Strukture podataka i algoritmi

Projekat 1 – Tema 3

Klasa Matrica

Class Documentation

Matrica Class Reference

`#include <matrica.h>`

Detailed Description

Zadatak je bio da se razvije klasa koja čuva matricu prozvoljnog formata i koja će imati mogućnost osnovnih operacija rada s matricama, kao što je sabiranje, oduzimanje, množenje (skalarno i matrično), računanje inverzne, transponovane, adjungovane matrice i determinante kvadratne matrice.

Ispod je puna dokumentacija za implementiranu klasu Matrica, koja čuva proizvoljnu matricu realnih brojeva pomoću dvostrukog pokazivača. Korisnik nije dužan vršiti alokaciju i dealokaciju ručno. U svrhu bržeg množenja kvadratnih matrica implementirana je Strassenova funkcija za brzo množenje matrica.

Podržano je unošenje matrica iz terminala (operator >>), koji također omogućava računanje nekih kompleksnijih izraza.

Pravila i sintaksa za unos matrica u konzolu:

1. Matrica se unosi isključivo između uglasti zagrada u sljedećem formatu:
   [a b c; d e f; g h i] (a,b,…,i su realni brojevi), redovi se moraju odvajati sa simbolom ";"
2. Osnovne operacije se unose trivijalno (+,*,-)
3. Stepenovanje matrice je omogućeno preko operatora "^", cijelobrojni eksponent se piše isključivo bez zagrada!
4. Transponovanje ^T (slovo "t" je obavezno veliko) i invertovanje "^-1" (isključivo bez zagrada)
5. Jedinična matrica se unosi slovom "E" ili "I", obavezan je cijeli broj odmah nakon oznake koji će određivati red jedinične matrice.
6. Za grupisanje izraza je moguće koristiti obične zagrade "( )".

Sve ove funkcije, i dodatno funkciju determinanta(), moguće je pozvati direktno iz programa u main.cpp.

Unošenje nedozvoljenih znakova rezultira bacanjem izuzetka.

Author

Benjamin Hodžić

Public Member Functions

- **Matrica** ()
  Konstruktor bez parametara. Dinamički alocira nul-matricu formata 3x3.

- **Matrica** (int r)
  Konstruktor s jednim parametrom koji generiše jediničnu matricu reda r.

- **Matrica** (int j, int k)
  Konstruktor sa dva parametra.

- **Matrica** (const **Matrica** &r)
  *Konstruktor kopije klase **Matrica**.*

- **Matrica** & **operator=** (**Matrica** &r)
  *Operator dodjele klase **Matrica**.*

- **Matrica** (**Matrica** &&r)
  *Move konstruktor klase **Matrica**.*

- **Matrica** & **operator=** (**Matrica** &&r)
  *Move operator dodjele klase **Matrica**.*

- **~Matrica** ()
  *Destruktor klase **Matrica**.*

- **Matrica** & **operator*** (double skalar)
  *Operator * definisan za množenje matrice skalarom.*

- **Matrica** & **operator+** (const **Matrica** &a)
  Sabiranje matrica.

- **Matrica** & **operator-** (**Matrica** a)
  Oduzimanje matrica.

- **Matrica operator*** (**Matrica** &a)
  *Operator * definisan za množenje matrica.*

- **Matrica operator^** (int stepen)
  Brzo stepenovanje matrica.

- **Matrica adjungovana** ()
  Adjungovana matrica.

- **Matrica submatrica** (int red, int kol)
  Funkcija za kreiranje submatrice izuzimanjem predefinisanog reda i kolone.

- double **determinanta** ()
  Determinanta matrice.

- bool **regularna** ()
  Provjera regularnosti matrice.

- **Matrica transponovana** ()
  Transponovana matrica.

- **Matrica inverzna** ()
  Inverzna matrica.

Static Public Member Functions

- static **Matrica** * **ucitajMatricu** ()
  Statička funkcija koja učitava matricu iz ulaznog toka.

Friends

- **Matrica strassen** (**Matrica** &l, **Matrica** &d, int red)
  Funkcija za brzo množenje matrica.

- **Matrica** & **operator*** (double skalar, **Matrica** &a)
  Množenje matrice skalarom.

- void **izvrsiBinarnuOperaciju** (stack< **Matrica** * > &m, stack< char > &o, stack< st > &op, stack< double > &sk)
  Izvrsavanje operacije i vraćanje odgovarajućeg elementa na `stack` .

- ostream & **operator<<** (ostream &izlaz, const **Matrica** &a)
  Ispisivanje matrice na izlazni tok.

- istream & **operator>>** (istream &ulaz, **Matrica** &a)
  Operator izdvajanja, čitanje matrice iz ulaznog toka.

Constructor & Destructor Documentation

Matrica::Matrica (int *r*)

Konstruktor s jednim parametrom koji generiše jediničnu matricu reda r.

Parameters

r	Red matrice za instanciranje.

Matrica::Matrica (int *j*, int *k*)

Konstruktor sa dva parametra.

Dinamički alocira nul-matricu predefinisanog formata.

Parameters

redovi	Broj redova matrice.
kolone	Broj kolona matrice.

Matrica::~Matrica ()

Destruktor klase **Matrica**.

Uništava svaki niz(red) koji je dinamički alociran pri instanciranju objekta, uništava i dvostruki pokazivač (niz) koji pokazuje na nizove.

Member Function Documentation

Matrica Matrica::adjungovana ()

Adjungovana matrica.

Funkcija koja kreira i vraća odgovarajuću adjungovanu matricu. Adjungovana matrica je jednaka ekvivalentnoj matrici svojih kofaktora koja se na kraju transponuje.

See also

`Matrica transponovana();`

Exceptions

exception	Funkcija baca izuzetak ukoliko matrica nije kvadratna.

double Matrica::determinanta ()

Determinanta matrice.

Determinanta matrice je proizvod svih kofaktora jednog reda/kolone i njima odgovarajućih elemenata. $A_{ij} = ((-1)^{i+j} * M_{ij}$, gdje je M_{ij} minor odgovarajuceg elementa).

Funkcija je rekurzivna i radi u vremenu *O(n!)*

Exceptions

exception	Funkcija baca izuzetak ukoliko matrica nije kvadratna.

Matrica Matrica::inverzna ()

Inverzna matrica.

Funkcija vraća inverznu matricu kvadratne, regularne matrice. Računa se kao `1/detA * adj(A)`

Exceptions

exception	Baca izuzetak ukoliko je matrica singularna.

Matrica Matrica::operator* (Matrica & *a*)

Operator * definisan za množenje matrica.

Klasično množenje matrica, ukoliko je tačan uslov `this->kolone != a.redovi` funkcija baca izuzetak.

Ukoliko su matrice istog formata, kvadratne, i reda koji je stepen broja 2, poziva se funkcija brzog množenja matrica

See also

`friend` **`Matrica strassen(Matrica& l, Matrica& d, int red);`**

Returns

Vraća se matrica koja ima redova koliko i prva matrica, a kolona kao druga matrica.

Matrica & Matrica::operator+ (const Matrica & *a*)

Sabiranje matrica.

Ukoliko matrice nisu istog formata, funkcija baca izuzetak.

Matrica & Matrica::operator- (Matrica *a*)

Oduzimanje matrica.

Ukoliko matrice nisu istog formata, funkcija baca izuzetak.

Matrica Matrica::operator^ (int *stepen*)

Brzo stepenovanje matrica.

Funkcija se izvršava rekurzivno sve dok stepen ne dosegne 1, radi u vremenu $O(\log_2 n)$

Parameters

stepen	Stepen/eksponent izraza. Pri svakom rekurzivnom pozivu se polovi.

Exceptions

exception	Funkcija baca izuzetak ukoliko matrica nije formata nxn.

bool Matrica::regularna ()

Provjera regularnosti matrice.

Matrica je regularna ukoliko joj je determinanta različita od 0, inače je singularna.

Matrica Matrica::submatrica (int *red*, int *kol*)

Funkcija za kreiranje submatrice izuzimanjem predefinisanog reda i kolone.

Služi kao pomoćna funkcija pri određivanju determinante matrice.

Rezultujuća matrica se prepisuje "ručno", pri čemu pomoćne bool varijable služe za evidenciju pozicije (red, kol) u matrici radi izbjegavanja prekoračenja indeksa.

See also

`double` **`determinanta();`**

Parameters

red	Zadati red koji će se ignorisati pri kreiranju matrice.
kolona	Zadata kolona koja će se ignorisati pri kreiranju matrice.

Returns

Instanca klase **Matrica** koja će biti submatrica matrice nad kojom je funkcija pozvana.

Matrica Matrica::transponovana ()

Transponovana matrica.

Funkcija koja vraća transponovanu matricu, tj gdje vrijedi $A_{ij} = A_{ji}$

Matrica * Matrica::ucitajMatricu () [static]

Statička funkcija koja učitava matricu iz ulaznog toka.

Pri nailasku na znak '[' poziva se ova funkcija.

See also

`friend istream& operator >> (istream& ulaz, Matrica& a);`

Brojevi se izdvajaju iz ulaznog toka, jedan po jedan. Kraj reda se označava sa znakom ';'

Returns

Vraća pokazivač na novokreiranu instancu klase **Matrica**.

Exceptions

exception	Izuzetak se baca ako je matrica grbava ili ako je unesen nepčekivan znak.

Friends And Related Function Documentation

void izvrsiBinarnuOperaciju (stack< Matrica * > & *m*, stack< char > & *o*, stack< st > & *op*, stack< double > & *sk*) [friend]

Izvrsavanje operacije i vraćanje odgovarajućeg elementa na `stack` .

Referentna tačka gledišta funkcije je stack pobrojanog tipa `st` .

```
typedef enum {matrica, otvorenaZ, zatvorenaZ, skalar, operacija} st;
```

Warning

Ovaj stack prima samo tipove *matrica, skalar*

Na osnovu datih kominacija ova 2 tipa računa se, i vraća rezultat na odgovarajući stack. (npr. matrica*matrica = matrica ili skalar*matrica = matrica).

Funkcija omogućava čak i isključivo operacije sa skalarima, sve dok rezultat čitavog izraza nije skalar (tada dolazi do bacanja izuzetka).

Parameters

m	Stack pokazivača na sve unesene matrice izraza.
p	Stack karaktera koji predstavljaju operacije.
op	Stack pobrojanih tipova `st` koji predstavljaju generičke operande.
sk	Stack svih unesenih skalara izraza.

Exceptions

exception	Izuzetak se baca ukoliko neki stack nema dovoljno elemenata za rad funkcije.

Matrica& operator* (double *skalar*, Matrica & *a*) [friend]

Množenje matrice skalarom.

Ista kao i **Matrica**& operator* (double skalar); Služi da omogući komutativnost množenja matrica skalarom.

ostream& operator<< (ostream & *izlaz*, const Matrica & *a*) [friend]

Ispisivanje matrice na izlazni tok.

Formatirani ispis matrice realnih brojeva. Svi elementi su odvojeni praznim mjestom, maksimalan broj decimala je 5 , a redovi su odvojeni praznim redom.

istream& operator>> (istream & *ulaz*, Matrica & *a*) [friend]

Operator izdvajanja, čitanje matrice iz ulaznog toka.

Ova funkcija omogućava i čitanje, parsiranje i računanje složenijih izraza, kao što su +,-,*,^,...

Vodi se računa o prethodnom znaku/operandu pomoću globalnog pobrojanog tipa

```
typedef enum {matrica, otvorenaZ, zatvorenaZ, skalar, operacija} st;
```

Koncept računanja izraza je čuvanje operanda na stack<st> op , pri čemu u stacku op je dozvoljeno čuvanje samo tip *matrica, skalar* iz st pobrojanog tipa. S tim je moguće utvrditi posljednja 2 operanda i samo pozivanje odgovarajućih funkcija implementiranih ranije:

See also

friend void **izvrsiBinarnuOperaciju(stack<Matrica*>& m, stack<char>& o, stack<st>& op, stack<double>& sk);**

kao i njihovu odgovarajuću binarnu operaciju koristeći stack karaktera koji predstavljaju binarne operacije stack<char> operacija i njihov prioritet *('*' > '+' = '-')*

See also

int **prioritetOperacije(char znak);**

Rezultujuća matrica je jedina matrica na stack<Matrica*> m

, ukoliko je izraz bio ispravan.

Matrica strassen (Matrica & *l*, Matrica & *d*, int *red*) [friend]

Funkcija za brzo množenje matrica.

Poziva se samo ukoliko su obje matrice kvadratne i red im je stepen broja 2.

Ideja je da se množenje svede na 7 rekurzivnih poziva, za razliku od 8 kod klasičnog množenja.

Matrice se dijele na 4 podmatrice koje su reda red/2 , rekurzija staje kad matrice dosegnu red 2. Neka su *a,b,c,d* podmatrice(kvadranti) lijeve matrice, a *e,f,g,h* podmatrice desne matrice respektivno. Tada su pomoćne matrice (označimo ih sa p1, p2,.., p7):

```
p1 = a*(f-h);
p2 = (a+b)*h;
p3 = (c+d)*e;
p4 = d*(g-e);
p5 = (a+d)*(e+h);
```

```
p6 = (b-d)*(g+h);
p7 = (a-c)*(e+f);
```

a rezultuće su jednake:

```
c11 = p5+p4-p2+p6;
c12 = p1+p2;
c21 = p3+p4;
c22 = p1+p5-p3-p7;
```

Vremenska kompleksnost funkcije je $O(n^{\log_2 7})$ ili $O(n^{2.81})$

Parameters

lijeva	Lijeva matrica u izrazu.
desna	Desna matrica u izrazu.
red	Trenutni red matrica u izrazu.

Returns

Novogenerisana kvadratna matrica inicijalnog reda koja je proizvod matrice `lijeva` i `desna`.