

# תכנות מונחה עצמים

מטרות התרגיל:

Classes, constructors, copy-constructor, destructors, exceptions, operators overloading

יש לטפל בשגיאות (exception): במקרה של שגיאה יש להחזיר לפונ' הקוראת את ההודעה מסודרת ולהמשיך לפי לוגיקה של התאוששות (לא לסיים את הריצה). מותר להוסיף מתודות. יש להקפיד על כל הכללים שנלמדו (כולל נושא ה-const, &, friend וכדומה). הקפדה יתרה על הוראות ההגשה אשר פורסמו במודל!

הגדרת המערכת

עליכם לתכנת מערכת לניהול מטריצה ב  $R^2$  (מערך דו-ממדי).

מימוש

כתבו מחלקה בשם MyMatrix המכילה פוינטר (דינמי) למערך דו-ממדי של double. בנוסף המחלקה מכילה 2 משתנים int \_m ו int \_n המייצגים את הגודל של המערך.

שיטות (פונקציות):

יש לממש את הפונקציות הבאות:

פונקציה	הסבר
MyMatrix(const MyMatrix & other)	Copy constructor
MyMatrix(int m, int n)	Constructor בונה מערך בגודל N*M כאשר כל הערכים של המערך שווים 0
MyMatrix()	Default constructor בונה מערך בגודל 3*3 כאשר כל הערכים של המערך שווים 0
~MyMatrix()	Destructor
... Operator+(...)	חיבור 2 מטריצות. לדוגמא: MyMatrix res=myMatrix1+myMatrix2;

... Operator-(...)	Operator overloading	חיסור 2 מטריצות. לדוגמא: MyMatrix res=myMatrix1- myMatrix2;
... Operator*(...)	Operator overloading	מכפלה 2 מטריצות. לדוגמא: MyMatrix res=myMatrix1*myMatrix2;
... Operator*(...)	Operator overloading	מכפלה מטריצה בסקלר. לדוגמא: MyMatrix res=3*myMatrix1; או MyMatrix res=myMatrix1*3; שימו לב: יש להגדיר 2 אופרטורים כאן
... Operator=(...)	Operator overloading	השמה של מטריצה למטריצה אחרת. לדוגמא: MyMatrix res; res=myMatrix1;
... Operator==(...)	Operator overloading	השוואה בין 2 מטריצות. לדוגמא: MyMatrix myMatrix1; MyMatrix myMatrix2; bool res=myMatrix1== myMatrix2; או if(myMatrix1== myMatrix2)
... Operator<<(...)	Operator overloading	הדפסת מטריצה: cout<< myMatrix1;
... Operator>>(...)		קליטת מטריצה: cin>>myMatrix1;
... get...(...)		לפי הצורך
... set...(...)		לפי הצורך
... Operator[](...)	Operator overloading	חזרת ערך לפי INDEX. לדוגמא:

		double n=myMatrix1[1][2]; myMatrix1[1][2] = 3.4;
Operator double(...).		אופרטור קסטינג אשר מחזיר את סכום איברי המטריצה. double x = (double)myMatrix1;

חשוב :

- במקרה ואם הפעולה לא חוקית יש לטפל בדבר בעזרת מנגנון של Exception.
- הפעולות מתמטיות צריכות לעמוד בדרישה של יותר מ 2 משתנים בכל פעולה. כלומר :  
$$D=A+B+C+Z$$
- הפונקציות צריכות לעמוד בדרישה של אסוציאטיביות. כלומר :  
$$D=A+B-C=(A+B)-C=...$$
- במידת הצורך יש לממש פונקציות נוספות.
- את ההגדרות מתמטיות המדויקות אפשר לראות באתר הבא :

- [http://he.wikibooks.org/wiki/%D7%90%D7%9C%D7%92%D7%91%D7%A8%D7%94\\_%D7%9C%D7%99%D7%A0%D7%90%D7%A8%D7%99%D7%AA/%D7%97%D7%A9%D7%91%D7%95%D7%9F\\_%D7%9E%D7%98%D7%A8%D7%99%D7%A6%D7%95%D7%AA](http://he.wikibooks.org/wiki/%D7%90%D7%9C%D7%92%D7%91%D7%A8%D7%94_%D7%9C%D7%99%D7%A0%D7%90%D7%A8%D7%99%D7%AA/%D7%97%D7%A9%D7%91%D7%95%D7%9F_%D7%9E%D7%98%D7%A8%D7%99%D7%A6%D7%95%D7%AA)
- [http://en.wikipedia.org/wiki/Matrix\\_multiplication](http://en.wikipedia.org/wiki/Matrix_multiplication)

### ממשק משתמש

יש להכין תכנית ראשית אשר מריצה את כל הפעולות של המחלקה MyMatrix.

בהצלחה!!! ☺



```

1 #include ...
2 using namespace std;
3
4 int main() {
5     {
6         MyMatrix mDefault;//default constructor
7         MyMatrix matrix3X5(3, 5), matrix2X2(2, 2), matrix2X4(2, 4), matrix4X2(4, 2); // ...
8         for (int i = 0; i < mDefault.getM(); i++)
9             for (int j = 0; j < mDefault.getN(); j++)
10                 mDefault.setValue(i, j, 2);
11         MyMatrix mCopy(mDefault), matrix3X3(mDefault); // ...
12         for (int i = 0; i < matrix3X3.getM(); i++)
13             for (int j = 0; j < matrix3X3.getN(); j++)
14                 matrix3X3.setValue(i, j, 3);
15         try {
16             matrix3X3 = matrix3X3 + matrix3X3 + (matrix3X3 - matrix3X3) - matrix3X3; //add and subtract with the same numbers of rows and columns
17             cout << matrix3X3 << endl; //print matrix
18             matrix3X3 = matrix3X5 + mDefault; //EXP6 matrix3X5(3,5) cannot add with mDefault(3,3). exception throw
19         }
20         catch (ExceptionObj e)
21         {
22             e.printError();
23         }
24
25         try {
26             matrix3X3 = 2*matrix3X3 *2; //multiplt using Mymatrix-int * friend *
27             cout << matrix3X3 << endl; //print matrix
28             matrix3X3 = matrix3X3 * matrix3X5; //EXP8 "cannot use = operator. Matrices with different number of rows or columns. "
29         }
30         catch (ExceptionObj e)
31         {
32             e.printError();
33         }
34
35         try {
36             matrix3X3 = matrix3X3 * matrix3X3 * matrix3X3; //operator * for My Mymatrix objects
37             cout << "matrix3X3 :" << endl << matrix3X3 << endl; //print matrix
38             matrix3X3 = (matrix3X5 * matrix3X3) * mDefault; //EXP7 (matrix3X5 * matrix3X3) range is (5,3) cannot multiply with mDefault(3,3). exception throw
39         }
40         catch (ExceptionObj e)
41         {
42             e.printError();
43         }
44
45         try {
46             cin >> matrix2X2; //create matrix using user values
47             cout << "matrix2X2 :" << endl << matrix2X2 << endl; //print matrix
48         }
49         catch (ExceptionObj e)
50         {
51             e.printError();
52         }
53     }
54 }

```

```

*****
[3][3][3]
[3][3][3]
[3][3][3]
*****
#####
Error number : 6
Error description : cannot add or subtract two matrices with range difference (columns/rows)
#####
*****
[12][12][12]
[12][12][12]
[12][12][12]
*****
#####
Error number : 8
Error description : cannot use = operator. Matrices with different number of rows or columns
#####
matrix3X3 :
*****
[15552][15552][15552]
[15552][15552][15552]
[15552][15552][15552]
*****
#####
Error number : 7
Error description : multiply error : first matrix columns doesn't match second matrix rows
#####
Insert values for row number : 1

```



```

41 }
42 catch (ExceptionObj e)
43 {
44     e.printError();
45 }
46 try {
47     cin >> matrix2X2; //create matrix using user values
48     cout << "matrix2X2 : " << endl << matrix2X2 << endl; //print matrix
49 }
50 catch (ExceptionObj e)
51 {
52     e.printError();
53 }
54 int row, column;
55 cout << "enter row and column index in order to get value from matrix using [][] operator " << endl;
56 cin >> row >> column;
57 if (column >= 0 && column < matrix2X2.getN()) //columns in range
58 {
59     try {
60         cout << "use operator [][] on MyMatrix type : " << endl;
61         cout << "matrix2X2 in row : " << row << "and column : " << column << " is : " << matrix2X2[row][column] << endl;
62     }
63     catch (ExceptionObj e) //rows out of range
64     {
65         e.printError();
66     }
67 }
68 else
69     if (row < 0 || row > matrix2X2.getM())
70         cout << "both rows and columns are out of range " << endl;
71     else
72         cout << "columns out of range " << endl;
73 cout << "casting on Matrix : " << (double) matrix2X2 << endl; //casting
74 if (matrix2X2 == matrix2X2) //comparing two matrices
75     cout << "the matrecis are geuals " << endl;
76 return 0;
77 }
78
79

```

```

[15552][15552][15552]
[15552][15552][15552]
*****
#####
Error number : 7
Error description : multiply error : first matrix columns doesnt match s
econd matrix rows
#####
Insert values for row number : 1
77
55
Insert values for row number : 2
3
4
matrix2X2 :
*****
[77][55]
[3][4]
*****
enter row and column index in order to get value from matrix using [][]
operator
66
-3
both rows and columns are out of range
casting on Matrix : 139
the matrecis are geuals

C:\Users\yaniv\source\repos\homeWork2\Debug\homeWork2.exe (process 9712)
exited with code 0.
To automatically close the console when debugging stops, enable Tools->O
ptions->Debugging->Automatically close the console when debugging stops.

Press any key to close this window . . .

```