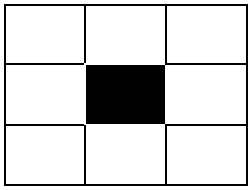
מבוא למדעי המחשב

תרגיל בית מס' 2

<u>הנחיות הגשה:</u>

- *ההגשה תתבצע בקובץ אשר יכיל רק את קבצי java.*
- *ההגשה תתבצע בקובץ אשר שמו יכיל את השם הפרטי של המגיש .*
  למשל: <u>HW1_AviLevi.java</u> כאשר <u>AviLevi</u> הינו שמו של המגיש ו <u>HW1</u>-מציין תרגיל בית 1
  שם המחלקה יהיה בהתאם: <u>public class HW1_AviLevi</u>
- *על התכניות לעבור הידור (קומפילציה) והרצה.*
- *יש להקפיד על הזחה(אינדנטציה) מתאימה בכדי שהקוד יהיה קריא. וכן כול שמות משתנים בעלי משמעות ועל פי הקונבנציה שהוצגה בקורס*
- *הגשת העבודה תתבצע דרך תפריט המטלות שבאתר הקורס*
- ***אין להשתמש בחבילות קוד מלבד אלו שנלמדו בקורס !***
- ***חובה לרשום את פרטי המגישים כהערה בראש כול קובץ!***
- *תאריך הגשה: 29/11/2018 בשעה 23:00. בהצלחה!*

עליכם לכתוב משחק "Game Of Life", אשר מבוסס על לוח בגודל של X × X משבצות (X יוגדר בהמשך). כול משבצת על הלוח מייצגת תא "חי" או "מת". כמו כן, יש להבין שכל שמונת התאים הסמוכים לתא נקראים שכנים של התא. במידה ומסתכלים על התא הראשון, אז השכן השמאלי של התא הינו התא האחרון בלוח ולהיפך והשכן של כול תא בשורה התחתונה ביותר בלוח נמצא בשורה העליונה.



- ✓ **תא חי יכול למות במידה ומתקיימים תנאים הבאים:**
  - **צפיפות – אם יש לו יותר מ- 4 שכנים חיים.**
  - **בדידות – אם יש לו פחות מ- 2 שכנים חיים.**

  כלומר, התא ממשיך לחיות רק במידה ויש לו בין 2 ל-4 שכנים חיים.
- ✓ **תא מת יכול לקום לתחיה במידה ויש לו בדיוק 3 שכנים חיים.**

למטרת המשחק יש לכתוב שיטות הבאות:

- ✓ ***public static boolean[][] moveTimeBy(int numberOfSteps)*** - השיטה מקבלת מספר שלם כפרמטר המייצג מספר צעדים ומחזירה מערך בוליאני דו ממדי המייצג את מצב הלוח לאחר ***numberOfSteps*** צעדים, כאשר תא חי מיוצג ע"י ***true***.

- ✓ ***public static boolean setBoard(boolean[][] board)*** השיטה מקבלת מערך בוליאני דו ממדי כפרמטר שמייצג לוח ומעדכנת את הלוח המשחק. לוח המשחק חייב להיות זהה ללוח ***board***. אם השיטה נכשלת היא מחזירה ***false***.

פרטים:

1. בתחילת התוכנית, המשתמש מכניס שני פרמטרים:
   - גודל הלוח ע"י הכנסת פרמטר ***X (הלוח בגודל X * X).***
   - מספר לוחות מקסימלי עד לסיום התוכנית.

```
********** GAME OF LIFE ON BOARD X * X **********
Enter X Size:5
Enter Max Number Of Boards:5
```

2. עליכם להגריל מצב התחלת של הלוח ולהדפיסו על הקונסול.

```
Start Board:
0 1 1 0 0
0 1 0 1 0
0 1 1 1 0
0 1 0 1 0
1 0 0 0 0
```

3. לאחר מכן, יש להעתיק את הלוח ללוח חדש.

4. כמו כן, עליכם לבדוק את מצב כול תא בלוח החדש ולעדכן אותו בהתאם לדרישות.

5. אחרי העדכון, על התוכנית להציג את המצב החדש של הלוח.

```
Step 1
1 1 1 0 0
1 1 0 1 0
1 1 0 1 1
1 1 0 1 1
1 0 0 0 0
Board Changed.
```

6. התוכנית תיעצר במקרים הבאים:
   - ***אין יותר שינויים במצב התאים בלוח החדש.***
   - ***התוכנית הגיעה למספר מקסימלי של הלוחות.***

```
Step 5
0 1 0 0 0
1 0 0 0 0
0 1 0 0 0
0 0 0 0 0
0 0 0 0 0
Board Changed.
```

7. יש לספור את מספר הלוחות שהתוכנית יצרה.

8. בסיום התוכנית מדפיסה את הלוח האחרון ואת מספר הלוחות שיצרה.

```
Finished!
0 1 0 0 0
1 0 0 0 0
0 1 0 0 0
0 0 0 0 0
0 0 0 0 0
Board Changed 5 Times.
```

<u>הערות:</u>

- *אין צורך לבדוק את תקינות הקלט של המשתמש וניתן להניח שהקלט תקין.*
- *אין צורך להתייחס לצבע של הטקסט וניתן לכתוב הכול בשחור.*
- *יש להשתמש בפונקציות/שיטות לכל פעולה חוזרת (יצירת, בדיקת, הדפסה, השוואת לוח וכד...)*
- *יש לדאוג שתא בצד אחד מכיר את התאים בצד השני. כלומר, השכנים של השורה הראשונה הם גם בשורה האחרונה. כמו כן, השכנים של העמודה הראשונה הם גם בעמודה האחרונה ולהיפך.*

ב ה צ ל ח ה !

```
********** GAME OF LIFE ON BOARD X * X **********
Enter X Size:5
Enter Max Number Of Boards:5

Start Board:
0 1 1 0 0
0 1 0 1 0
0 1 1 1 0
0 1 0 1 0
1 0 0 0 0

Step 1
1 1 1 0 0
1 1 0 1 0
1 1 0 1 1
1 1 0 1 1
1 0 0 0 0
Board Changed.

Step 2
1 0 1 0 0
0 0 0 1 0
0 0 0 1 0
0 1 0 1 0
0 0 0 1 0
Board Changed.

Step 3
0 0 1 1 1
0 0 1 1 1
0 0 0 1 1
0 0 0 1 1
0 1 0 1 1
Board Changed.
```

```
Step 4
0 1 0 0 0
1 0 1 0 0
1 0 0 0 0
0 0 0 0 0
0 0 0 0 0
Board Changed.

Step 5
0 1 0 0 0
1 0 0 0 0
0 1 0 0 0
0 0 0 0 0
0 0 0 0 0
Board Changed.

Finished!
0 1 0 0 0
1 0 0 0 0
0 1 0 0 0
0 0 0 0 0
0 0 0 0 0
Board Changed 5 Times.
```