```
1
2  /**
3   * מחלקת שירות לטיפול במערך דו-ממדי
4   * @author Hila Kadman
5   *
6   *      : עמודות N -ו שורות N הגדרת מטריצה בגודל
7   *
8   *      int [][] mat = new int [N,M] ;
9   *
10  */
11
12 public class Matrix {
13
14     /**
15      * פעולה המאפסת את כל איברי המטריצה
16      */
17     public static void InitMat(int [,] m)
18     {
19         for (int i = 0 ; i < m.GetLength(0) ; i++)
20             for (int j = 0 ; j < m.GetLength(1) ; j++)
21                 m[i,j] = 0;
22     }
23
24
25     /**
26      * פעולה הקולטת איברים למטריצה
27      */
28     public static void MatKelet (int [,] m)
29     {
30         for (int i = 0 ; i < m.GetLength(0) ; i++)
31             for (int j = 0 ; j < m.GetLength(1) ; j++)
32             {
33                 Console.Write(j + " : עמודה " + i + " : הקש ערך לאיבר בשורה --> ");
34                 m[i,j] = int.Parse(Console.ReadLine());
35             }
36     }
37
38
39     /**
40      * 1 או 0 פעולה הממלאה מטריצה במספרים אקראיים
41      */
42     public static void MatFill (int [,] m)
43     {
44         Random rnd = new Random ();
45         for (int i = 0 ; i < m.GetLength(0) ; i++)
46             for (int j = 0 ; j < m.GetLength(1) ; j++)
47                 m[i,j] = rnd.Next(2);
48     }
49
50
51     /**
52      * פעולה המציגה את המטריצה כטבלה
53      */
54     public static void MatPelet (int [,] m)
55     {
56
57         for (int i = 0 ; i < m.GetLength(0) ; i++)
58         {
59             Console.Write("Line #" + i + ":");
60             for (int j = 0 ; j < m.GetLength(1) ; j++)
61                 Console.Write("  {0}", m[i,j]);
62             Console.WriteLine();
63         }
64     }
65
66
67
68
```

```
69      /**
70       *            m טענת כניסה: מטריצה
71       *   טענת יציאה: מוצזר סכום המטרצה
72       */
73      public static int MatSum (int [,] m)
74      {
75          int sum = 0;
76          for (int i = 0 ; i < m.GetLength(0) ; i++)
77              for (int j = 0 ; j < m.GetLength(1) ; j++)
78                  sum = sum + m[i,j];
79          return sum;
80      }
81
82
83      /**
84       *   line ושורה m טענת כניסה: מטריצה
85       *      טענת יציאה: מוצזר סכום השורה
86       */
87      public static int SumOfLine (int [,] m, int line)
88      {
89          int sum = 0;
90          for (int i = 0 ; i < m.GetLength(1) ; i++)
91              sum = sum + m[line,i];
92          return sum;
93      }
94
95
96      /**
97       *   col ועמודה  m טענת כניסה:  מטריצה
98       *      col -טענת יציאה: סכום העמודה ה
99       */
100     public static int SumOfColumn (int [,] m, int col)
101     {
102         int sum = 0;
103         for (int i = 0 ; i < m.GetLength(0) ; i++)
104             sum = sum + m[i,col];
105         return sum;
106     }
107
108
109     /**
110      *           m טענת כניסה: מטריצה ריבועית
111      *   טענת יציאה: מוצזר סכום האלכסון הראשי
112      */
113     public static int SumOfFirstDiagonal (int [,] m)
114     {
115         int n = m.GetLength(0), sum = 0;
116         for (int i = 0 ; i < n ; i++)
117             sum = sum + m[i,i];
118         return sum;
119     }
120
121
122     /**
123      *           m טענת כניסה: מטריצה ריבועית
124      *   טענת יציאה: מוצזר סכום האלכסון המשני
125      */
126     public static int SumOfSecondDiagonal (int [,] m)
127     {
128         int n = m.GetLength(0);
129         int j = n-1 , sum = 0;
130         for (int i = 0 ; i < n ; i++)
131         {
132             sum = sum + m[i,j];
133             j-- ;
134         }
135         return sum;
136     }
```

```
137
138
139
140         /**
141          *                    טענת כניסה: מטריצה m בגודל NxN כך ש ת- N אי-זוגי
142          *     טענת יציאה: מוחזר "אמת" אם סכום פינות המטריצה שוות לאיבר האמצעי
143          *                                          ו-"שקר" אחרת.
144          */
145         public static bool MatCorners (int [,] m)
146         {
147             int N = m.GetLength(0);
148             int sum = m[0,0] + m[0,N-1] +
149                       m[N-1,0] + m[N-1,N-1];
150             if (m[N/2,N/2] == sum)
151                 return true;
152             return false;
153         }
154
155
156         /**
157          *                 טענת כניסה: מטריצה m , מספר שורה i , מספר עמודה j    X   X
158          *     טענת יציאה: מוצזר סכום האיברים שנמצאים ב- 4 פינות האיבר הנתון      O
159          *     1 <= i < m.GetLength(0)-1  ,  1 <= j < m.GetLength(1)-1 :הנחה    X   X
160          */
161         public static int SumOfCorners (int [,] m, int i , int j)
162         {
163             int sum = m[i-1,j-1] + m[i-1,j+1] +
164                       m[i+1,j-1] + m[i+1,j+1];
165             return sum;
166         }
167
168
169         /**
170          *               טענת כניסה: מטריצה m , מספר שורה i , מספר עמודה j          X
171          *    טענת יציאה: מוצזר סכום האיברים שנמצאים ב- 4 צידי האיבר הנתון      X  O  X
172          *    1 <= i < m.length-1  ,  1 <= j < m[i].length-1         :הנחה          X
173          */
174         public static int SumOfSides (int [,] m, int i , int j)
175         {
176             int sum = m[i-1,j] + m[i,j-1] + m[i,j+1] + m[i+1,j];
177             return sum;
178         }
179
180
181         /**
182          *               טענת כניסה: מטריצה m , מספר שורה x , מספר עמודה y      X  X  X
183          *       טענת יציאה: מוחזר סכום האיברים שנמצאים מסביב לאיבר הנתון      X  O  X
184          *    1 <= i < m.length-1  ,  1 <= j < m[i].length-1         :הנחה      X  X  X
185          */
186         public static int SumOfNeighbours (int [,] m, int x , int y)
187         {
188             int sum = 0;
189             for (int i = x-1 ; i <= x + 1 ; i++)
190                 for (int j = y-1 ; j <= y + 1 ; j++)
191                     sum = sum + m[i,j];     // הלולאה סיכמה גם את האיבר הנתון
192             return sum - m[x,y];            //     לכן נחסר אותו מהערך המוחזר
193         }
194
195 }
196
197
198
199
```