

ש N101137561  
  
 מספר 15  
 סידורי ת.ז.

מספר התלמיד הנבחן  
 רשום את כל תשע הספרות

האוניברסיטה  
 הפתוחה 

ל' בשבט תשע"ח

מס' שאלון - 483

15

בפברואר 2018

סמסטר 2018 א

מס' מועד 85

20441 / 4

שאלון בחינת גמר

20441 - מבוא למדעי המחשב ושפת Java

משך בחינה: 3 שעות

בשאלון זה 12 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!

\* את התשובות לשאלות 3 - 5 יש לכתוב על גבי השאלון. לא נבדוק תשובות

שייכתבו במקום אחר!

\* אפשר לתעד בעברית. אין צורך בתיעוד API.

חומר עזר:

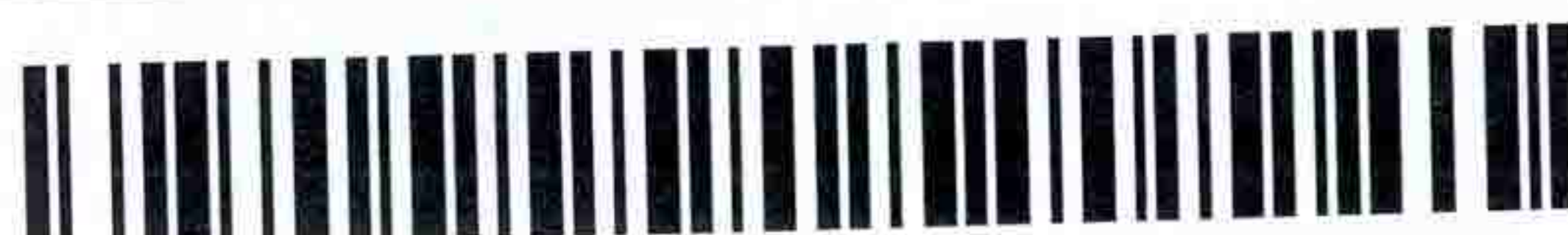
ספר הלימוד : java software solutions מאת : lewis/loftus  
 חוברת השקפים של הקורס של ד"ר אמיר גורן ותמר וילנר.  
 יחידות 1-6, 7-12. מותרות הערות בכתב יד, ע"ג הספרים.  
 אין להכניס חומר מודפס או כל חומר אחר מכל סוג שהוא.

החזירו

למשגיח את השאלון

וכל עזר אחר שקיבלתם בתור מחברת התשובות

בהצלחה !!!





## חלק א – עליכם לענות על כל השאלות בחלק זה במחברת הבחינה

### שאלה 1 - 25 נקודות

עבור מערך דו-ממדי של מספרים שלמים, נגדיר **מדרון** (slope) **מסדר** num באורך k במערך להיות סדרה של k תאים, שכל אחד מהם סמוך לקודמו, כלומר, הוא אחד מארבעת שכניו במערך (שכן מעליו, מתחתיו, מימינו ומשמאלו, אך לא באלכסון לו), כך שהערכים בתאים מהווים סדרה יורדת של מספרים שלמים **שההפרש ביניהם הוא num**:

$n, n - \text{num}, n - (2 * \text{num}) \dots, n - ((k - 1) * \text{num})$ .

n, האיבר הראשון בסדרה, עשוי להיות כלשהו. למשל, במערך:

	0	1	2	3	4
0	3	13	15	28	30
1	55	54	53	27	26
2	54	12	52	51	50
3	50	10	8	53	11

אם אנחנו מחפשים מדרון מסדר 2, אז

- בתא [1][2] מתחיל המדרון מסדר 2 באורך שלוש הבא: 12, 10, 8 (הנגמר בתא [3][2])

אם אנחנו מחפשים מדרון מסדר 1, אז

- בתא [0][3] מתחיל המדרון מסדר 1 באורך שלוש הבא: 28, 27, 26 (הנגמר בתא [1][4])
- בתא [1][0] מתחיל המדרון מסדר 1 באורך שש הבא: 55, 54, 53, 52, 51, 50 (הנגמר בתא [2][4])

- בתא [1][0] מתחיל המדרון מסדר 1 באורך שתיים הבא: 55, 54 (הנגמר בתא [2][0])

כתבו שיטה סטטית **רקורסיבית** המקבלת מערך דו-ממדי מלא במספרים שלמים, ומספר שלם num, ומחזירה את אורכו של המדרון מסדר num הארוך ביותר במערך.

**לדוגמא,**

- אם השיטה תקבל את המערך שלעיל, וערך  $\text{num} = 1$  היא תחזיר את הערך 6.
- אם השיטה תקבל את המערך שלעיל, וערך  $\text{num} = 2$  היא תחזיר את הערך 3.

**חתימת השיטה:**

```
public static int longestSlope (int [][] mat, int num)
```



```

public static int longestSlope (int [][] mat, int num)
{
    return longestSlope (mat,num,0,0,1);
}

private static int longestSlope (int [][] mat,int num, int i, int j, int max)
{
    if (j == mat[0].length) {
        return longestSlope(mat,num,i+1,0,max);
    }else if(i == mat.length) {
        return max;
    }else {
        int newPath = findPath(mat,num,i,j,mat[i][j]+num);
        return longestSlope(mat,num,i,j+1,Math.max(max, newPath));
    }
}

private static int findPath (int [][] mat,int num, int i, int j, int prev)
{
    if (i < 0 || j < 0 || i >= mat.length || j >= mat[0].length || mat[i][j] == -1)
        return 0;

    if(prev - mat[i][j] != num)
        return 0;

    int up = findPath(mat,num,i-1,j,mat[i][j])+1;
    int down = findPath(mat,num,i+1,j,mat[i][j])+1;
    int right = findPath(mat,num,i,j+1,mat[i][j])+1;
    int left = findPath(mat,num,i,j-1,mat[i][j])+1;
    int max1 = Math.max(right, left);
    int max2 = Math.max(up,down);
    return Math.max(max1, max2);
}

```



השיטה צריכה להיות רקורסיבית ללא שימוש בלולאות כלל. כך גם כל שיטות העזר שתכתבו (אם תכתבו) לא יכולות להכיל לולאות.

אפשר להשתמש בהעמסת-יתר (overloading).

מותר לשנות את המערך במהלך השיטה, אבל חובה להחזיר אותו למצבו ההתחלתי בסופה.

אין צורך לדאוג ליעילות השיטה!

## שאלה 2 - 25 נקודות

נתונות השיטות הסטטיות הבאות:

```
private static int f (int[]a, int low, int high)
{
    int res = 0;
    for (int i=low; i<=high; i++)
        res += a[i];
    return res;
}

public static int what (int []a)
{
    int temp = 0;
    for (int i=0; i<a.length; i++)
    {
        for (int j=i; j<a.length; j++)
        {
            int c = f(a, i, j);
            if (c%2 == 1)
            {
                if (j-i+1 > temp)
                    temp = j-i+1;
            }
        }
    }
    return temp;
}
```

- המשך השאלה בעמוד הבא -



**ענו על ארבעת הסעיפים הבאים:**

א. מה מבצעת השיטה what בהינתן לה מערך a מלא במספרים שלמים (חיוביים ושליילים)? הסבירו בקצרה מה מבצעת השיטה ולא כיצד היא מבצעת זאת. כלומר, כתבו מה המשמעות של המספר המוחזר מהשיטה what. (6 נק')

ב. מהי סיבוכיות הזמן של השיטה what ? (2 נק')

ג. כתבו את השיטה what כך שתבצע את מה שביצעה בסעיף א בסיבוכיות זמן ריצה קטנה יותר. (15 נק')

**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.**

ד. מה סיבוכיות זמן הריצה של השיטה שכתבתם בסעיף ג? הסבירו תשובתכם. (2 נק')

**אל תשכחו לתעד את מה שכתבתם!**

א. השיטה בודקת את התת מערך שסכומו יהיה הקטן ביותר כאי זוגי

ב.  $O(n)^3$

ד.  $O(n)$



```
public static int what(int [] a) {  
    int low = 0;  
    int high = a.length-1;  
    int len = a.length;  
    int sum = 0;  
  
    for(int i=0; i< high; i++) {  
        sum+=a[i];  
    }  
    if(sum%2 == 1) {  
        return len;  
    }  
    len--;  
    while(low <= high) {  
        if(sum%2 == 1) {  
            return len;  
        }  
        if(a[low]%2 == 1 || a[high]%2 == 1) {  
            return len;  
        }  
        low++;  
        high--;  
        len--;  
    }  
    return 0;  
}
```



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

### שאלה 3 - 15 נקודות

נניח שהמחלקה Node שלהלן מממשת עץ בינרי.

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

    public Node (int data)
    {
        _data = data;
        _leftSon = null;
        _rightSon = null;
    }

    public int  getData()           {return _data; }
    public Node getLeftSon()        {return _leftSon; }
    public Node getRightSon()       {return _rightSon; }
}
```

המחלקה BinaryTree מאגדת בתוכה שיטות סטטיות לטיפול בעץ בינרי.

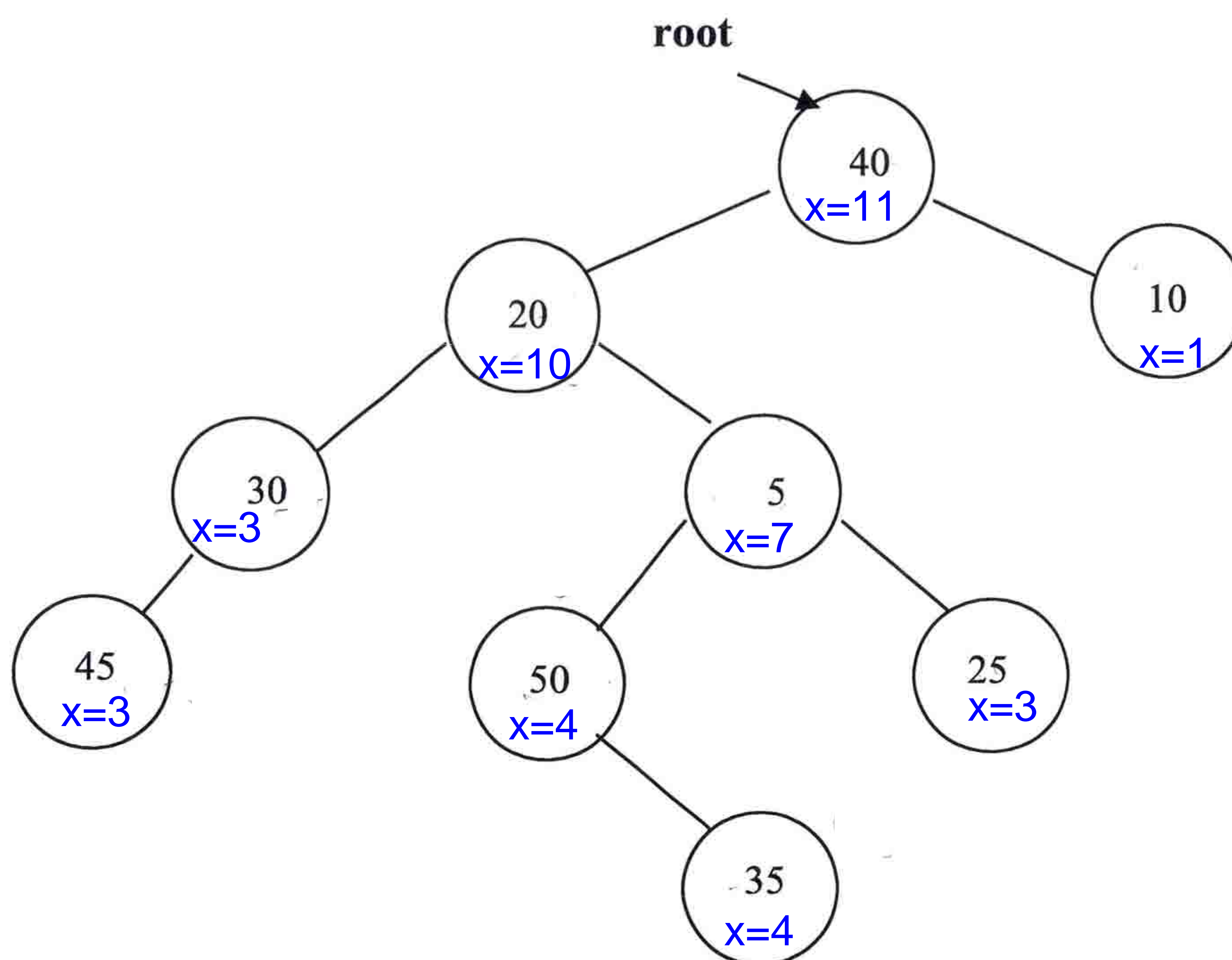
בין השיטות נתונות השיטות what הבאות:

```
public static int what (Node node)
{
    return what (node, 0);
}

private static int what( Node node, int x )
{
    if ( node == null )
        return 0;
    if ( node.getLeftSon() == null &&
        node.getRightSon() == null)
        return x;
    return what(node.getLeftSon(), x + 1) +
           what(node.getRightSon(), x + 1);
}
```



נתון העץ הבינרי הבא, ששורשו הוא root



ענו על שני הסעיפים הבאים:

(5 נק') (i) איזה ערך תחזיר השיטה what בעקבות הקריאה BinaryTree.what(root)?

התשובה היא:

11

(10 נק') (ii) מה מבצעת השיטה what באופן כללי? 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.

התשובה היא:

השיטה מחזירה את סכום הגבהים בעץ, כלומר את סכום הצעדים שיש לבצע מהשורש עד לכל עלה בעץ.  
אם העץ ריק השיטה תחזיר 0.



## שאלה 4 - 18 נקודות

בהינתן המחלקות Text, Sentence ו- Statement הבאות (כל מחלקה נמצאת בקובץ נפרד):

```
public class Text
{
    private String _word;

    public Text(String w)
    {
        _word = w;
    }

    public String getWord()
    {
        return _word;
    }

    public boolean equals (Text other) // שיטה 1
    {
        return ((other!=null)&&(_word.equals(other._word)));
    }
}

public class Sentence
{
    private String _word;

    public Sentence(String w)
    {
        _word = w;
    }

    public String getWord()
    {
        return _word;
    }

    public boolean equals (Object other) // שיטה 2
    {
        return ((other!=null) && (other instanceof Sentence)
                && (_word.equals(((Sentence) other)._word)));
    }
}
```



```

public class Statement extends Text
{
    public Statement(String w)
    {
        super(w);
    }

    public boolean equals (Object other) // שיטה 3
    {
        return ((other!=null) && (other instanceof Statement)
            &&(getWord().equals(((Statement)other).getWord())));
    }

    public boolean equals (Statement other) // שיטה 4
    {
        return ((other!=null)
            && (getWord().equals((other).getWord())));
    }
}

```

בהינתן המחלקה Question הבאה:

```

public class Question
{
    public static void main (String []args)
    {
        Text y1 = new Text("abc");
        Object y2 = new Text ("abc");

        Sentence z1 = new Sentence ("abc");
        Object z2 = new Sentence ("abc");

        Statement x1 = new Statement ("abc");
        Object x2 = new Statement ("abc");
        Text x3 = new Statement ("abc");
        Statement x4 = new Statement ("abd");

        // ***
    }
}

```

לכל אחת מהשורות שבעמוד הבא, כתבו מה יקרה בעקבות הוספתה לשיטה main שלעיל, לאחר ההצהרות על האובייקטים (במקום שמסומן בכוכביות \*\*\*).

השלימו את התשובה וסמנו מה מספר השיטה שנקראה, בדרך שאינה משתמעת לשני פנים. כל סעיף 2 נקודות.

מאגזין תוכנות



1. `System.out.println (x1.equals(y1)) ;`

יודפס True בעקבות קריאה לשיטה 1 אחרת

2. `System.out.println (x2.equals(z2)) ;`

יודפס False בעקבות קריאה לשיטה 3 אחרת

3. `System.out.println (x3.equals(z2)) ;`

יודפס False בעקבות קריאה לשיטה 3 אחרת

4. `System.out.println (y2.equals(x1)) ;`

יודפס False בעקבות קריאה לשיטה Else אחרת

5. `System.out.println (z2.equals(x2)) ;`

יודפס False בעקבות קריאה לשיטה 2 אחרת

6. `System.out.println (y2.equals(x3)) ;`

יודפס False בעקבות קריאה לשיטה Else אחרת

7. `System.out.println (x1.equals(x2)) ;`

יודפס true בעקבות קריאה לשיטה 3 אחרת

8. `System.out.println (x3.equals(x1)) ;`

יודפס True בעקבות קריאה לשיטה 1 אחרת

9. `System.out.println (x4.equals(x1)) ;`

יודפס False בעקבות קריאה לשיטה 4 אחרת



## שאלה 5 - 17 נקודות

נתונה המחלקה IntNodeTwo הבאה, המייצגת איבר ברשימה מקושרת דו-סטריק בה יש מצביעים גם לאיבר הבא וגם לקודם, המכילה מספרים שלמים:

```
public class IntNodeTwo
{
    private int _num;
    private IntNodeTwo _next, _prev;

    public IntNodeTwo(int n) {
        _num = n;
        _next = null;
        _prev = null;
    }

    public IntNodeTwo(int num, IntNodeTwo n, IntNodeTwo p) {
        _num = num;
        _next = n;
        _prev = p;
    }

    public int getNum() { return _num; }
    public IntNodeTwo getNext() { return _next; }
    public IntNodeTwo getPrev() { return _prev; }
    public void setNum(int n) { _num = n; }
    public void setNext(IntNodeTwo node) { _next = node; }
    public void setPrev(IntNodeTwo node) { _prev = node; }
}
```

נתונה רשימה מקושרת דו-סטריק, המכילה מספרים שלמים הממומשת בעזרת המחלקה IntListTwo שלהלן: השיטה what פועלת על רשימה שאיבריה מסודרים בסדר עולה.

```
public class IntListTwo
{
    private IntNodeTwo _head, _tail;
    public IntListTwo()
    {
        _head = null;
        _tail = null;
    }

    // כאן יש עוד בנאים ושיטות...
```



```

public int f() {
    int res = 0;
    IntNodeTwo temp = _head;
    while (temp != null)
    {
        res = res + temp.getNum();
        temp = temp.getNext();
    }
    return res;
}

public int g() {
    int res = 0;
    IntNodeTwo temp = _head;
    while (temp != null)
    {
        res++;
        temp = temp.getNext();
    }
    return res;
}

public boolean what (double target) {
    int temp = g();
    int res = f();
    double c = (double) temp;
    double num=res/c;
    IntNodeTwo p1 = _head;
    IntNodeTwo p2 = _tail;
    while (p1 != p2)
    {
        if (num == target)
            return true;
        if (num < target)
        {
            res = res - p1.getNum();
            p1 = p1.getNext();
            c--;
            num = res/c;
        }
        else
        {
            res = res - p2.getNum();
            p2 = p2.getPrev();
            c--;
            num = res/c;
        }
    }
    return num == target;
}
} //end of class IntListTwo

```



## סעיף א (3 נקודות)

מה מבצעת השיטה f באופן כללי? הסבירו בקצרה מה השיטה עושה ולא כיצד היא מבצעת זאת. 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, עליכם לכתוב מה המשמעות של הערך שמוחזר מהשיטה f, כשהיא מופעלת על רשימה כלשהי.

השיטה סוכמת את איברי הרשימה ומחזירה את סכום האיברים ברשימה

## סעיף ב (3 נקודות)

מה מבצעת השיטה g באופן כללי? הסבירו בקצרה מה השיטה עושה ולא כיצד היא מבצעת זאת. 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, עליכם לכתוב מה המשמעות של הערך שמוחזר מהשיטה g, כשהיא מופעלת על רשימה כלשהי.

השיטה סוכמת את כמות האיברים ברשימה ומחזירה את כמות האיברים ברשימה

## סעיף ג (2 נקודות)

מה הערך אותו תחזיר השיטה what כשנפעיל אותה על הרשימה {2, 3, 8, 14, 15, 35} והערך target = 10?  
התשובה היא  
true

## סעיף ד (2 נקודות)

מה הערך אותו תחזיר השיטה what כשנפעיל אותה על הרשימה {2, 3, 8, 14, 15, 35} והערך target = 2.5?  
התשובה היא  
true

## סעיף ה (2 נקודות)

מה הערך אותו תחזיר השיטה what כשנפעיל אותה על הרשימה {2, 3, 8, 14, 15, 35} והערך target = 8.5?  
התשובה היא  
False

## סעיף ו (5 נקודות)

מה מבצעת השיטה what באופן כללי כשהיא מופעלת על רשימה מסודרת בסדר עולה? הסבירו בקצרה מה השיטה עושה ולא כיצד היא מבצעת זאת. 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, עליכם לכתוב מה המשמעות של הערך שמוחזר מהשיטה what, כשהיא מופעלת על רשימה כלשהי. התייחסו למקרי קצה.

השיטה מחפשת את הרצף שהממוצע שלו שווה לtarget. במידה והוא מוצא יוחזר אמת במידה ולא יוחזר שקר.

## בהצלחה