  koji su dobijeni recimo mjerenjem. Ovi parovi će se čuvati u dinamički alociranom nizu objekata tipa “Par”, pri čemu “Par” predstavlja običnu strukturu sa dva realna atributa “x” i “y”. Interfejs ove klase treba sadržavati sljedeće elemente:
  - a) Konstruktor sa jednim parametrom koji vrši alokaciju prostora za čuvanje parova. Parametar predstavlja inicijalno predviđeni broj parova koji se mogu pohraniti (vidjećemo da se ovaj broj može naknadno mijenjati tokom rada klase). Ovaj parametar treba imati podrazumijevanu vrijednost 30, koja se koristi ukoliko korisnik ne navede parametar. U svakom slučaju, ovaj konstruktor se ne smije koristiti za automatsku konverziju cjelobrojnih podataka u tip ove klase.
  - b) Destruktor, koji oslobađa sve resurse koje su primjerci ove klase zauzeli tokom svog života.
  - c) Konstruktor kopije i preklopljeni operator dodjele koji omogućavaju da se primjerci ove klase mogu bezbjedno kopirati i međusobno dodjeljivati na bazi dubokog kopiranja.
  - d) Metodu “DodajPar” sa dva parametra koja dodaje novi par  $(x, y)$  u postojeći skup parova. Parametri metode su upravo  $x$  i  $y$ . Ukoliko u skupu parova već postoji par čija je prva koordinata jednaka  $x$ , treba baciti izuzetak (s obzirom da nije moguće imati dvije vrijednosti funkcije  $y$  za istu vrijednost  $x$ ). Ukoliko se popuni sav alocirani prostor, treba alocirati novi prostor sa 20 dodatnih mjesta, iskopirati u njega sve pohranjene parove, obrisati stari (popunjeni) prostor i nastaviti dodavanje u novi prostor.
  - e) Metodu “ObrisiSve” koja briše sve unesene parove.
  - f) Metodu “ObrisiPar” sa jednim parametrom  $x$  koja briše iz kolekcije par čija je prva koordinata  $x$  ukoliko takav postoji, a u suprotnom baca izuzetak.
  - g) Metodu “TabelirajFunkciju” sa četiri parametra  $f$ ,  $xmin$ ,  $xmax$  i  $dx$  pri čemu je  $f$  funkcija koja prima realni broj a vraća realni broj kao rezultat, dok su  $xmin$ ,  $xmax$  i  $dx$  realni brojevi. Metoda treba da u kolekciju doda sve parove oblika  $(x, f(x))$  za sve vrijednosti  $x$  od  $xmin$  do  $xmax$  u koraku  $dx$  (tj. da tabelira funkciju  $f$  na zadanom intervalu sa zadanim korakom i da pohrani rezultate tabeliranja u kolekciju).
  - h) Preklopljeni operator “( )” koji vraća vrijednost funkcije u tački  $x$  (koja se zadaje kao argument) dobijenu postupkom linearne interpolacije (to je ono što se dobije kada se

pretpostavi da su sve tačke  $(x_i, y_i)$  u rastućem poretku po  $x$ -ovima prosto spojene dužima). Formula za linearnu interpolaciju glasi  $y = y' + (y'' - y')(x - x') / (x'' - x')$  gdje je  $x'$  najveća vrijednost među vrijednostima  $x_i$  koje su manje ili jednake od  $x$ ,  $x''$  je najmanja vrijednost među vrijednostima  $x_i$  koje su veće od  $x$ , dok su  $y'$  i  $y''$  odgovarajuće vrijednosti  $y_i$  koje odgovaraju  $x'$  i  $x''$ . Ukoliko  $x'$  ili  $x''$  ne postoje (npr. ako nema vrijednosti  $x_i$  koje su manje od  $x$ ), interpolacija nije moguća i treba baciti izuzetak.

Sve metode implementirajte izvan klase, osim trivijalnih metoda čija implementacija može stati u jedan red ekrana. Sve metode koje su po prirodi inspektori obavezno treba deklarirati kao takve. Obavezno napišite i mali testni program u kojem će se testirati *svi zahtijevani elementi* ove klase.

3. Za potrebe neke meteorološke stanice neophodno je vršiti čestu registraciju količine padavina. Za tu svrhu meteorološka stanica koristi računarski program u kojem je definirana i implementirana klasa nazvana "Padavine". Ova klasa omogućava čuvanje podataka o količini padavina za izvjesni vremenski period u vektoru cijelih brojeva, kojem se pristupa preko odgovarajućeg privatnog atributa (količina padavina se zadaje u centimetrima, zaokruženo na cijeli broj). Interfejs klase sadrži sljedeće elemente:
  - a) Konstruktor sa jednim parametrom koji predstavlja maksimalno dozvoljenu količinu padavina koja se može registrirati (oprez: ovo nije maksimalan broj količina padavina koje se mogu registrirati, nego maksimalan iznos količine padavina koji se smije zadati pri jednoj registraciji). Potrebno je zabraniti da se ovaj konstruktor koristi za automatsku konverziju cijelih brojeva u objekte tipa "Padavine", s obzirom da bi takva konverzija bila besmislena.
  - b) Metodu "RegistrirajPadavine" koja vrši registraciju nove količine padavina, pri čemu se količina padavina koja se registrira prenosi kao parametar metode. U slučaju je količina padavina manja od nule ili veća od maksimalne dozvoljene količine padavina, metoda treba da baci izuzetak.
  - c) Metodu "DajBrojRegistriranihPadavina" koja daje broj registriranih količina padavina.
  - d) Metodu "BrisiSve" koja briše sve unesene količine padavina.
  - e) Metode "DajMinimalnuKolicinuPadavina" i "DajMaksimalnuKolicinuPadavina" koje vraćaju kao rezultat minimalnu i maksimalnu količinu padavina (u slučaju da nema registriranih količina padavina, obje metode treba da bace izuzetak). Za realizaciju nije dozvoljeno koristiti petlje, nego odgovarajuće funkcije i/ili funktore iz biblioteka "algorithm" i "functional".
  - f) Metodu "DajBrojDanaSaPadavinamaVecimOd" koja vraća kao rezultat broj dana u kojima je količina padavina bila veća od vrijednosti koja se zadaje kao parametar (u slučaju da nema registriranih količina padavina, metode treba da bace izuzetak). Za realizaciju nije dozvoljeno koristiti petlje, nego odgovarajuće funkcije i/ili funktore iz biblioteka "algorithm" i "functional".
  - g) Metodu "Ispisi" koja ispisuje sve unesene (registrirane) količine padavina sortirane u opadajućem poretku (tj. najveća količina padavina se ispisuje prva), pri čemu se svaka količina padavina ispisuje u posebnom redu. Pri tome je neophodno koristiti funkcije i/ili funktore iz biblioteka "algorithm" i "functional". Ova funkcija obavezno treba biti inspektor funkcija, tj. treba biti deklarirana sa modifikatorom "const"!
  - h) Preklopljeni operator "[ ]" koji omogućava da se direktno pročita  $i$ -ta registrirana količina padavina (numeracija ide od jedinice). Ukoliko je indeks izvan dozvoljenog opsega, treba baciti izuzetak. Pri tome, ovaj operator se *ne može koristiti* za izmjenu podataka, odnosno ne može se koristiti sa lijeve strane znaka jednakosti.
  - i) Preklopljeni operator "++" koji povećava sve registrirane količine padavina za jedinicu (pri tome se za jedinicu povećava i informacija o maksimalno dozvoljenoj količini padavina). Potrebno je podržati kako prefiksnu, tako i postfixnu verziju ovog operatora. Za realizaciju nije dozvoljeno koristiti petlje, nego odgovarajuće funkcije i/ili funktore iz biblioteka "algorithm" i "functional".
  - j) Preklopljene operatore "+" i "-" koji djeluju na sljedeći način: Ukoliko je "x" objekat tipa "Padavine", a "Y" cijeli broj, tada je "x + Y" novi objekat tipa "Padavine" u kojem su sve registrirane količine padavina povećane za iznos "Y" (u novodobijenom objektu treba

ažurirati informaciju o maksimalno dozvoljenoj količini padavina). Na sličan način se interpretira i izraz  $x - y$  u slučaju da je  $x$  objekat tipa "Padavine", a  $y$  cijeli broj. Pri tome, ukoliko se kao rezultat sabiranja ili oduzimanja dobije da neka od količina padavina postane negativna, treba baciti izuzetak. U slučaju kada su i  $x$  i  $y$  objekti tipa "Padavine", tada je izraz  $x - y$  novi objekat tipa "Padavine" koji sadrži *razlike* odgovarajućih količina padavina iz objekata  $x$  i  $y$ . U ovom posljednjem slučaju se podrazumijeva da  $x$  i  $y$  sadrže isti broj registriranih količina padavina, kao i da su registrirane padavine u objektu  $x$  veće ili jednake od odgovarajućih registriranih padavina u objektu  $y$  (u suprotnom, treba baciti izuzetak). U svim ostalim slučajevima, značenje izraza  $x + y$  odnosno  $x - y$  nije definirano. Za realizaciju ovih operadora nije dozvoljeno koristiti petlje, nego odgovarajuće funkcije i/ili funktore iz biblioteka "algorithm" i "functional".

- k) Preklopljene operatore "+=" i "-=" čiji je cilj da značenje izraza oblika  $x += y$  odnosno  $x -= y$  bude identično značenju izraza  $x = x + y$  i  $x = x - y$  kad god oni imaju smisla.
- l) Preklopljeni unarni operator "-" koji daje kao rezultat novi objekat tipa "Padavine" u kojem su sve količine padavina oduzete od maksimalno dozvoljene količine padavina. Za realizaciju nije dozvoljeno koristiti petlje, nego odgovarajuće funkcije i/ili funktore iz biblioteka "algorithm" i "functional".
- m) Preklopljene relacione operatore "==" i "!=" koje ispituju da li su dva objekta tipa "Padavine" jednaka ili nisu. Dva objekta ovog tipa smatraju se jednakim ukoliko sadrže isti broj registriranih količina padavina, i ukoliko su sve odgovarajuće registrirane količine padavina oba objekta jednake. Za realizaciju ovih operadora nije dozvoljeno koristiti petlje, nego odgovarajuće funkcije i/ili funktore iz biblioteka "algorithm" i "functional".

Implementirajte klasu sa navedenim svojstvima. Sve neophodne atribute treba obavezno izvesti kao privatne članove klase, a sve metode implementirajte izvan klase, osim metoda čija je implementacija dovoljno kratka, u smislu da zahtijeva recimo jednu ili dvije naredbe. Sve metode koje su po prirodi inspektori obavezno treba deklarirati kao takve. Obavezno napišite i mali testni program u kojem će se testirati sve elemente napisane klase.