

# מטלת מנחה (ממ"ן) 13

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

נושא המטלה: לולאות ומערכים

חומר הלימוד למטלה: יחידות 5 – 6

משקל המטלה: 5 נקודות

מספר השאלות: 2

מועד אחרון להגשה: 20.4.2019

סמסטר: 2019

(ת)

## חלק א – מערך חד-ממדי

בחלק הראשון של מטלה זו אנו משתמשים במחלקות Point3D ו-Box3D שכתבנו בממ"ן 12.

אתם יכולים להשתמש במחלקות Point3D ו-Box3D שכתבתם, או בקבצים Point3D.class ו-Box3D.class שאנו שמנו באתר בספר הדיגיטלי של יחידות 5 - 6, בפרק של מטלה 13. אנא קראו את הכתוב במדריך creating\_a\_project\_and\_using\_existing\_classes שנמצא בלשונית "מדריכי עזר" במשאבי הלמידה בדף הבית של האתר הקורס. כך תדעו איך להשתמש במחלקה שכבר כתובה, וניתנת לכם כקובץ class ללא הקוד. שמנו באתר טסטר בסיסי לבדיקה ראשונית של המטלה. חובה להריץ את המטלה מול הטסטר ולבדוק שאין טעויות קומפילציה.

## שאלה 1 - להרצה (70%)

המחלקה Collection מכילה אוסף של תיבות.

הייצוג נעשה על-ידי מערך ששומר את רשימת התיבות. התכונות במחלקה הן:

- מערך של תיבות `Box3D [] _boxes`
- מספר התיבות במערך `int _noOfBoxes`

כמו כן קיים קבוע מספרי MAX\_NUM\_BOXES המציין את המספר המקסימלי של התיבות במערך – 100.

התיבות (כלומר האובייקטים מהמחלקה Box3D) נמצאים במערך ברצף, ללא "חורים" מתחילת המערך. המערך צריך להישאר כך (ללא חורים) לאחר כל פעולה. התיבות מסודרות במערך לפי גודלן (בנפח), מהקטנה ביותר עד לגדולה ביותר.

## עליכם לממש ב-Java את המחלקה Collection לפי הסעיפים להלן:

1. הגדרת התכונות של המחלקה.
2. בנאי שמאתחל את תכונות המחלקה כך שהמערך יהיה בגודל מקסימלי.
3. שיטה בוליאנית addBox שמוסיפה תיבה למערך. היא מקבלת תכונות של תיבה (נקודה בסיסית, אורך, רוחב וגובה) ומכניסה את התיבה למערך למקום המתאים לה לפי סדר התיבות. השיטה מחזירה true בהצלחה (יש מקום במערך להוסיף תיבה) ו-false בכישלון. יתכן שתהיה כבר תיבה במערך באותם ממדים. זה לא מפריע. במקרה זה התיבה החדשה צריכה להיכנס במערך **לפני** התיבה שבאותם ממדים.
4. שיטה mostUpperBaseCorner המחזירה את התיבה שהפינה הבסיסית שלה היא הכי גבוהה במערך. אם המערך ריק יוחזר null. אם יש כמה תיבות כאלו, תוחזר התיבה ה**ראשונה** ביניהן.
5. שיטה totalSurfaceArea המחזירה את שטח הפנים הכולל של כל התיבות במערך.
6. שיטה longestDistance המחזירה מה המרחק הגדול ביותר בין שתי תיבות במערך (כלומר בין שתי הנקודות המרכזיות של התיבות). אם מספר התיבות במערך קטן מ-2, יוחזר 0.
7. שיטה howManyContains המקבלת תיבה כפרמטר ומחזירה מספר המציין כמה תיבות במערך יכולות להכיל את התיבה שהתקבלה בפרמטר.
8. שיטה volumeOfSmallestBox המקבלת שני אינדקסים i ו-j במערך, ומחזירה את הנפח של התיבה הקטנה ביותר (**בנפחה**) שתוכל להכיל את התיבות שנמצאות במערך בין אינדקסים i ו-j (**טווח האינדקסים כולל את i ו-j עצמם**). **מדובר בהכלה** של כל תיבה **בנפרד**, לא ביחד.
- שימו לב** שאי אפשר להניח כלום על i ו-j. אם הם לא מייצגים תת-מערך חוקי (חשבו מתי זה קורה) יוחזר 0.
9. שיטה getBoxes המחזירה מערך של תיבות בגודל מספר התיבות **בפועל** שבמערך, ובו עותק של כל אחת מהתיבות במערך.
10. שיטה getNumOfBoxes המחזירה את מספר התיבות במערך.
11. כתבו את השיטה toString המחזירה מחרוזת המכילה את המידע על כל התיבות במערך. שימו לב להשתמש בשיטות שכבר קיימות. מידע על כל תיבה יתחיל בשורה חדשה.

## לדוגמא,

Box no. 1: The base point is (3.0,4.0,5.0), length = 10, width = 4, height = 6

Box no. 2: The base point is (4.0,2.0,5.0), length = 5, width = 11, height = 6

Box no. 3: The base point is (0.0,0.0,0.0), length = 5, width = 8, height = 10

## שימו לב לא לבצע aliasing במקומות המועדים.

מותר להוסיף שיטות נוספות (פרטיות), לפי ראות עיניכם. אסור להוסיף תכונות.

אתם צריכים לכתוב בעצמכם API למחלקה, לבנאים ולשיטות לפי הנהוג בכתיבת API. כמו כן, עליכם לתעד בתיעוד פנימי כל מה שדורש הבהרה ואינו פשוט.

המימוש אשר תכתבו צריך להיות בהתאם ל-API אשר נמצא כאן להלן. את הערות ה-API אתם צריכים לכתוב בעצמכם.

Constructor Summary	
<u>Collection</u> ( )	
Method Summary	
boolean	<u>addBox</u> (Point3D base, int length, int width, int height)
Box3D[ ]	<u>getBoxes</u> ( )
int	<u>getNumOfBoxes</u> ( )
int	<u>howManyContains</u> (Box3D box)
double	<u>longestDistance</u> ( )
Box3D	<u>mostUpperBaseCorner</u> ( )
java.lang.String	<u>toString</u> ( )
double	<u>totalSurfaceArea</u> ( )
int	<u>volumeOfSmallestBox</u> (int i, int j)

## חלק ב – מערך דו-ממדי

### שאלה 2 - להרצה (30%)

המחלקה Matrix מייצגת **מטריצה** = מערך דו-ממדי של מספרים שלמים בטווח 0-255. המטריצה מייצגת תמונה דו-ממדית בשחור-לבן, כאשר המספרים מייצגים את גווני האפור בתמונה 0 הוא לבן ו-255 הוא שחור. עליכם לממש את המחלקה לפי הפירוט הבא:

תכונות – מותרת רק **תכונה אחת** שמייצגת את המערך. מותר להוסיף קבועים כפי רצונכם.

- בנאים – עליכם לממש שני בנאים לפי הממשק הבא:

Constructor Summary	
<b>Matrix</b> (int[] array)	Constructs a Matrix from a two-dimensional array; the dimensions as well as the values of this Matrix will be the same as the dimensions and values of the two-dimensional array.
<b>Matrix</b> (int size1, int size2)	Constructs a size1 by size2 Matrix of zeroes.

size1 מייצג את מספר השורות ו-size2 מייצג את מספר העמודות. ניתן להניח כי הפרמטר array המועבר לבנאי הראשון מכיל מטריצה בגודל 1 על 1 לפחות, והערכים במטריצה זו הם בטווח 0-255. ניתן להניח כי הפרמטרים size1 ו-size2 המועברים לבנאי השני הם חיוביים. שימו לב לא לבצע aliasing בבנאי שמקבל מערך.

- ממשו שיטת toString() למחלקה Matrix. אלמנטים באותה שורה יופרדו על-ידי tab (המחרוזת "\t"), ושורות יופרדו על-ידי מעבר שורה.

**לדוגמה:**

אם המטריצה היא זו:

	0	1	2
0	3	8	72
1	4	6	60
2	253	2	1

המחרוזת שתוחזר מהשיטה toString() שתופעל על המטריצה היא :

"3"+"\t"+"8"+"\t"+"72"+"\n"+"4"+"\t"+"6"+"\t"+"60"+"\n"+"253"+"\t"+"2"+"\t"+"1"+"\n"

כלומר, המחרוזת: "3\t8\t72\n4\t6\t60\n253\t2\t1\n"

ולכן היא תודפס כך :

```
3      8      72
4      6      60
253    2      1
```

שימו לב לא להוסיף tab בסוף השורות

• הוסיפו למחלקה Matrix את השיטה public Matrix makeNegative()

המחזירה את תמונת הנגיב של התמונה. כלומר, כל נקודה בתמונה (תא במערך) שהיה שחור הופך להיות לבן, כל תא שהיה לבן הופך להיות שחור, וכך גם כל גווני האפור מתהפכים (למספר המשלים אותם ל-255). ניתן להניח שהמטריצה מכילה מספרים בטווח 0-255.

יש להחזיר מטריצה חדשה ואין לשנות את המטריצה המקורית.

לדוגמה, אם התמונה המקורית היא זו :

	0	1	2	3	4
0	19	124	28	35	38
1	115	22	25	230	31
2	9	21	22	249	230
3	0	6	9	232	255
4	2	5	10	116	129

אז תמונת הנגיב, שצריכה להיות מוחזרת מהשיטה היא :

	0	1	2	3	4
0	236	131	227	220	217
1	140	233	230	25	224
2	246	234	233	6	25
3	255	249	246	23	0
4	253	250	245	139	126

- הוסיפו למחלקה Matrix את השיטה `public Matrix imageFilterAverage()` המחזירה את התמונה שמתקבלת מהחלקת התמונה. פעולת ההחלקה (שנקראת לפעמים גם הסרת רעש מתמונה או טשטוש), נעשית על-ידי כך שכל תא בתמונה הנוכחית הופך להיות בתמונה החדשה הערך של הממוצע שלו עם שכניו (בתמונה הנוכחית). אנחנו נתייחס לשמונת שכניו הצמודים אליו. שימו לב שלפעמים לתא אין שמונה שכנים, אלא רק חמישה או אפילו רק שלושה (מתי?) שימו לב שתוצאת הממוצע צריכה גם היא להיות מטיפוס `int`. לדוגמא, אם התמונה המקורית היא זו:

	0	1	2	3	4
0	19	124	28	35	38
1	115	22	25	230	31
2	9	21	22	249	230
3	0	6	9	232	255
4	2	5	10	116	129

אז התמונה המוחלקת, שצריכה להיות מוחזרת מהשיטה היא:

	0	1	2	3	4
0	70	55	77	64	83
1	51	42	84	98	135
2	28	25	90	142	204
3	7	9	74	139	201
4	3	5	63	125	183

- הוסיפו למחלקה Matrix את השיטה `public Matrix rotateClockwise()` המסובבת את התמונה ב- 90 מעלות עם כיוון השעון. שימו לב שהסיבוב יכול לשנות את מימדי התמונה.

לדוגמה, אם התמונה המקורית היא זו :

	0	1	2	3
0	19	124	28	35
1	115	22	25	230
2	19	21	22	249
3	0	16	9	232
4	62	35	10	116

אז התמונה לאחר השיטה היא :

	0	1	2	3	4
0	62	0	19	115	19
1	35	16	21	22	124
2	10	9	22	25	28
3	116	232	249	230	35

- הוסיפו למחלקה את השיטה `public Matrix rotateCounterClockwise()` המסובבת את התמונה ב- 90 מעלות נגד כיוון השעון. שימו לב שהסיבוב יכול לשנות את מימדי התמונה.

לצורך מימוש שיטות אלו, **מותר** לממש שיטות נוספות (פרטיות), לפי ראיות עיניכם. יש להחזיר מטריצה חדשה ואין לשנות את המטריצה המקורית.

ניתן להניח שהמטריצה מכילה מספרים בטווח 0-255. דאגו לכך שהקוד יהיה ברור וקריא, וכרגיל, מתועד על-פי כללי javadoc ותיעוד פנימי.

### שימו לב,

באתר הקורס תמצאו גם טסטר לבדיקת האיות והפרמטרים של השמות של השיטות והמחלקות שאתם צריכים לכתוב. חובה עליכם לבדוק את המחלקות שכתבתם בטסטר זה, ולהגיש אותן רק אם הטסטר עובר קומפילציה. שימו לב שהטסטר לא מכסה את כל האפשרויות, ובפרט לא את מקרי הקצה. הוא רק בודק

את השמות של השיטות במחלקות כלומר שגיאות קומפילציה. מאד מומלץ להוסיף לו בדיקות.

### **הגשה**

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו לתעד בתיעוד פנימי וב-API את כל השיטות שיש במחלקות השונות.
3. הקפידו ששמות השיטות יהיו בדיוק כפי שכתוב במטלה. וכן שההדפסות יהיו בדיוק כפי שמופיע במטלה.
4. עליכם להגיש את הקבצים Collection.java, Matrix.java עטפו אותם בקובץ zip ושלחו. אין לשלוח קבצים נוספים.

### **בהצלחה**