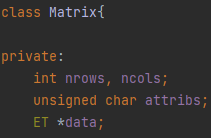
به نام خدا

تمرین شماره 3

روح اله خادمی

9932803

در این تمرین سعی شده از ویژگی های شی گرای زبان c++ استفاده شود. به همین دلیل بنای اصلی این تمرین را کلاس ها قرار می دهیم.



اسم کلاس مورد نظر را Matrix می گذاریم. متغیرهای خصوصی را

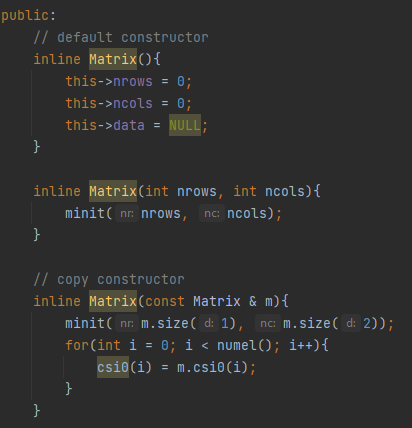
Nrows

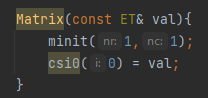
Ncols

Attribs

Data

قرار می دهیم. همان طور که از نام متغیرها مشخص است، متغیر Nrows بیانگر تعداد سطرها، Ncols بیانگر تعداد ستون ها، Attribs بیانگر ویژگی ها ماتریس که در این تمرین مورد نیاز نیست و Data که بیانگر عناصر ماتریس است.





سپس عضوهای عمومی کلاس را تعریف میکنیم.

عضوهای عمومی کلاس را توابع سازنده کلاس و توابع دیگر کلاس تشکیل می دهند.

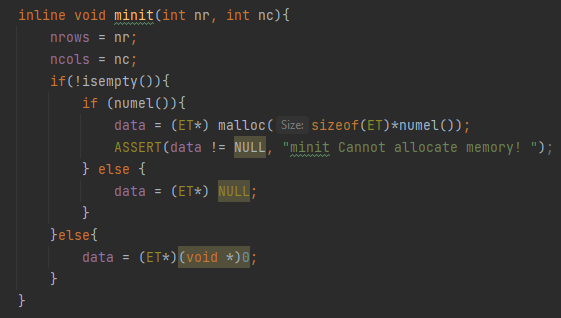
برای کلاس چهار تابع سازنده تعریف می کنیم. یکی تابع سازنده پیش فرض و سه تای دیگر سازنده کپی.

تابع سازنده پیش فرض یک ماتریس با تعداد سطر و ستون صفر می سازد و مقدار data را برابر NULL قرار می دهد.

سه تابع سازنده دیگر یکی سطر و ستون را می گیرد و ماتریسی با اندازه مورد نیاز می سازد و دیگری یک ماتریس را میگیرد و یک کپی از آن می سازد. آخری هم یک عدد اعشاری تک دقتی دریافت می کند و یک ماتریس تک عضوی از روی آن می سازد.

تابع استفاده شده در این سازنده ها تحت عنوان minit به طور جداگانه توضیح داده خواهد شد.

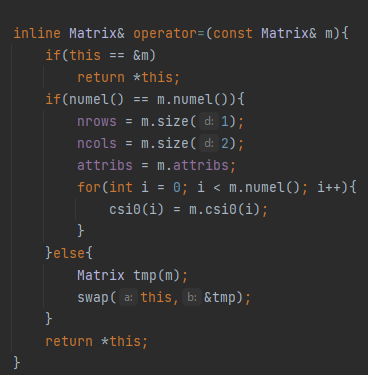
تابع minit:



این تابع دو عدد طبیعی به عنوان سطر و ستون دریافت می کند.سپس بررسی می کند که ماتریس ما خالی نباشد و سپس بررسی میکند که برای سطر یا ستون ماتریس عدد صفر قرار داده نشود.اگر ماتریس خالی بود مقدار NULL را به ET\* کست میکند (چون نوع داده ای فیلد دیتا ET\* است.) اگر هم سطر یا ستون ماتریس برابر صفر باشد void\* 0 را به ET\* کست می کند و در فیلد data قرار می دهد. اگر شرایط برای همه چیز اوکی بود به اندازه ی مورد نیاز بر اساس تعداد سطر و ستون ها و حافظه ی مورد نیاز هر عضو داده ای ET، حافظه از سیستم دریافت می کنند و در اختیار فیلد data ماتریس مورد نظر می دهد. سپس موفق آمیز بودن عملیات را چک می کند.

تعریف اپراتور مساوی ( = ) :

قرارداد: this را ماتریس کلاس می نامیم.

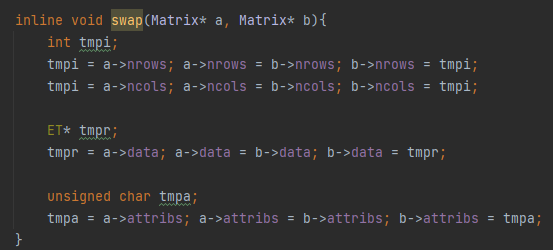


با استفاده از این تابع می توان عملگر تساوی را بین دو ماتربیس تعریف کرد. بدین صورت که وقتی عمل تساوی را بین دو ماتریس قرار می دهیم سمت چپ تساوی ماتریس موجود در کلاس ما

(همان this) و سمت راست تساوی &m قرار میگیرد. درون تابع قبل از هرچیز بررسی میکنیم که اگر این دو ماتریس برابر بودند همان **مقدار** this را برگرداند.

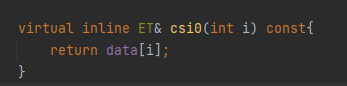
در اخر بررسی می کند اگر اندازه ی این دو ماتریس برابر بود، تعداد سطرهای ماتریس کلاس را برابر ماتریس ورودی، تعداد ستون های ماتریس کلاس را برابر ماتریس ورودی و ویژگی های ماتریس کلاس را برابر ویژگی های ماتریس ورودی قرار می دهد. سپس با استفاده از تابع اندیس دهی خطی csi0 و یک حلقه، اعداد موجود در ماتریس ورودی را یک به یک در ماتریس کلاس قرار می دهد.

اما اگر اندازه ی دو ماتریس برابر نبود، با استفاده از سازنده کپی که ماتریس می گرفت و یک کپی از ماتریس میساخت، یک کپی از ماتریس m می سازیم وبا استفاده از تابع swap جای آنرا با ماتریس کلاس عوض می کنیم. سپس مقدار ماتریس کلاس را خروجی می دهیم.



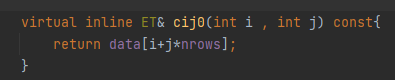
این تابع اشاره گر های دو ماتریس را دریافت می کند و و تعداد سطرها، تعداد ستون ها و مقادیر موجود در فیلد data هر ماتریس را با هم جابجا می کند.

تابع csi0:



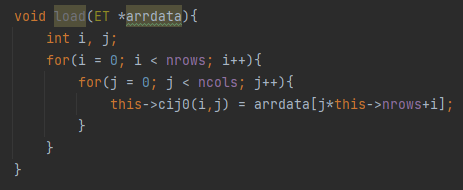
این تابع یک عدد طبیعی i دریافت می کند و عنصر i ام ماتریس را خروجی می دهد.

تابع cij0:



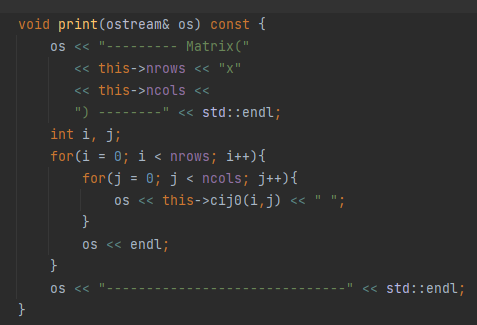
این تابع i و j طبیعی را دریافت می کند و عدد موجود در سطر i ام و ستون j ام ماتریس را بر می گرداند.

تابع load:



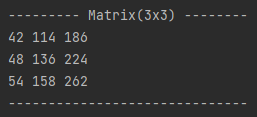
این تابع یک اشاره گر به نوع داده ای ET(در واقع همان آرایه ای از اعداد float) دریافت میکند و اعداد را به ترتیب در ماتریس کلاس می چیند.

تابع print:



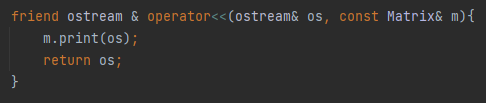
این تابع ماتریس کلاس را به شکل ماتریس های موجود در واقعیت و به شکل سطر و ستون های زیر هم چاپ می کند. البته قبل از چاپ ماتریس تعداد سطر و ستون آنرا نیز چاپ می کند.

بدین شکل:



تعریف اپراتور << :

با استفاده از تابع print ماتریس کلاس را چاپ می کند.

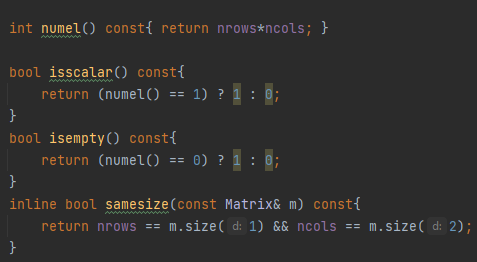


تابع size:



این تابع یک عدد صحیح دریافت می کند. اگر عدد دریافت شده 1 بود تعداد سطرها و اگر یک نبود تعداد ستون ها را خروجی می دهد. البته بهتر است کاربر فقط عدد 1 یا 2 را بررسی کند.

توابع numel، isscalar، isempty، samesize:



این توابع به ترتیب :

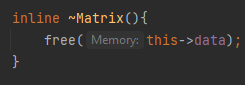
تعداد کلی داده های موجود در ماتریس را بر می گرداند.

اسکالر بودن ماتریس را بررسی می کند.

خالی بودن ماتریس را بررسی می کند.

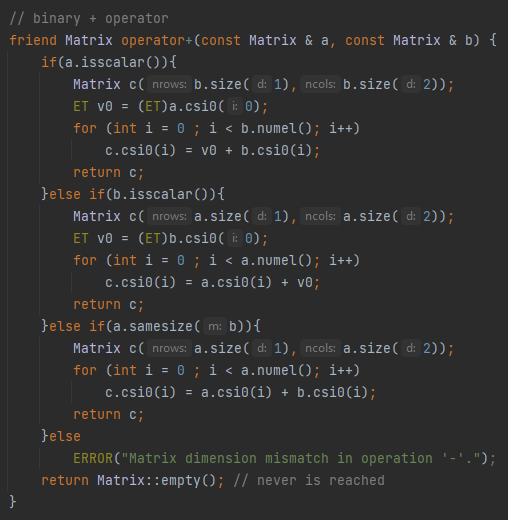
هم اندازه بودن ماتریس کلاس با ماتریس ورودی را بررسی میکند.

تابع مخرب:



این تابع حافظه ای را که توسط دستور malloc به data اختصاص دادیم را آزاد می کند.

تعریف عملگر + بین دو ماتریس یا عدد و ماتریس:

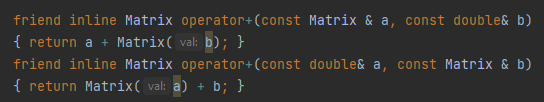


اگر ماتریس اول اسکالر باشد عدد آنرا استخراج می کند و عدد موجود را با تمام اعضای ماتریس دیگر جمع می کند. اگر ماتریس دوم اسکالر باشد همینکار را می کند.

اگر هیچکدام اسکالر نبودند و هم اندازه بودند اعدادنظیر به نظیر ماتریس را با هم جمع می کند.

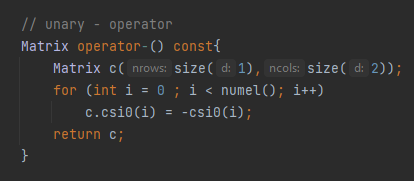
برای هرکدام از این سه حالت با استفاده از تابع سازنده یک ماتریس هم اندازه با این دو ماتریس ساخته می شود و حاصل این عملیات ها در این ماتریس جدید ریخته شده و برگردانده می شود.

البته اگر اندازه ی ماتریس ها برابر نبودند پیغام خطا چاپ می شود و برنامه بسته خواهد شد و یک ماتریس خالی برگردانده ی شود.



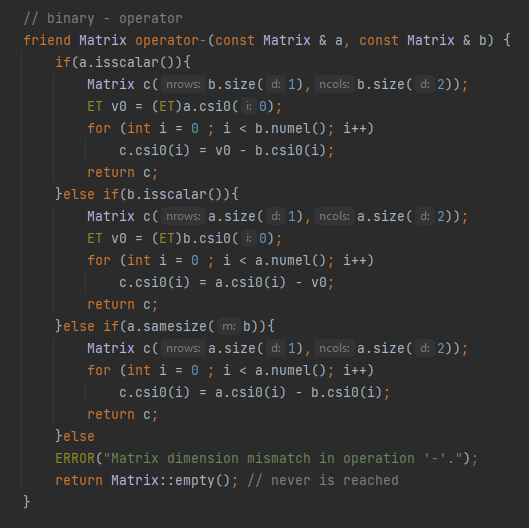
حال اگر هرکدام از ورودی ها عدد بودند، به یک اسکالر تبدیل می شود و همان عملیات قبلی تکرار خواهد شد.

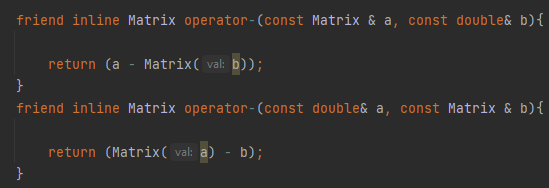
تعریف عملگر یگانی - :



این تابع یک ماتریس هم اندازه ماتریس کلاس می سازد و تمام اعضای ماتریس کلاس را قرینه می کند و در ماتریس جدید قرار می دهد. نهایتا ماتریس جدید را خروجی می دهد.

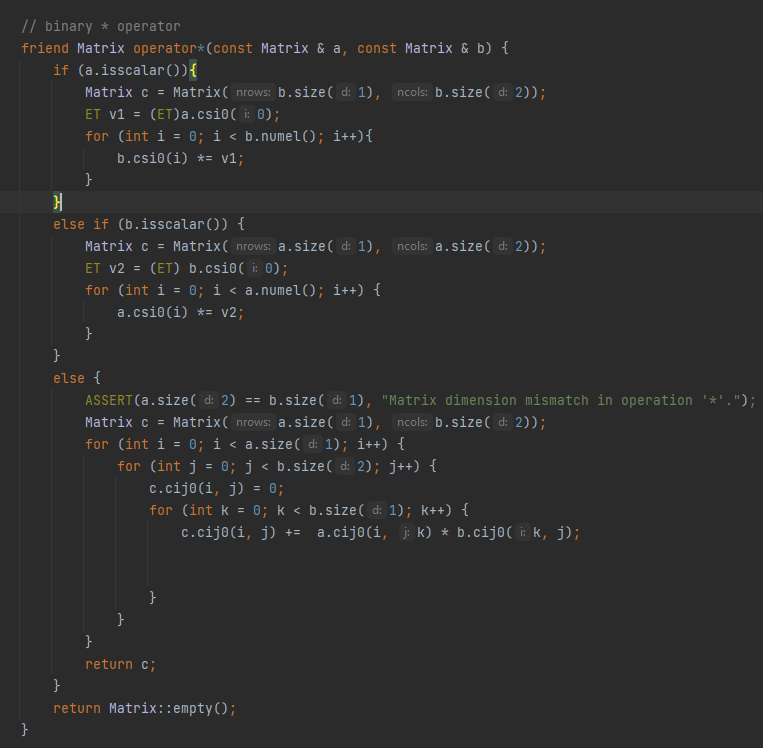
تعریف عملگر – بین دو ماتریس یا عدد و ماتریس:

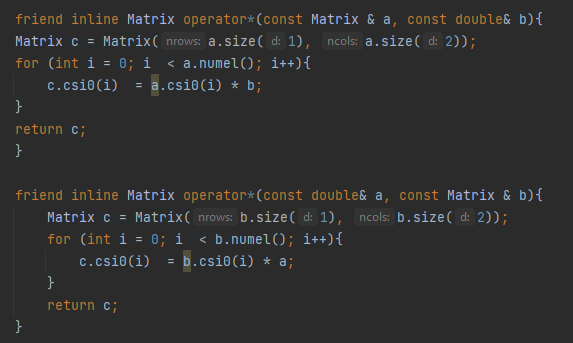




همان روند عملگر + را این بار برای عملگر – انجام می دهد.

تعریف عملگر \* بین دو ماتریس یا عدد و ماتریس:



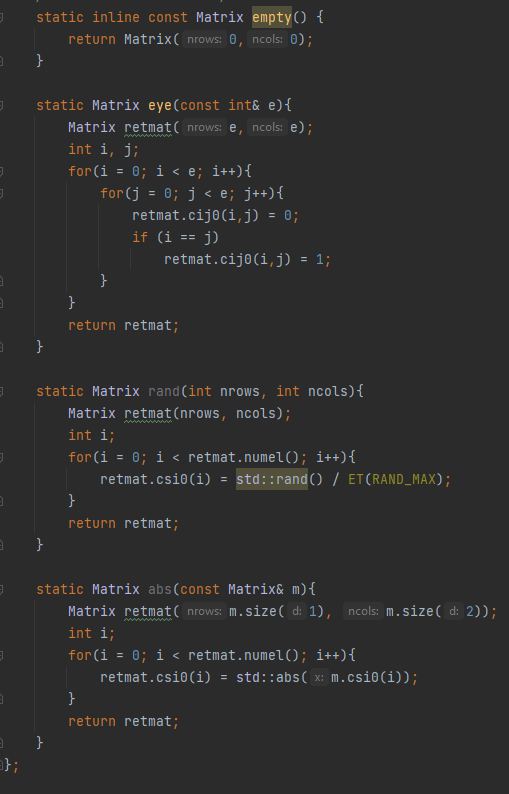


اگر هردو ماتریس بودند، چک می شود که اگر هرکدام از آنها اسکالر بود ابتدا عدد موجود در آن استخراج می شود سپس یک ماتریس جدید هم اندازه با ماتریس داده شده ساخته می شود. سپس تمام اعضای ماتریس داده شده در عدد موجود ضرب شده و نظیر به نظیر در ماتریس جدید قرار داده می شود و در نهایت ماتریس جدید برگردانده می شود. اگر هر دو ماتریس هایی غیر اسکالر بودند،ابتدا یک ماتریس جدید که سطر آن برابر سطرهای ماتریس اول و تعداد ستون های آن برابر تعداد ستون های ماتریس دوم بودند(طبق خواص ضرب ماتریس ها ماتریس حاصلضرب دو ماتریس بدین شکل باید باشد) عملیات ضرب ماتریسی بین دو ماتریس انجام می شود. دقت شود که اگر عدد موجود در سطر iام و ستون jام ماتریس حاصلضرب صفر قرار داده نشود یک عدد نامعلوم و رندوم از حافظه برمیگزیند و بنابر این رفتار برنامه و پاسخ چاپ شده غیرقابل پیش بینی خواهد بود.

اگر هم یک طرف عملگر عدد قرار داشت، عدد مورد نظر به اسکالر تبدیل شده و طبق توضیحات ضرب اسکالر در ماتریس عمل خواهد شد.

متدهای static:

کلمه کلیدی static باعث می شود بتوانیم متدها یا ویژگی های یک کلاس را بدون ساختن آبجکت در دسترس داشته باشیم. در واقع زمانی که کلمه static را همراه متد یا ویژگی کلاس می آوریم، آنها را تبدیل به متد یا ویژگی static کرده ایم.



این متدها به ترتیب:

1. یک ماتریس خالی میسازد.
2. یک عدد صحیح دریافت کرده و یک ماتریس مربعی e\*e همانی می سازد.
3. یک عدد به عنوان سطر ویک عدد به عنوان ستون دریافت کرده و یک ماتریس با این تعداد سطر و وستون و اعداد رندوم می سازد.
4. یک ماتریس دریافت کرده و قدر مطلق تمام اعداد موجود در ماتریس رادر ماتریس جدیدی می ریزد و ماتریس جدید را برمیگرداند.