

מטלת מנחה (ממ"ן) 13

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 5 – 6 נושא המטלה: לולאות ומערכים

מספר השאלות: 4 משקל המטלה: 5 נקודות

סמסטר: 2022 א מועד אחרון להגשה: 11.12.2021

במטלה זו נשחק קצת במשחק סודוקו **Sudoku**. לשם כך נגדיר 2 מחלקות: **Square3x3** ו- **Sudoku**.

שאלה 1 (25%)

כתבו את המחלקה **Square3x3** המייצגת מערך דו-ממדי ריבועי של מספרים שלמים בגודל 3×3 . למחלקה הממשק הבא (בקיצור):

הבנאים:

Constructor Summary	
<u>Square3x3()</u>	Constructor for objects of class Square3X3 . Constructs and initializes a 2-dimensional array of the size 3X3 , with the values of -1 in each cell.
<u>Square3x3(int[][] array)</u>	Constructs a 2-dimensional array of the size 3X3 , whose values are taken from the given array. If the given array's size is bigger than 3X3 , only the first 3X3 cells are taken. If the given array is smaller, the rest of the cells are initialized to -1 . Note that the given array may be non-symmetrical, or may even have rows of different lengths (!!). Make sure to initialize cells to -1 if and only if the corresponding cell in the given array does not exist. You may assume the array is NOT null.
<u>Square3x3(Square3x3 other)</u>	Copy constructor. Constructs a 2-dimensional array of the size 3X3 , whose values are taken from the array of the given Square3x3 object.

השיטות שעליכם לממש במחלקה Square3X3 הן:

Method Summary										
int	<u>getCell</u> (int row, int col) Returns the value in the (row, col) cell. If the row and/or col are out of the array bounds, returns -1. Legal values for row/col are 0,1,2.									
void	<u>setXY</u> (int row, int col, int value) Sets the cell (row, col) in the array to the given value. If the row and/or col are out of the array bounds – does nothing. Legal values for row/col are 0,1,2.									
String	<u>toString</u> () Returns a String representation of the array. For example: The square <table border="1"><tr><td>4</td><td>2</td><td>4</td></tr><tr><td>8</td><td>3</td><td>5</td></tr><tr><td>7</td><td>6</td><td>2</td></tr></table> will be represented like this: 4 2 4 8 3 5 7 6 2	4	2	4	8	3	5	7	6	2
4	2	4								
8	3	5								
7	6	2								

שימו לב! בשיטה toString יש להדפיס את הספרות בדיוק (!!) באופן המצוין. בכל שורה, יופיעו שלוש ספרות, כאשר כל שתי ספרות יופרדו באמצעות טאב ("t").

כלומר, בדוגמה הנתונה, שלוש השורות יודפסו באמצעות שלוש המחרוזות הבאות:

"4\t2\t4\n"

"8\t3\t5\n"

"7\t6\t2\n"

למען הסר ספק, אין להוסיף סימני פיסוק נוספים!

למחלקה זו יש תכונה פרטית אחת בלבד – מערך דו מימדי של שלמים בגודל 3x3.

שאלה 2 (25%)

הוסיפו למחלקה Square3x3 את השיטות הבאות, וממשו אותן :

- שיטה בוליאנית בשם allThere. שיטה זו תבדוק האם במערך של המחלקה נמצאים כל המספרים מ-1 עד 9. היא תחזיר true אם כן, ו-false אחרת. חתימת השיטה היא :

```
public boolean allThere()
```

- שיטה בשם whosThereRow וחתימתה היא :

```
public void whosThereRow (int row, boolean[] values )
```

אופן פעולת השיטה: השיטה מקבלת מספר שורה (מבין הערכים 0,1,2) ומערך בוליאני בגודל 10. עבור כל מספר בין 1-9 המופיע בשורה המתאימה במערך הדו-ממדי של המחלקה השיטה תציב ערך true בתא המתאים במערך values. שאר התאים במערך ישארו ללא שינוי.

לדוגמה, אם המערך values נשלח לשיטה זאת עם התוכן הבא :

```
{ false, true, true, false, false, true, false, false, false, true }
```

ובשורה row מופיעות הספרות 7 8 ו-9, אזי תוכן המערך values לאחר החזרה מהשיטה יהיה

```
{ false, true, true, false, false, true, false, true, true, true }
```

כלומר מקומות 7 8 9 במערך הם עתה true, והאחרים לא השתנו.

~~אם באחד התאים בשורה המתאימה הופיע המספר 7, השיטה תציב true בתא בעל אינדקס 7 במערך. שימו לב שהתא בעל האינדקס 0 במערך values אינו בשימוש.~~

~~השיטה תקבל מספר שורה ומערך בוליאני שכל תאיו מאותחלים ל-false. עבור כל מספר שמופיע בשורה row הנתונה, השיטה תציב true בתא המתאים במערך values. (רמז – ניתן להגדיר את המערך values בגודל 10, ולהתייחס רק לתאים 1 עד 9).~~

- שיטה בשם whosThereCol וחתימתה היא :

```
public void whosThereCol (int col, boolean[] values )
```

השיטה עובדת כמו השיטה הקודמת, רק שהיא עושה את הבדיקה עבור עמודה נתונה, ולא עבור שורה.

שאלה 3 (20%)

כתבו מחלקה בשם Sudoku.

מחלקה זו תייצג לוח של המשחק סודוקו כמערך דו-ממדי ריבועי בגודל 3×3 של אובייקטים מסוג Square3x3. שימו לב, הכוונה היא לא לבנות מערך של int בגודל 9×9 , אלא מערך בגודל של 3×3 שכל אחד מאיבריו הוא אובייקט של המחלקה Square3x3 ! **מערך דו-ממדי זה הוא התכונה היחידה של המחלקה.**

למחלקה יהיה 2 בנאים: בנאי ריק שיאתחל את כל הלוח ל 1-, ובנאי שיקבל כפרמטר מערך 3×3 של אובייקטים של המחלקה Square3x3 וייצר על פיו את הלוח.

למחלקה יהיו שני בנאים :

1. בנאי ריק שיאתחל את לוח הסודוקו כך שהוא יורכב מאובייקטים של המחלקה Square3x3 שמלאים כולם בערכים 1- (כפי שייצר אותם הבנאי הריק של המחלקה Square3x3).
2. בנאי שיקבל כפרמטר מערך 3×3 של אובייקטים של המחלקה Square3x3, ויעתיק את הערכים שבאובייקטים הללו אל לוח הסודוקו החדש.

חתימות הבנאים :

```
public Sudoku()
```

```
public Sudoku(Square3x3[][] square3x3Array)
```

שאלה 4 (30%)

ממשו את השיטה isValid של המחלקה Sudoku. שיטה זו תבדוק האם לוח הסודוקו הוא חוקי. אם כן – תחזיר true. אם הלוח לא חוקי, אין צורך להדפיס למה הוא לא חוקי – אלא להחזיר false בלבד.

חתימת השיטה:

```
public Boolean isValid()
```

לצורך מימוש השיטה יש להשתמש בשיטות שכתבתם בשאלה 3. מומלץ גם לכתוב שיטות עזר פרטיות שישמשו אתכם.

תזכורת – לוח סודוקו הוא חוקי אם בכל שורה מופיעים כל המספרים מ-1 עד 9, וגם בכל עמודה מופיעים כל המספרים מ-1 עד 9, וגם בכל אחד מתשעת הריבועים המרכיבים את הלוח מופיעים כל המספרים מ-1 עד 9. לדוגמא, הלוח הבא אינו חוקי:

1	9	8	9	2	3	4	5	7
7	3	2	6	5	4	1	8	9
5	6	4	8	1	7	2	3	6
3	1	9	2	8	5	7	6	4
8	4	7	1	3	6	5	9	2
2	5	6	4	7	9	8	1	3
4	8	5	3	6	2	9	7	1
6	2	1	7	9	8	3	4	5
9	7	3	5	4	1	6	2	8

כיוון שבשורה הראשונה שמסומנת המספר 9 מופיע פעמיים, ובשורה השנייה המסומנת, המספר 6 מופיע פעמיים. השיטה תחזיר שהלוח אינו חוקי כיוון שיש בשורה הראשונה מספר שמופיע פעמיים (אין צורך "לגלות" את כל הטעויות בלוח, מספיק למצוא את הראשונה ~~בלתי~~ ולהחזיר false).

אם נחליף בין התא השני בשורה הראשונה (9) עם התא השני בשורה השלישית (6) נקבל לוח חוקי (אתם מוזמנים לבדוק).

שימו לב לא לבצע aliasing במקומות המועדים.
מותר להוסיף שיטות נוספות (פרטיות), לפי ראות עיניכם.
אתם צריכים לכתוב בעצמכם API למחלקה, לבנאים ולשיטות לפי הנהוג בכתיבת API. כמו כן, עליכם לתעד בתיעוד פנימי כל מה שדורש הבהרה ואינו פשוט.

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 Square3x3.java, Sudoku.java עטפו אותם בקובץ zip ושלחו. אין לשלוח קבצים נוספים.

בהצלחה