

מטלת מנחה (ממ"ן) 11

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 1-2

נושאי המטלה: יסודות השפה

מספר השאלות: 1

משקל המטלה: 3 נקודות

סמסטר: 2022

מועד אחרון להגשה: 26.3.2022


(ת)

שימו לב:

- יש להקפיד על שמות המחלקות בדיוק כמו שנכתבו.
- יש לתעד את התכניות בתיעוד פנימי באנגלית בלבד (בתחילת התכנית התיעוד מסביר מה מבצעת התכנית באופן כללי ובמהלך התכניות התיעוד מסביר את הקוד).
- אין להוסיף שיטות מעבר לאלה הנדרשות במטלה במפורש.
- אין להשתמש בחומר מתקדם ובפרט לא בלולאות.
- אין להשתמש בפקודה `exit()`.
- יש להשתמש בקבועים היכן שאפשר.
- יש להקפיד על הזחה (אינדנטציה - עימוד) נכונה, ועל שמות משתנים בעלי משמעות (באנגלית) ולפי המוסכמות בקורס.
- יש להקפיד על פורמט הפלט בדיוק כפי שמצוין בשאלה: איות נכון, אותיות גדולות וקטנות, רווחים, וכו'.
- באתר הקורס תוכלו למצוא קובץ הנחיה לפתרון המטלות התכנותיות. כדאי מאד לעיין בו ולפעול לפיו. הקובץ נמצא באתר הקורס בתוך "יחידה 1" ב"מדריכי עזר וקישורים" והוא נקרא "הנחיות לכתיבת תכניות ומטלות בקורס".
- הגשת המטלה נעשית אך ורק בעזרת מערכת המטלות המקוונת שבאתר הקורס.
- אל תשכחו לשמור את מספר האסמכתא שתקבלו מהמערכת לאחר ההגשה.

שאלה 1 - להרצה (100%)

לוח השחמט מחולק ל-64 משבצות: 8 שורות ו-8 עמודות. כל שורה ממוספרת מ-1 עד 8 וכן כל עמודה ממוספרת מ-1 עד 8. מיקום של כלי על לוח השחמט מוגדר באמצעות מספר שורה ומספר עמודה. כך, למשל, המשבצת (1,1) היא המשבצת השמאלית העליונה בלוח, והמשבצת (8,8) היא המשבצת הימנית התחתונה בלוח.

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

מסע פֶּרֶשׁ (Knight) על לוח שחמט מורכב ממעבר על שתי משבצות רצופות בכוון מאוזן ואחת בכוון מאונך, או שתי משבצות רצופות בכוון מאונך ואחת בכוון מאוזן. שימו לב כי אם הפרש נמצא על אחת המשבצות שבמסגרת הלוח או קרוב לה, הוא אינו יכול לבצע את כל שמונת המסעות.

התאים אליהם יכול לנוע הפרש המסומן על לוח השחמט שלעיל מסומנים בתו X.

סעיף א (40%):

כתבו תכנית הקוראת מהקלט את מיקומו של פרש על לוח השחמט ומדפיסה את כל המקומות על הלוח שהפרש יכול לנוע אליהן.

שימו לב:

- יש לכל היותר 8 אפשרויות בפלט.
- סדר ההדפסה אינו משנה.

לדוגמה, עבור הקלט: 5 7 יודפס הפלט הבא המתאר את המיקומים האפשריים:

Moves:

3 8

3 6

4 5

6 5

7 6

7 8

שימו לב, במקרה של קלט לא חוקי (כלומר, מיקום החורג מגבולות הלוח) התוכנית תדפיס הודעת שגיאה כזו: `input is illegal` ותסיים. הניחו שהקלט הוא אכן מספרים שלמים בלבד, אין צורך לבדוק זאת.

לשם קריאה מהקלט השתמשו במחלקה `Scanner`.

כדי להשתמש בה צריך לכתוב בראשית התכנית את השורה

```
import java.util.Scanner;
```

אפשר למצוא את הממשק של המחלקה `Scanner` בתוך סעיף 2.6 Interactive Programs מהספר `Java Software Solutions` שנמצא לאחר הסרטון 2.4, שם מובאות חלק מהשיטות. הסברים על המחלקה `Scanner` והשימוש בה אפשר למצוא באתר הקורס בתוך "יחידה 2" ב"מדריכי עזר וקישורים" בקובץ "מדריך לעבודה עם המחלקה `Scanner` לקבלת קלט מהמשתמש".

התכנית שכתבתם צריכה להיות במחלקה בשם `Knight`.

כתבנו כאן חלקים מהמחלקה. עליכם להשתמש בקוד זה ולהשלים את החסר (גם את התיעוד החסר).

אל תשכחו להוסיף קבועים לפי הצורך.

שימו לב – פלט התוכנית חייב להיות זהה לפלט המופק מהתוכנית המופיעה כאן (למטה) ואז ההודעה `"input is illegal"` או המילה `"Moves:"` ואז רשימת המיקומים האפשריים לפרש. עבור רשימת המיקומים – כל מיקום אפשרי יוצג בשורה נפרדת עם רווח בין סיפורה לסיפורה.

להלן תחילת המחלקה Knight:

```
import java.util.Scanner;
public class Knight
{
    public static void main (String [] args)
    {
        // כאן צריכים להופיע קבועים מתאימים ...

        Scanner scan = new Scanner (System.in);
        System.out.println ("This program reads two integers which " +
            "represent the knight's location on the chess board: ");
        System.out.println ("Please enter the number of row");
        int row = scan.nextInt();
        System.out.println ("Please enter the number of column");
        int col = scan.nextInt();

        // כאן עליכם להמשיך ...

    } // end of method main
} //end of class Knight
```

סעיף ב (60%):

כתבו תכנית הבודקת "איומים" של כלים על לוח השחמט. אנחנו מתמקדים בכלים הבאים:

1. רץ – bishop, יכול ללכת רק באלכסון לכל ארבעת הכיוונים האפשריים. הוא יכול לנוע משבצת אחת או מספר משבצות (עד לקצה גבול הלוח). אם הוא נתקל בכלי אויב, הוא יכול להכותו - ולהתייצב במקומו.
 2. צריח – rook, יכול ללכת רק על אותה שורה או על אותה עמודה. הוא יכול לנוע משבצת אחת או מספר משבצות (עד לקצה גבול הלוח). אם הוא נתקל בכלי אויב, הוא יכול להכותו - ולהתייצב במקומו.
 3. פרש – knight, את מסעותיו תיארונו לעיל. אם הפרש מסיים את מסעו במיקום שבו נמצא כלי אויב, הוא יכול להכותו - ולהתייצב במקומו.
- כשכלי מסוים יכול להגיע לכלי אחר ולהכות אותו לפי הכללים שציינו לעיל, אנו אומרים שהוא "מאיים" עליו.

התכנית צריכה לקרוא מהקלט שמות של שני כלים לפי האות הראשונה שלהם: פרש ('k'), רץ ('b') וצריח ('r') ואת מיקומיהם. הניחו כי האותיות בקלט הן אכן 'k', 'b' או 'r' והמספרים הם אכן מספרים שלמים. אין צורך לבדוק זאת.

התכנית צריכה לבדוק האם הכלים שונים זה מזה (מבחינת סוג הכלי), אם מיקומיהם של שני הכלים חוקיים (כלומר הם בגבולות הלוח) וכן אם המיקומים שונים זה מזה. אם כן, התכנית צריכה לבדוק אם יש איום כלשהו בין שני הכלים, ולהדפיס הודעה מתאימה.

ההודעות יהיו בדיוק אלו:

- אם הכלים זהים, יודפס: Chessmen should be different from each other
- אם לפחות אחד המיקומים לא חוקי, יודפס: Position is not legal
- אם המיקומים זהים, יודפס: Chessmen positions should not be identical
- אם כלי אחד מאיים על אחר (נניח, פרש מאיים על צריח), יודפס: knight threatens rook
- אם אין איומים, יודפס: no threat

דוגמאות:

- עבור הקלט:

b 2 3

r 4 5

יודפס: bishop threatens rook

- עבור הקלט:

b 2 3

r 5 5

יודפס: no threat

- עבור הקלט:

k 10 3

r 7 5

יודפס: Position is not legal

שימו לב, הבדיקות יתבצעו לאחר קריאת כל הקלט. ברגע שהתגלתה שגיאה מסוימת, אין צורך לבדוק הלאה, וצריך לסיים את התכנית. לכן אין צורך להדפיס את כל השגיאות שיש אלא רק את הראשונה. סדר הבדיקות צריך להיות לפי הסדר המתואר בהודעות למעלה. כמו כן, שימו לב שבשל אופן תנועת הכלים לא יתכן מצב בו שני כלים שונים יאיימו זה על זה בו זמנית.

גם כאן, כמו בסעיף א, לשם קריאה מהקלט השתמשו במחלקה Scanner.

התכנית שכתבתם צריכה להיות במחלקה בשם Chess.

כתבנו כאן חלקים מהמחלקה. עליכם להשתמש בקוד זה ולהשלים את החסר (גם את התיעוד החסר).

אל תשכחו להוסיף קבועים לפי הצורך.

שימו לב – פלט התוכנית חייב להיות זהה לפלט המופק מהתוכנית המופיעה כאן (למטה) ואז ההודעה המתאימה.

להלן תחילת המחלקה Chess:

```
import java.util.Scanner;
public class Chess
{
    public static void main (String [] args)
    {
        // כאן צריכים להופיע קבועים מתאימים...

        Scanner scan = new Scanner (System.in);
        System.out.println("Please enter the type"+
                           " of the first chessman");
        char first = scan.next().charAt(0);
        System.out.println ("Please enter the number of row");
        int row1 = scan.nextInt();
        System.out.println ("Please enter the number of column");
        int col1 = scan.nextInt();
        System.out.println("Please enter the type"+
                           " of the second chessman");
        char second = scan.next().charAt(0);
        System.out.println ("Please enter the number of row");
        int row2 = scan.nextInt();
        System.out.println ("Please enter the number of column");
        int col2 = scan.nextInt();

        // כאן עליכם להמשיך...

    } // end of method main
} //end of class Chess
```

בממ"ן זה אתם צריכים לכתוב שתי מחלקות. מחלקה Knight עבור השאלה הראשונה ומחלקה Chess עבור השאלה השניה. בכל מחלקה תהיה שיטה אחת – main. אין להוסיף שיטות נוספות.

במטלה זו אסור להשתמש בלולאות ו/או במערכים!

הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הפתרון לסעיף א כולל את הקובץ Knight.java.
3. הפתרון לסעיף ב כולל את הקובץ Chess.java.
4. פתרון שיכלול שגיאה בשמות אחת המחלקות או השיטה main לא יבדק וציונו יהיה 0.
5. ארזו את שני הקבצים בקובץ zip (ולא rar) יחיד ושלחו אותו בלבד.
6. **אל תשכחו לשמור את מספר האסמכתא שקיבלתם מהמערכת לאחר ההגשה. אם לא קיבלתם מספר אסמכתא, סימן שההגשה לא התקבלה.**
7. שימו לב, אתם יכולים לשלוח שוב ושוב את המטלה במערכת, אם אתם רוצים לתקן משהו בה. כל הגשה דורסת את ההגשה הקודמת. **אבל עשו זאת אך ורק עד לתאריך ההגשה.** אחרי התאריך, ייחשב לכם כאילו הגשתם באיחור, גם אם ההגשה הראשונה היתה בזמן!
8. כמו כן, אם המנחה הוריד כבר את המטלה שלכם מהמערכת, לא תוכלו לשלוח עותק מעודכן יותר.

ב ה צ ל ח ה

