

מטלת מנחה (ממ"ן) 14

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

נושא המטלה: רשימות מקושרות

חומר הלימוד למטלה: יחידה 11

משקל המטלה: 3 נקודות

מספר השאלות: 4

מועד אחרון להגשה: 21.6.2025

סמסטר: 2025ב

ביחידה 6 למדנו על מערכים דו-ממדיים (מטריצות). במטלה זו נעסוק ברשימות מקושרות של מספרים שלמים, כאשר נייצג מטריצה בשימוש ברשימות מקושרות. הרשימות שלנו ייצגו אך ורק מטריצות מלבניות (כלומר, שבכל השורות יש אותו מספר עמודות). לא בהכרח מספר השורות שווה למספר העמודות.

כל תא במטריצה ייוצג על ידי חוליה שיש לה ארבעה מצביעים. אחד לכל "שכן" של התא (למעלה, למטה, מימין ומשמאל). לשם כך מצורף לממ"ן זה קובץ המחלקה `IntNodeMat` המייצגת חוליה ברשימה שהיא תא במערך. המערך ייוצג על ידי רשימה של חוליות כאלו. המחלקה המייצגת את המערך, מצורפת גם היא למטלה זו ונקראת `MatrixList` המייצגת רשימה מקושרת כזו.

פתחו את קובץ המחלקה `IntNodeMat` ועברו עליו. בראש הקובץ נמצאות הגדרות התכונות של המחלקה וכן כמה בנאים ושיטות עזר בהן תוכלו להסתייע בעבודתכם.

פתחו את קובץ המחלקה `MatrixList` ועברו עליו. זהו הקובץ בתוכו תממשו את הפתרון שלכם לממ"ן זה. בראש הקובץ נמצאות הגדרות התכונות של המחלקה וכן כמה בנאים ושיטות עזר בהן תוכלו להסתייע בעבודתכם.

הורידו את שתי המחלקות הללו אל המחשב שלכם ועבדו על הקובץ `MatrixList.java`. אל תמחקו ואל תשנו דבר בקבצים שהעתקתם, רק הוסיפו לקובץ `MatrixList.java` (ורק לו) את הקוד שלכם!

הערות חשובות לגבי כל השאלות בממ"ן זה:

- אסור לכם להעביר את תוכן הרשימה למערך ואז לעבוד על המערך. כך גם לא להעביר את תוכן הרשימה לרשימה אחרת ועבודה עליה. פתרון בגישה כזאת יוביל להורדה של כמעט כל הנקודות על השאלה.
- חוץ מאשר בכתיבת הבנאי של המחלקה, אסור לכם להעביר את תוכן הרשימה למערך ואז לעבוד על המערך. כך גם לא להעביר את תוכן הרשימה לרשימה אחרת ועבודה עליה. פתרון בגישה כזאת יוביל להורדה של כמעט כל הנקודות על השאלה.
- מותר לשיטות שאתם כותבים לשנות את תוכן או את מבנה הרשימה עליה אתם עובדים. עם זאת כאשר השיטה מסיימת את עבודתה, המבנה ותוכן הרשימה חייבים להיות בדיוק כמו שהיו לפני הקריאה לשיטה.
- אסור להוסיף תכונות למחלקות IntNodeMat ו-MatrixList. קבועים מותר להוסיף.
- אפשר להניח שהפרמטר שמתקבל לכל אחת מהשיטות אינו null. אי אפשר להניח מעבר לכך כל הנחה שהיא לגבי גודל הרשימה. כלומר, יכול להיות שהרשימה תהיה ריקה (כלומר head_ יצביע על null).
- כתבו באנגלית API לכל השיטות הציבוריות שלכם. הוסיפו תיעוד רגיל (באנגלית) בתוך השיטות.
- ניתן להשתמש בשיטות עזר פרטיות ככל הנדרש. לשיטות אלו עליכם לכתוב תיעוד רגיל בלבד, לא API, הן לפני כל שיטה (מה היא עושה) והן בתוכה.
- אין להשתמש בשום מחלקה בג'אווה ובשום מרכיבים של השפה שלא נלמדו במהלך הקורס. אי עמידה בכלל זה יוביל להורדה משמעותית ביותר של נקודות.

למחלקה MatrixList יש תכונה אחת – מצביע על החוליה הראשונה במטריצה (שורה 0 ועמודה 0).
היא מטיפוס IntNodeMat ונקראת _m00.

```
IntNodeMat _m00;
```

אסור להוסיף תכונות למחלקה. התכונה היחידה היא המצביע _m00

שאלה 1 - 20 נקודות

במחלקה MatrixList יש שני בנאים :

1. בנאי ריק – היוצר אובייקט שמצביע על מטריצה ריקה (החוליה הראשונה היא null).

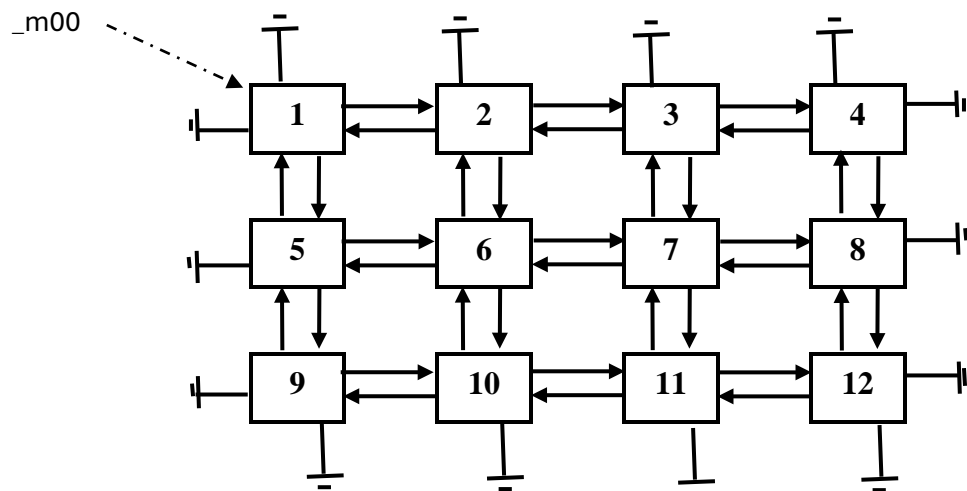
```
public MatrixList()
{
    _m00= null;
}
```

2. בנאי המקבל מטריצה mat (מערך דו ממדי של מספרים שלמים) ויוצר רשימה המייצגת את המטריצה mat.

לדוגמא, אם המטריצה היא :

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12

המטריצה שתיווצר מהבנאי תהיה :



חתימת הבנאי :

```
public MatrixList(int[][]mat)
{
    // את זה עליכם להשלים
}
```

שימו לב לדייק בין השורות והעמודות ולא להפוך ביניהן. כך למשל, במטריצה וברשימה לעיל יש 3 שורות ו- 4 עמודות.

שאלה 2 - 30 נקודות

הוסיפו למחלקה MatrixList את השיטות הבאות:

א. `toString` – המחזירה ייצוג של המטריצה בצורת מחרוזת כאשר בין שורה לשורה במטריצה יש ירידת שורה " \n " (לאחר השורה האחרונה יש ירידת שורה), ובין עמודה לעמודה במטריצה יש טאב " \t " (לאחר העמודה האחרונה אין טאב).

לדוגמא, אם המטריצה היא זו שלעיל, המחרוזת שתוחזר צריכה להראות כך (בדיוק!):

1	2	3	4
5	6	7	8
9	10	11	12

שימו לב לדייק במחרוזת לפי הכתוב כאן. ללא רווחים נוספים וללא תווים נוספים.

אם המטריצה ריקה, המחרוזת שהשיטה תחזיר צריכה להיות המחרוזת הריקה: "".

חתימת השיטה:

```
public String toString()
```

אסור להשתמש במחלקה `StringBuilder`.

ב. השיטה `getDataIJ`, המקבלת כפרמטרים משתנים `i` ו-`j` המייצגים את השורה והעמודה (בהתאמה), ומחזירה את הערך שיש בתא ה-`[i][j]` במטריצה.

אם אחד מהפרמטרים לא תקין (לדוגמא, יש 3 שורות במטריצה, ו-`i = 4`), השיטה תחזיר את הערך `Integer.MIN_VALUE` (המספר השלם הקטן ביותר האפשרי).

חתימת השיטה:

```
public int getDataIJ (int i, int j)
```

ג. השיטה `setDataIJ`, המקבלת כפרמטרים משתנה שלם `num` ומשתנים `i` ו-`j` המייצגים את השורה והעמודה (בהתאמה), ושמה את הערך `num` בתא ה-`[i][j]` במטריצה.

אם אחד מהפרמטרים לא תקין (לדוגמא, יש 3 שורות במטריצה, ו-`i = 4`), השיטה לא תבצע כלום.

חתימת השיטה:

```
public void setDataIJ (int num, int i, int j)
```

שאלה 3 - 25 נקודות

הוסיפו למחלקה MatrixList את השיטה הבאה :

כתבו שיטה **רקורסיבית בוליאנית** בשם **isDescending** המחזירה true אם כל השורות במטריצה ממוינות (כל אחת מהן) בסדר יורד ממש (ללא מספרים שווים באותה שורה), וכל העמודות במטריצה ממוינות (כל אחת מהן) בסדר יורד ממש (ללא מספרים שווים באותה עמודה). אם התנאי לא מתקיים, השיטה תחזיר false.

חתימת השיטה :

```
public boolean isDescending()
```

- השיטה שתכתבו צריכה להיות ללא כל שימוש בלולאות מכל סוג שהוא.
- מותר לכם לכתוב שיטות עזר, אך גם הן חייבות להיות ללא כל שימוש בלולאות מכל סוג שהוא.
- כל שימוש בלולאה בתשובה יוביל להורדה משמעותית ביותר של נקודות.
- אפשר להשתמש בהעמסת-יתר (overloading).
- אין צורך לדאוג ליעילות השיטה הרקורסיבית, ואין צורך לציין מה סיבוכיות הזמן שלה, אבל כמובן שצריך לשים לב לא לעשות קריאות רקורסיביות מיותרות!

שאלה 4 - 25 נקודות

בשאלה זו אנו מניחים שהמטריצה ממוינת בצורה הבאה :

- כל שורה במטריצה ממוינת בסדר יורד ממש (ללא מספרים שווים באותה שורה)
- כל עמודה במטריצה ממוינת בסדר יורד ממש (ללא מספרים שווים באותה עמודה)

כתבו שיטה **יעילה** בשם **howManyNegative**, המחזירה כמה מספרים שליליים יש במטריצה עליה מופעלת השיטה.

חתימת השיטה :

```
public int howManyNegative()
```

השיטה שתכתבו צריכה להיות יעילה ככל הניתן, גם מבחינת סיבוכיות הזמן וגם מבחינת סיבוכיות המקום. תשובה שאינה יעילה מספיק כלומר, שתהיה בסיבוכיות גדולה יותר מזו הנדרשת לפתרון הבעיה תקבל מעט נקודות בלבד.

מה סיבוכיות זמן הריצה וסיבוכיות המקום של השיטה שכתבתם? הסבירו תשובתכם.

הגשה:

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו ששמות המחלקות והשיטות יהיו בדיוק כפי שמוגדר בממ"ן. **אחרת המחלקה לא תעבור קומפילציה עם הטסטר שלנו והציון יהיה 0.**
3. עליכם להריץ את הטסטר שנמצא באתר הקורס על המחלקה שכתבתם. שימו לב שהטסטר לא מכסה את כל האפשרויות, ובפרט לא את מקרי הקצה. הוא רק בודק את השמות של השיטות במחלקה. מאד מומלץ להוסיף לו בדיקות. שימו לב שאם הטסטר לא יעבור קומפילציה מול המחלקה שכתבתם, הציון על המטלה יהיה אפס. אם יש שיטה שאתם מעוניינים לדלג עליה, עליכם לרשום את חתימת השיטה ולהחזיר ערך סתמי על מנת שהטסטר יעבור קומפילציה.
4. **גם במטלה זו – אם הוספתם הדפסות שלא ביקשנו בשיטות שכתבתם, כדי להיעזר בהן בפתרון השאלה, עליכם למחוק הדפסות אלו לפני ההגשה. הדפסות מיותרות כאלו יורידו בניקוד.**
5. **אסור להשתמש במחלקה `StringBuilder`, או במחלקות מובנות אחרות, פרט לאלו שאנחנו למדנו בקורס, ורק בשיטות שלהם שלמדנו (בהרצאות הקורס!).**
6. את התשובות לכל השאלות יש לכתוב בקובץ `Java` ששמו `MatrixList` שאנחנו שמנו **באתר**. אין להוסיף אף קובץ אחר!
7. שימו לב שהקובץ שאתם שולחים חייב להיות אותו הקובץ שאנחנו שמנו באתר, רק עם התוספות שלכם.
8. שימו לב שהתכונה `_m00` בקובץ `MatrixList` לא הוגדרה ב-`private`. זאת לא טעות אלא נועד להקל על בדיקת הממ"ן שלכם. כלומר, באופן יוצא דופן (ובניגוד למה שעליכם לעשות בכל הקשר אחר) בראש המחלקה שלכם צריך להופיע בדיוק כך :

`MatrixList _m00;`

ולא

`private MatrixList _m00;`
9. **אין להגיש את קובץ ה-API שנוצר מה-API שכתבתם, וגם לא את הקובץ `IntNodeMat`.**
10. **ארזו את קובץ ה-java ששמו `MatrixList` בקובץ `zip` יחיד ושלחו אותו בלבד.**