**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 R2(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:

|  |  |  |
| --- | --- | --- |
| הסבר |  | פונקציה |
|  | Copy constructor | MyMatrix(const MyMatrix & other) |
| בונה מערך בגודל M\*N כאשר כל הערכים של המערך שווים 0 | Constructor | MyMatrix(int m, int n) |
| בונה מערך בגודל 3\*3 כאשר כל הערכים של המערך שווים 0 | Default constructor | MyMatrix() |
|  | Destructor | ~MyMatrix() |
| חיבור 2 מטריצות.  לדוגמא:  MyMatrix res=myMatrix1+myMatrix2; | Operator overloading | … Operator+(…)… |
| חיסור 2 מטריצות.  לדוגמא:  MyMatrix res=myMatrix1-myMatrix2; | Operator overloading | … Operator-(…)… |
| מכפלה 2 מטריצות.  לדוגמא:  MyMatrix res=myMatrix1\*myMatrix2; | Operator overloading | … Operator\*(…)… |
| מכפלה מטריצה בסקלר.  לדוגמא:  MyMatrix res=3\*myMatrix1;  או  MyMatrix res=myMatrix1\*3;  שימו לב: יש להגדיר 2 אופרטורים כאן | Operator overloading | … Operator\*(…)… |
| השמה של מטריצה למטריצה אחרת. לדוגמא:  MyMatrix res;  res=myMatrix1; | Operator overloading | … Operator=(…)… |
| השוואה בין 2 מטריצות.  לדוגמא:  MyMatrix myMatrix1;  MyMatrix myMatrix2;  bool res=myMatrix1== myMatrix2;  או  if(myMatrix1== myMatrix2) | Operator overloading | … Operator==(…)… |
| הדפסת מטריצה:  cout<< myMatrix1; | Operator overloading | … Operator<<(…)… |
| קליטת מטריצה:  cin>>myMatrix1; |  | … Operator>>(…)… |
| לפי הצורך |  | … get…(…)… |
| לפי הצורך |  | … set…(…)… |
| החזרת ערך לפי INDEX.  לדוגמא:  double n=myMatrix1[1][2];  myMatrix1[1][2] = 3.4; | Operator overloading | … Operator[](…)… |
| אופרטור קסטינג אשר מחזיר את סכום איברי המטריצה.  double x = (double)myMatrix1; |  | Operator double(…)… |

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://en.wikipedia.org/wiki/Matrix_multiplication>

**ממשק משתמש**

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

***בהצלחה!!! ☺***