



מס' שאלון - 473
14
בפברואר 2022

מס' מועד 67

סמסטר 2022א

20441 / 4

שאלון בחינת גמר

20441 - מבוא למדעי המחשב ושפת Java

משך בחינה: 3 שעות

בשאלון זה 14 עמודים

מבנה הבחינה:

- * קראו בעיון את ההנחיות שלהלן: בבחינה יש חמש שאלות.
- * כל התכניות צריכות להיות מתועדות היטב.
- יש לכתוב תחילה בקצרה את האלגוריתם וכל הסבר נוסף הדרוש להבנת התכנית.
- יש לבחור בשמות משמעותיים למשתנים, לפונקציות ולקבועים שבתכנית.
- תכנית שלא תתועד כנדרש לעיל תקבל לכל היותר 85 % מהניקוד.
- * יש להקפיד לכתוב את התכניות בצורה מבנית ויעילה.
- תכנית לא יעילה לא תקבל את מלוא הנקודות.
- * אם ברצונכם להשתמש בתשובתכם בשיטה או במחלקה הכתובה בחוברת השקפים, אין צורך שתעתיקו את השיטה או את המחלקה למחברת הבחינה. מספיק להפנות למקום הנכון, ובלבד שההפניה תהיה מדויקת (פרמטרים, מיקום וכו').
- * אין להשתמש במחלקות קיימות ב-Java, חוץ מאלו המפורטות בשאלות הבחינה.
- * יש לשמור על סדר; תכנית הכתובה בצורה בלתי מסודרת עלולה לגרוע מהציון.
- * בכתיבת התכניות יש להשתמש אך ורק במרכיבי השפה שנלמדו בקורס זה
- אין להשתמש במשתנים גלובליים!
- * אפשר לתעד בעברית. אין צורך בתיעוד API.
- כל התשובות צריכות להיכתב בתוך קובץ המבחן במקומות המתאימים בלבד.
- תשובה שתיכתב שלא במקומה לא תיבדק.
- חומר עזר: חוברות השקפים 1-6, 7-12. אין להכניס חומר מודפס נוסף או כל חומר אחר. אין להכניס מחשב/מחשבון/מכשיר אלקטרוני מכל סוג שהוא.

בהצלחה !!!

חלק א – עליכם לענות על כל השאלות בחלק זה בקובץ הפתרון

שאלה 1 (25 נקודות)

נגדיר:

1. **מטריצת היחידה** (Identity matrix) (מסדר n) היא מטריצה (מערך דו-ממדי) ריבועית בגודל $n \times n$ (n שורות ו- n עמודות) שהאלכסון הראשי שלה מורכב מאחדות וכל שאר המטריצה מאפסים (ויקיפדיה).
לדוגמא, המטריצה A להלן היא מטריצת יחידה מסדר 3 (האלכסון הראשי מודגש), והמטריצה B להלן אינה מטריצת יחידה, אולם אם האיבר $B[1][1]$ היה 1 ולא 2, והאיבר $B[1][3]$ היה 0 ולא 3, אז היא היתה מטריצת יחידה מסדר 4.

	0	1	2	3
0	1	0	0	0
1	0	2	0	3
2	0	0	1	0
3	0	0	0	1

B=

	0	1	2
0	1	0	0
1	0	1	0
2	0	0	1

A=

2. **תת-מטריצה מרכזית** של מטריצה ריבועית A היא תת מטריצה ריבועית שמרכזה מתלכד עם המרכז של A .

לדוגמא:

	0	1	2	3	4	5	6
0							
1							
2							
3				מרכז			
4							
5							
6							

המטריצה המסומנת במרכז, בגודל 3×3 היא תת-מטריצה מרכזית שמתלכדת עם המטריצה הגדולה בגודל 7×7 .

בשאלה זו נחשב את הגודל המקסימלי של התת-מטריצה המרכזית של A שהיא מטריצת יחידה.

לדוגמא עבור המטריצה A שלהלן :

	0	1	2	3	4	5	6
0	1	2	3	2	0	1	2
1	0	1	0	0	0	3	0
2	0	0	1	0	0	0	0
3	5	0	0	1	0	0	0
4	7	0	0	0	1	0	0
5	8	0	0	0	0	1	0
6	1	0	0	0	0	0	0

הגודל המקסימלי של התת-מטריצה המרכזית שהיא מטריצת יחידה הוא 3. (המטריצה הזו מובלטת).

אם למשל $A[1][5]$ היה שווה 0 (במקום 3) אזי הגודל המקסימלי של התת-מטריצה המרכזית שהיא מטריצת יחידה היה 5.

סעיף א: (17 נקודות)

כתבו שיטה סטטית **רקורסיבית** בוליאנית `isIdentity` המקבלת כפרמטרים: מטריצה **ריבועית** בשם `mat`, משתנה שלם בשם `x` ומשתנה שלם בשם `size`. השיטה צריכה להחזיר את הערך `true` אם התת-מטריצה בגודל `size` שהפינה השמאלית העליונה שלה היא `mat[x][x]` היא מטריצת יחידה, ו-`false` אחרת.

לדוגמא, על המטריצה לעיל:

כאשר $x=0$, אם `size=1`, השיטה תחזיר `true`.

כאשר $x=1$, אם `size` שווה ל-1 או 2 או 3 או 4, השיטה תחזיר `true`.

כאשר $x=4$, אם `size` שווה ל-1 או 2, השיטה תחזיר `true`, אבל אם `size` שווה ל-3, השיטה תחזיר `false` (שכן המטריצה שהפינה השמאלית העליונה שלה היא באיבר $A[4][4]$ והיא בגודל 3 אינה מטריצת היחידה, כי האיבר הימני התחתון שלה הוא $A[6][6]$ והוא שונה מ-1).

חתימת השיטה היא:

```
public static boolean isIdentity (int[][] mat, int x, int size)
```

אתם יכולים להניח ש-`x` ו-`size` לא מובילים לחריגה מגבולות המטריצה. אין צורך לבדוק זאת.

סעיף ב: (8 נקודות)

כתבו שיטה סטטית רקורסיבית `maxMatrix` המקבלת כפרמטר מטריצה ריבועית בשם `mat`. המטריצה צריכה להחזיר את הגודל המקסימלי של התת-מטריצה המרכזית של `mat` שהיא מטריצת יחידה.

בסעיף זה אפשר להניח שהמטריצה `mat` היא ריבועית ושמספר השורות והעמודות הוא אי-זוגי.

לדוגמא, אם המטריצה לעיל היתה מועברת כפרמטר לשיטה, היה מוחזר הערך 3. שימו לב שאמנם יש במטריצה הזו מטריצת יחידה גדולה יותר, שמתחילה אף היא באיבר `A[2][2]` והיא בגודל 4 (הפינה הימנית התחתונה שלה היא `A[5][5]`, אבל המרכז של המטריצה הזו אינו מתלכד עם המטריצה `A`, ולכן היא אינה תת-מטריצה מרכזית של `A`).

חתימת השיטה היא:

```
public static int maxMatrix(int[][] mat)
```

שימו לב, מותר לכם להשתמש בשיטה `isIdentity` מסעיף א, גם אם לא פתרתם אותה.

כל השיטות שתכתבו צריכות להיות רקורסיביות ללא שימוש בלולאות כלל. כך גם כל שיטות העזר שתכתבו (אם תכתבו) לא יכולות להכיל לולאות.

בשני הסעיפים:

- אפשר להשתמש בהעמסת-יתר (overloading).
- אפשר להניח שהמטריצה אינה `null` ואינה ריקה. כמו כן אפשר להניח שהמטריצה היא ריבועית. אין צורך לבדוק זאת!
- אין לשנות את תוכן המטריצה (אפילו לא זמנית), ולא להשתמש במערכי עזר.
- אין צורך לדאוג ליעילות השיטה! אבל כמובן שצריך לשים לב לא לעשות קריאות רקורסיביות מיותרות!
- אל תשכחו לתעד את מה שכתבתם!

שאלה 2 (25 נקודות)

נתון מערך חד-ממדי a המלא במספרים שלמים (אין צורך לבדוק זאת) באורך גדול או שווה ל-3, המכיל רצף של איברים עוקבים מסדרה חשבונית. מהסדרה החשבונית נמחקה תת-סדרה, כלומר מספר איברים רצופים. לפחות איבר אחד ואולי יותר.

תזכורת: סדרה חשבונית היא סדרה של מספרים, שבה ההפרש בין כל שני איברים עוקבים הוא קבוע: $a_{n+1} - a_n = d$, כאשר a_n הוא האיבר ה- n בסדרה.

כתבו שיטה סטטית `findMissingIndex` המקבלת את המערך כפרמטר ומוצאת את האינדקס במערך בו היה אמור להופיע האיבר הראשון בתת-סדרה החסרה. אם לא חסר אף איבר בסדרה, יוחזר הערך של המקום שלאחר סוף המערך (כלומר אורך המערך `a.length`).

לדוגמא:

- עבור המערך $\{22, 24, 26, 32, 34, 36, 38, 40\}$ השיטה תחזיר 3, כי זהו האינדקס בו היה אמור להופיע האיבר הבא אחרי המספר 26 בסדרה החשבונית.
- עבור המערך $\{22, 26, 28\}$ השיטה תחזיר 1, כי זהו האינדקס בו היה אמור להופיע האיבר הבא אחרי המספר 22 בסדרה החשבונית.
- עבור המערך $\{2, 4, 6, 8, 10\}$ השיטה תחזיר 5, כי לא חסרה כאן תת-סדרה, ולכן זהו האינדקס בו היה אמור להופיע האיבר הבא אחרי המספר 10 בסדרה החשבונית.

חתימת השיטה היא:

```
public static int findMissingIndex(int []a)
```

מה סיבוכיות זמן הריצה והמקום של השיטה שכתבתם? הסבירו תשובתכם.

אפשר להניח שהמערך אינו `null` ואינו ריק. הוא מכיל לפחות 3 איברים, והוא מכיל סדרה חשבונית. אפשר גם להניח שאין שתי תת-סדרות שחסרות אלא מקסימום אחת. אין צורך לבדוק זאת.

שימו לב:

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.

אל תשכחו לתעד את מה שכתבתם!

**חלק ב - את התשובות לשאלות 3-5 יש לכתוב על גבי השאלון.
לא נבדוק תשובות שייכתבו במקום אחר!**

שאלה 3 (20 נקודות)

נניח שהמחלקה Node שלהלן מממשת עץ בינרי.

```
public class Node {
    private int _number;
    private Node _leftSon, _rightSon;

    public Node (int number) {
        _number = number;
        _leftSon = null;
        _rightSon = null;
    }

    public int getNumber() {return _number; }
    public Node getLeftSon() {return _leftSon; }
    public Node getRightSon() {return _rightSon; }
}
```

המחלקה BinaryTree מאגדת בתוכה שיטות סטטיות לטיפול בעץ בינרי.

בין השיטות נתונות השיטות f ו-something הבאות:

- השיטה f מקבלת שורש של עץ בינרי.

```
public static int f(Node root)
{
    if (root == null)
        return Integer.MIN_VALUE;

    if (root.getLeftSon() == null &&
        root.getRightSon() == null)
        return root.getNumber();

    int left = f(root.getLeftSon());
    int right = f(root.getRightSon());

    return (left > right? left : right) + root.getNumber();
}
```

תזכורת: המספר Integer.MIN_VALUE הוא המספר השלם הקטן ביותר שאפשר לייצג במחשב.

- השיטה something מקבלת שורש של עץ בינרי ומספר שלם.

```
public static boolean something(Node root, int num)
{
    if (num == 0 && root == null)
        return true;

    if (root == null)
        return false;

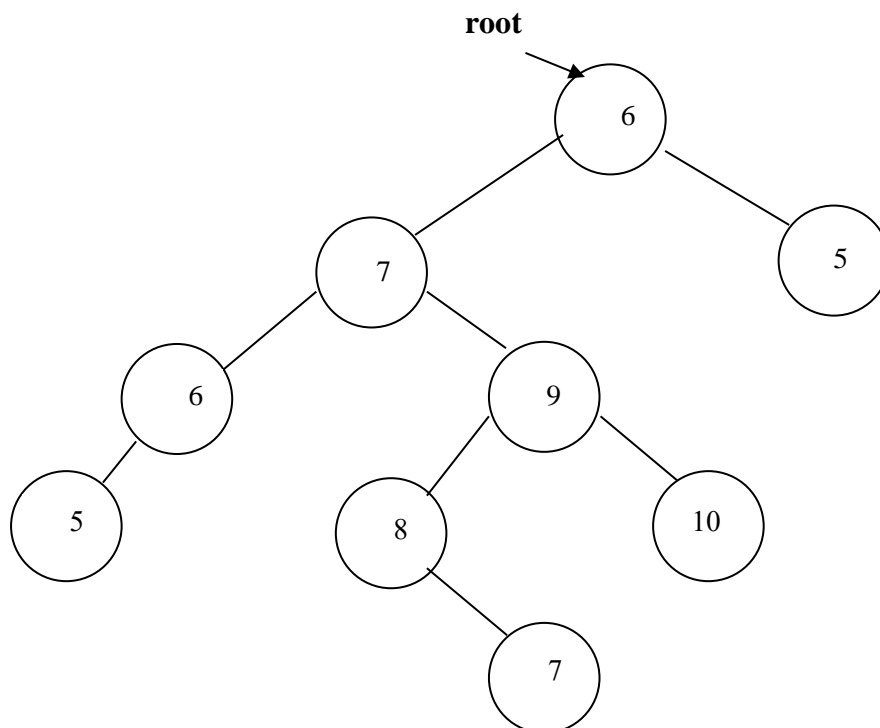
    boolean left = something(root.getLeftSon(),
                             num - root.getNumber());
    boolean right = false;

    if (!left)
        right = something(root.getRightSon(),
                          num - root.getNumber());

    if (left || right)
        System.out.print(root.getNumber() + " ");

    return left || right;
}
```

נתון העץ הבינרי הבא, ששורשו הוא root



ענו על הסעיפים הבאים:

4 נק') (א) איזה ערך תחזיר השיטה `f` בעקבות הקריאה `BinaryTree.f(root)`?

התשובה היא:

6 נק') (ב) מה מבצעת השיטה `f` באופן כללי כשהיא מקבלת כפרמטר שורש של עץ בינרי `root`? שימו לב, עליכם לתת תיאור ממצה של מה עושה השיטה באופן כללי, ולא תיאור של מה עושה כל שורה בשיטה, או איך היא מבצעת זאת. כלומר, מה המשמעות של הערך שהשיטה מחזירה? התייחסו למקרי קצה!

התשובה היא:

4 נק') (ג) מה יודפס בעקבות הקריאה `BinaryTree.something(root, 32)`?

התשובה היא:

6 נק') (ד) מה מבצעת השיטה `something` באופן כללי כשהיא מקבלת כפרמטר שורש של עץ בינרי `root`, ומספר שלם `num`? שימו לב, עליכם לתת תיאור ממצה של מה עושה השיטה באופן כללי, ולא תיאור של מה עושה כל שורה בשיטה, או איך היא מבצעת זאת. התייחסו למקרי קצה!

התשובה היא:

שאלה 4 (14 נקודות)

נתון פרויקט שהוגדרו בו המחלקות שלהלן. כל אחת בקובץ נפרד, כמובן.

```
public abstract class A
{
    protected int _aVal;
    public A(){
        _aVal = 1;
    }
    public A(int aVal){
        _aVal = aVal;
    }
    public abstract boolean f(int n);
    public void g(A x){
        System.out.println("result: " + (x._aVal + _aVal));
    }
}

-----

public class B extends A
{
    private int _bVal;
    public B(int aVal, int bVal){
        super(aVal);
        _bVal = bVal;
    }
    public boolean f(int n){
        return n == _bVal;
    }
    public void g(B x){
        System.out.println("result: " + (_bVal + ((B)x)._bVal));
    }
}

-----

public class C extends A
{
    private int _cVal;
    public C(int aVal, int cVal){
        _cVal = cVal;
    }
    public boolean f(int n){
        return n == _aVal;
    }
    public void g(A x){
        System.out.println("result: " + (_cVal + x._aVal));
    }
    public void g(C x){
        System.out.println("result: " + (_cVal + x._cVal));
    }
}
```

כמו כן נתונה המחלקה Driver הבאה באותו פרויקט:

```
public class Driver
{
    public static void main (String [] args)
    {
        A b1 = new B(3, 5);
        B b2 = new B(5, 5);
        A c1 = new C(4, 4);
        C c2 = new C(5, 6);

        //      כאן יופיעו הפקודות שבסעיפים להלן
    }
}
```

לפניכם שבעה סעיפים. בכל סעיף בחרו בתשובה המתאימה ביותר וסמנו אותה. הניחו כי הפקודות המתוארות בשאלות מופיעות במחלקה Driver בשיטה main במקום המסומן כהערה. אין תלות בין הסעיפים. כל סעיף 2 נקודות.

1. איזו מהקריאות הבאות תחזיר את הערך true?

- א. b1.f(3);
- ב. b2.f(1);
- ג. c1.f(4);
- ד. c2.f(1);

2. איזו מהקריאות הבאות תדפיס את השורה: result: 10

- א. b2.g(b1);
- ב. b2.g(b2);
- ג. c1.g(c2);
- ד. c1.g(c1);

3. איזו מהקריאות הבאות תדפיס את השורה: result: 9

- א. c2.g(b1);
- ב. b2.g(c1);
- ג. c1.g(b1);
- ד. b1.g(c2);

4. בעקבות הוספת הפקודה: ((A)c2).g(c2);

- א. תודפס השורה: result: 7
- ב. תודפס השורה: result: 10
- ג. תודפס השורה: result: 11
- ד. תודפס השורה: result: 12
- ה. תופיע שגיאת קומפילציה
- ו. תופיע שגיאת ריצה

5. בעקבות הוספת הפקודה: ((B)b1).g(b2);

א. תודפס השורה: result: 4

ב. תודפס השורה: result: 6

ג. תודפס השורה: result: 8

ד. תודפס השורה: result: 10

ה. תופיע שגיאת קומפילציה

ו. תופיע שגיאת ריצה

6. בעקבות הוספת הפקודה: ((C)b1).g(c2);

א. תודפס השורה: result: 4

ב. תודפס השורה: result: 6

ג. תודפס השורה: result: 8

ד. תודפס השורה: result: 10

ה. תופיע שגיאת קומפילציה

ו. תופיע שגיאת ריצה

7. בעקבות הוספת הפקודה: ((C)b2).g(c1);

א. תודפס השורה: result: 4

ב. תודפס השורה: result: 6

ג. תודפס השורה: result: 8

ד. תודפס השורה: result: 10

ה. תופיע שגיאת קומפילציה

ו. תופיע שגיאת ריצה

שאלה 5 (16 נקודות)

נתונה המחלקה StrStack המממשת מחסנית מחרוזות של תווים.

להלן ה-API של המחלקה StrStack :

Constructor Summary

Constructors
Constructor and Description
<div>StrStack()</div> <div>Constructs an empty StrStack object.</div>

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method and Description	
boolean	<div>isEmpty()</div> <div>Checks whether the stack is empty.</div>	
java.lang.String	<div>pop()</div> <div>Removes an item from the top of the stack and returns it.</div>	
void	<div>push(java.lang.String s)</div> <div>Pushes an item onto the top of the stack.</div>	

הניחו שהשיטות לעיל ממומשות במחלקה StrStack.

במחלקה StrStack הוגדרה גם השיטה what שתובא להלן (בעמוד הבא).

כיוון שאנחנו עוסקים במחסנית של מחרוזות, להלן תזכורת לשיטות מהמחלקה String שנשתמש בהן בשיטה what.

- `s.length()` – המחזירה את אורך המחרוזת `s`
- `s.charAt(int index)` – המחזירה את התו הנמצא במיקום `index` במחרוזת `s`
- `s.indexOf(char c)` – המחזירה את המיקום הראשון בו נמצא התו `c` במחרוזת `s`
- `s.substring(int i)` – המחזירה את התת-מחרוזת של `s` המתחילה במיקום `i` ועד לסוף המחרוזת.
- `s.substring(int i, int j)` – המחזירה את התת-מחרוזת של `s` המתחילה במיקום `i` ועד למיקום `j-1` (כולל).

השיטה what המופיעה להלן מקבלת מחרוזת תווים המכילה את התווים '0', '1', '?' בלבד.

```
public static void what (String pattern)
{
    if (pattern == null || pattern.length() == 0) {
        return;
    }

    StrStack stack = new StrStack();
    stack.push(pattern);

    int index;

    while (!stack.isEmpty())
    {
        String curr = stack.pop();
        if ((index = curr.indexOf('?')) != -1)
        {
            for (char ch = '0'; ch <= '1'; ch++)
            {
                curr = curr.substring(0, index) + ch +
                    curr.substring(index + 1);
                stack.push(curr);
            }
        }
        else {
            System.out.println(curr);
        }
    }
}
```

סעיף א (4 נקודות)

אם נפעיל את השיטה **what** כשהפרמטר `pattern = "1?1?1"`, מה יודפס? (לא בהכרח כל השורות יתמלאו).

התשובה היא

סעיף ב (8 נקודות)

אם נפעיל את השיטה **what** כשהפרמטר `pattern = "0???0"`, מה יודפס? לא בהכרח כל השורות יתמלאו).

התשובה היא

סעיף ג (4 נקודות)

לאחר הפעלת השיטה **what** עם הפרמטר מחרוזת תווים כלשהי `x`, התקבלה ההדפסה הבאה:

11111100
11111000
11110100
11110000
10111100
10111000
10110100
10110000

מה היתה המחרוזת `x` שהתקבלה כפרמטר לשיטה?

התשובה היא

ב ה צ ל ח ה !