

כ"ה באב תשע"ט

מס' שאלון - 466

26

באוגוסט 2019

סמסטר 2019ב

93 מ'סמוע

20441 / 4

שאלון בחינת גמר

20441 - מבוא למדעי המחשב ושפת Java

משך בחינה: 3 שעות

בשאלון זה 13 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.

בהצלחה !!!

החזירו

למשגיח את השאלון

וכל עזר אחר שקיבלתם בתוך מחברת התשובות

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

שאלה 1 (25 נקודות)

נתון מערך חד-ממדי a המלא במספרים שלמים חיוביים, ומספר שלם חיובי num . כתבו שיטה סטטית בוליאנית רקורסיבית, המקבלת כפרמטרים את המערך a ואת המספר num . השיטה צריכה להחזיר $true$ אם קיימת תת-קבוצה של איברי המערך, שסכום האיברים שלה שווה ל- num , שמקיימת את התנאים הבאים:

1. כוללת איברי המערך ללא חזרות. כלומר, אסור לקחת איבר מהמערך יותר מפעם אחת.
2. אינה כוללת שלושה איברים סמוכים במערך. כלומר, אסור להכניס לתת-קבוצה גם את האיבר באינדקס i , גם את האיבר באינדקס $i+1$ וגם את האיבר באינדקס $i+2$, לכל i .
אם אין תת-קבוצה במערך שמקיימת את התנאים האלו, השיטה תחזיר את הערך $false$.

לדוגמא,

אם המערך הוא $a = \{5, 4, 2, 1, 3\}$

אז:

- אם $num = 0$, השיטה תחזיר $true$ (תת-קבוצה ריקה).
- אם $num = 8$, השיטה תחזיר $true$ (המספרים $4 + 3 + 1 = 8$ או המספרים $5 + 3 = 8$).
- אם $num = 9$, השיטה תחזיר $true$ (המספרים $4 + 5 = 9$ אמנם הם רצופים במערך, אבל זו לא שלושה, אלא רק זוג מספרים רצופים).
- אם $num = 2$, השיטה תחזיר $true$ (המספר 2).
- אם $num = 11$, השיטה תחזיר $false$ ($4 + 5 + 2 = 11$, אבל הם שלושה רצופים, וגם $5 + 2 + 1 + 3 = 11$ יש שלושה רצופים).
- אם $num = 17$, השיטה תחזיר $false$ (אין במערך מספרים שסכומם 17).

חתימת השיטה היא:

```
public static boolean isSum (int[] a, int num)
```

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).

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)

נתונים שני מערכים חד-ממדיים road1 ו-road2 מלאים במספרים שלמים חיוביים באורך N. כל מערך מתאר נסיעה בכביש הבנוי מ-N מקטעים, כך שכל איבר i במערך שווה למספר הדקות הנדרשות לעבור מקטע i של הכביש. שני הכבישים מתחילים במקום A ומסיימים במקום B, ובכל מקטע אפשר לעבור מכביש אחד למשנהו.

נהג מתחיל מכביש מסוים (אחד מהשניים), כלומר עובר בו לפחות מקטע אחד, ומותר לו לעבור לכביש השני רק פעם אחת לכל היותר. כלומר, יתכן שהוא מתחיל בכביש 1 ונוסע עד סופו בלי לעבור בכלל לכביש 2, יתכן שהוא מתחיל בכביש 2 ונוסע עד סופו בלי לעבור בכלל לכביש 1 ויתכן שהוא מתחיל בכביש אחד (1 או 2) ועובר פעם אחת לכביש השני (2 או 1).

כתבו שיטה סטטית, המקבלת כפרמטרים את שני המערכים, ומחזירה את מספר הדקות במסלול המהיר ביותר שהנהג יכול לנסוע.

לדוגמא, אם נתונים המערכים הבאים:

	0	1	2	3	4	5	6	7
road1 =	5	4	5	8	12	9	9	3
road2 =	7	3	3	12	10	2	10	7

המסלול המהיר ביותר הוא זה הצבוע באפור. הנהג נוסע שישה מקטעים בכביש 2 ואז עובר לכביש 1 ונוסע בו את שני המקטעים האחרונים.

השיטה תחזיר את הערך 49 שהוא הזמן של המסלול המהיר ביותר. למשל, אם הנהג היה נוסע כל הזמן בכביש 1, ולא עובר לכביש 2 בכלל, זמן הנסיעה היה 55. אם הנהג היה נוסע כל הזמן בכביש 2, ולא עובר לכביש 1 בכלל, זמן הנסיעה היה 54. אם היה מתחיל בכביש 1 ועובר לכביש 2 אחרי 4 מקטעים, זמן הנסיעה היה 51, וכן הלאה.

חתימת השיטה היא:

```
public static int shortestRoad (int [] road1, int [] road2)
```

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.

מה סיבוכיות זמן הריצה של השיטה שכתבתם? הסבירו תשובתכם.

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 (17 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, what ו- secret הבאות:

```
public static boolean f (Node root, int num)
{
    if (root == null)
        return false;
    if (root.getNumber() == num)
        return true;
    return f(root.getLeftSon(), num) ||
           f(root.getRightSon(), num);
}
```

```

public static int what(Node root, int x)
{
    if (f(root, x))
        return secret (root, x);
    return 0;
}

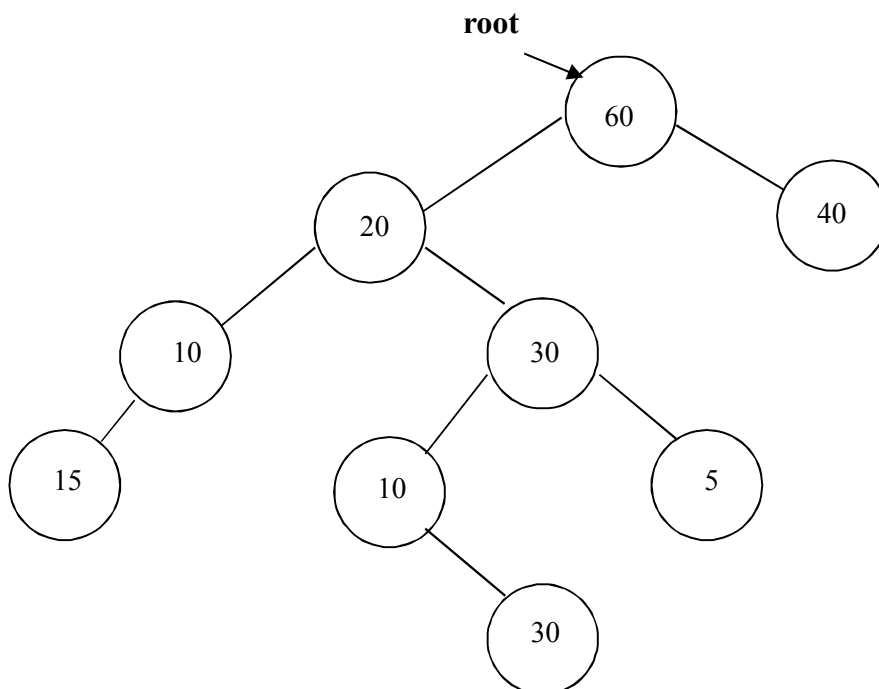
public static int secret(Node root, int x)
{
    if (root == null)
        return 0;

    if ((root.getLeftSon() != null &&
        (root.getLeftSon()).getNumber() == x) ||
        (root.getRightSon() != null &&
        (root.getRightSon()).getNumber() == x))
    {
        return root.getNumber()+
            secret(root.getLeftSon(), x)+
            secret(root.getRightSon(), x);
    }

    return secret(root.getLeftSon(), x)+
        secret(root.getRightSon(), x);
}

```

נתון העץ הבינרי הבא, ששורשו הוא root



ענו על הסעיפים הבאים:

(א) מה תחזיר השיטה `f` בעקבות הקריאה `BinaryTree.f(root, 30)`? (2 נק')

התשובה היא:

(ב) מה מבצעת השיטה `f` באופן כללי כשהיא מקבלת כפרמטרים שורש של עץ בינרי `root` ומספר `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. כלומר, כתבו בקצרה מה משמעות הערך שהשיטה מחזירה כשהיא מקבלת שורש של עץ ומספר כלשהו. (3 נק')

התשובה היא:

(ג) איזה ערך תחזיר השיטה `secret` בעקבות הקריאה `BinaryTree.secret(root, 25)`? (3 נק')

התשובה היא:

איזה ערך תחזיר השיטה `what` בעקבות הקריאה `BinaryTree.what(root, 10)`? (3 נק')

התשובה היא:

(ד) מה מבצעת השיטה `what` באופן כללי כשהיא מקבלת כפרמטרים שורש של עץ בינרי `root` ומספר כלשהו `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. כלומר, כתבו בקצרה מה משמעות הערך שהשיטה מחזירה כשהיא מקבלת שורש של עץ ומספר כלשהו. התייחסו למקרי קצה. (6 נק')

התשובה היא:

שאלה 4 (15 נקודות)

נתון פרויקט שהוגדרו בו המחלקות האלו: A, B, C, D, E.

להלן ינתנו פרטים שונים שיעזרו לכם להבין מהם הקשרים בין המחלקות:

במחלקה נוספת באותו פרויקט, בשם Driver נכתב קטע הקוד הבא בשיטה main אשר יוצר

את העצמים הבאים, והוא תקין.

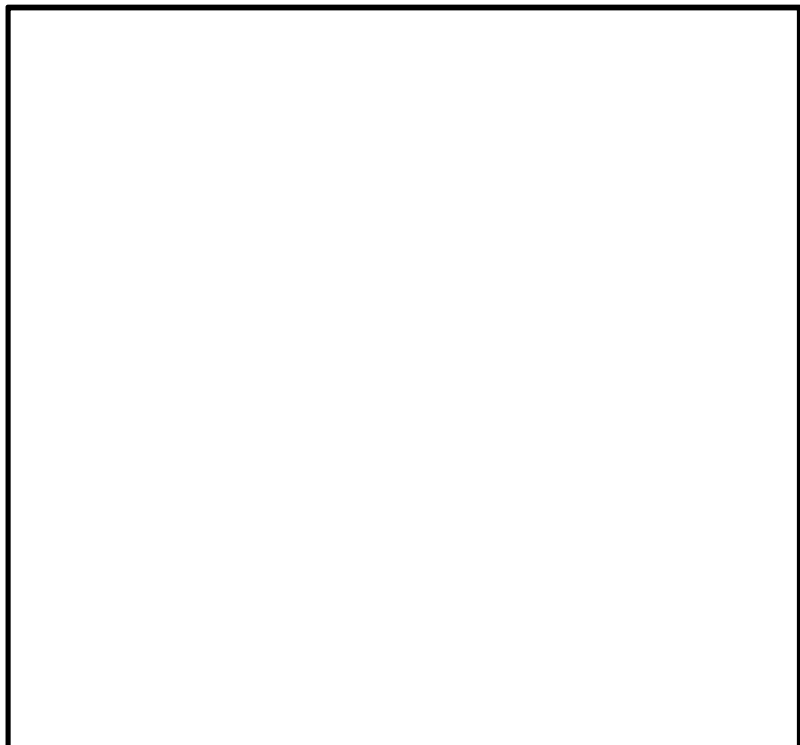
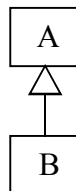
```
B b1 = new A();  
B b2 = new C();  
A a1 = (C) b2;  
D d1 = new C();  
B b3 = new B();  
D d2 = new A();  
D d3 = new B();  
E e1 = new E();  
D e2 = new E();  
B e3 = new E();  
C c1 = new C();
```

סעיף א

ציירו עץ ירושה אפשרי של כל המחלקות:

לדוגמא, אם המחלקה B יורשת ישירות מהמחלקה A, ציירו זאת כך:

ציירו כאן: (5 נק')



סעיף ב

במחלקות הבאות הוגדרו השיטות הבאות:

במחלקה A

```
public void aaa() {  
    System.out.println("A - aaa()");  
    super.aaa();  
}
```

במחלקה B

```
public void aaa() {  
    System.out.println("B - aaa()");  
    ccc();  
}
```

```
public void ccc() {  
    System.out.println("B - ccc()");  
}
```

במחלקה C

```
public void bbb() {  
    System.out.println("C - bbb()");  
    super.ccc();  
}
```

```
public void ccc() {  
    System.out.println("C - ccc()");  
}
```

במחלקה E

```
public void bbb() {  
    aaa();  
    System.out.println("E - bbb()");  
}
```

```
public void ccc() {  
    System.out.println("E - ccc()");  
}
```


בהמשך השיטה main במחלקה Driver נכתבו הפקודות הבאות, והן **תקינות**.

```
B    var1 = new A();  
C    var2 = new C();  
B    var3 = new E();  
B    var4 = new C();  
Object var5 = new A();
```

לכל אחת מהפקודות הבאות, סמנו בצורה ברורה ומפורשת אם היא תקינה או גורמת לשגיאת קומפילציה או גורמת לשגיאת ריצה, ומהי הבעיה. (10 נק')
אם הקריאה לשיטה תקינה כתבו מה יודפס בעקבות קריאה זו.
הפקודות אינן תלויות זו בזו.

1. `var1.aaa();`

תקין / שגיאת קומפילציה / שגיאת זמן ריצה
הסבר או פלט:

2. `var1.ccc();`

תקין / שגיאת קומפילציה / שגיאת זמן ריצה
הסבר או פלט:

3. `var2.aaa();`

תקין / שגיאת קומפילציה / שגיאת זמן ריצה
הסבר או פלט:

4. `var2.bbb();`

תקין / שגיאת קומפילציה / שגיאת זמן ריצה
הסבר או פלט:

5. `var3.aaa();`

תקין / שגיאת קומפילציה / שגיאת זמן ריצה
הסבר או פלט:

6. `var3.bbb();`

תקין / שגיאת קומפילציה / שגיאת זמן ריצה
הסבר או פלט:

7. `var4.aaa();`

תקין / שגיאת קומפילציה / שגיאת זמן ריצה
הסבר או פלט:

8. `var5.aaa();`

תקין / שגיאת קומפילציה / שגיאת זמן ריצה
הסבר או פלט:

9. `((C)var5).aaa();`

תקין / שגיאת קומפילציה / שגיאת זמן ריצה
הסבר או פלט:

10. `((E)var3).bbb();`

תקין / שגיאת קומפילציה / שגיאת זמן ריצה
הסבר או פלט:

שאלה 5 (18 נקודות)

נתונה המחלקה הבאה, המייצגת איבר ברשימה:

```
public class IntNode {
    private int _value;
    private IntNode _next;

    public IntNode(int val, IntNode n) {
        _value = val;
        _next = n;
    }

    public int getValue() {
        return _value;
    }
    public IntNode getNext() {
        return _next;
    }

    public void setValue(int v) {
        _value = v;
    }
    public void setNext(IntNode node) {
        _next = node;
    }
}
```

נתונה רשימה מקושרת של מספרים שלמים, הממומשת בעזרת המחלקה IntList שלהלן:

```
public class IntList
{
    private IntNode _head;
    public IntList( ) {
        _head = null;
    }

    public IntList(IntNode h ) {
        _head = h;
    }

    // המשך המחלקה בעמוד הבא
```

```

public boolean what (IntList list)
{
    IntNode p = _head;
    IntNode q = list._head;

    while (p!=null && q!=null)
    {
        if (p.getValue() == q.getValue())
        {
            p = p.getNext();
            q = q.getNext();

            if (q == null)
                return true;
        }
        else
        {
            p = p.getNext();
            q = list._head;
        }
    }
    return false;
}

public int something()
{
    int temp = 1;
    int cnt = 1;
    IntNode p = _head, q = p;
    p = p.getNext();
    while (p!=null )
    {
        if (p.getValue() * q.getValue() < 0)
        {
            cnt++;
            temp = Math.max(temp, cnt);
        }
        else
            cnt=1;
        q = p;
        p = p.getNext();
    }
    return temp;
}

. . .          // other methods
}

```

אתם יכולים להניח שהרשימה מלאה במספרים שלמים.

בטענות להלן, נסמן את איברי הרשימה כמספרים מופרדים בפסיקים, בתוך סוגריים מסולסלים.

כך לדוגמא, נסמן $\{-4, 9, 12, 21\}$ את הרשימה $-4 \rightarrow 9 \rightarrow 12 \rightarrow 21 \rightarrow \text{null}$

סעיף א (3 נקודות)

נתונה הרשימה $\{2, 6, 3, 4, 7, 4, 6, 1, 3\}$. אם נפעיל עליה את השיטה **what** ונעביר לה

כפרמטר את הרשימה $\{6, 1, 9\}$, איזה ערך תחזיר השיטה?

סעיף ב (5 נקודות)

מה מבצעת השיטה **what** 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 **what**, כשהיא מופעלת על רשימה כלשהי. התייחסו למקרי קצה.

סעיף ג (4 נקודות)

נתונה הרשימה $\{2, -6, -3, 4, -7, 4, 6, -1, 3\}$. אם נפעיל עליה את השיטה **something**, איזה ערך תחזיר השיטה?

סעיף ד (6 נקודות)

מה מבצעת השיטה **something** 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 **something**, כשהיא מופעלת על רשימה כלשהי. התייחסו למקרי קצה.

ב ה צ ל ח ה !