



מספר החליטור הנבחר
רשום את כל השם המפורט

האוניברסיטה
הפתוחה



כ' באדר א' תשפ"ב

מס' שאלון - 473 21
בפברואר 2022

מס' מועד 89

סמסטר 2022א

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 לגרום 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 1-8, 12-7.
אין להכניס חומר מודפס נוסף או חומר אחר מכל סוג.
אין להכניס מחשב או מחשבון או מכשיר אלקטרוני מכל סוג שהוא.

בהצלחה !!!

החזיר

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)

בחינתן שתי מחרזות תווים str1 ו-str2. נאמר שהמחרזות str2 מופיעה בדילוגים קבועים בתוך המחרזות str1, אם כל התווים שנמצאים במחרזות str2 מופיעים בתוך המחרזות str1, באותו הסדר בו הם מופיעים ב-str2, וגם, בין תו לתו של str2 בתוך str1 יש אותו מספר תווים שלא קשורים למחרזות str2.

לדוגמא,

- המחרזות str2 = "abc" מופיעה בתוך המחרזות str1 = "adbrcfa" בדילוגים קבועים של 2. האותיות של str2 מודגשות.
- המחרזות str2 = "abcd" מופיעה בתוך המחרזות str1 = "abcdfa" בדילוגים קבועים של 1. האותיות של str2 מודגשות.
- המחרזות str2 = "abc" מופיעה בתוך המחרזות str1 = "adbfbaagcxabcd" פעמיים בדילוגים קבועים. פעם אחת בדילוגים של 4 (החל מהתו הראשון במחרזות str1), ופעם שניה בדילוגים של 1 (מיד אחרי התו 'x'). האותיות של str2 מודגשות.
- המחרזות str2 = "abc" לא מופיעה בתוך המחרזות str1 = "cbdaadbrsdcfa" בדילוגים קבועים.

שימו לב, ברור ש-str1 צריכה להיות ארוכה או שווה ל-str2. צריך לבדוק זאת!

בשאלה זו נמצא את הדילוג הקבוע בו נמצאת המחרזות str2 בתוך str1 (אם יש כזה). נתייחס רק למחרזות שמתחילות באותו תו.

סעיף א: (15 נקודות)

כתבו שיטה סטטית רקורסיבית בוליאנית בשם isJump המקבלת כפרמטרים שתי מחרזות תווים str1, str2 ומספר שלם חיובי step, ומחזירה true אם המחרזות str2 מופיעה בדילוגים קבועים של step בתוך המחרזות str1 החל מהתו הראשון בשתי המחרזות, ו-false אחרת.

לדוגמא:

- str1 = "adbrcfa", str2 = "abc" אם step = 2 יוחזר true.
- str1 = "cbdaadbrcfa", str2 = "abc" אם step = 2 יוחזר false למרות שהמחרזות str2 מופיעה בדילוג קבוע של 2 בתוך str1, אבל הדילוג לא מתחיל בתו הראשון של str1.
- str1 = "adcfbaagcxabcd", str2 = "abc" אם step = 4 יוחזר true, אבל אם step = 1 יוחזר false כי הדילוג הזה לא מתחיל בתו הראשון של str1.
- str1 = "abcfbaagcxabcd", str2 = "abc" אם step = 1 יוחזר true, וגם אם step = 4 יוחזר true. כי שני הדילוגים מתחילים בתו הראשון של str1.

חתימת השיטה היא:

```
public static boolean isJump (String str1, String str2, int step)
```

סעיף ב: (10 נקודות)

כתבו שיטה סטטית **רקורסיבית** `strStep` המקבלת כפרמטרים שתי מחרוזות `str1` ו-`str2` ומחזירה את הערך של הדילוג הקבוע בו נמצאת המחרוזת `str2` בתוך `str1` **החל מהתו הראשון של `str1`**. אם המחרוזת `str2` לא נמצאת בדילוג קבוע בתוך `str1`, או שהדילוג לא מתחיל מהתו הראשון של `str1`, השיטה תחזיר -1. אם המחרוזת `str2` נמצאת בכמה דילוגים קבועים החל מהתו הראשון, השיטה תחזיר את ערך הדילוג הראשון בו `str2` נמצאת ב-`str1`.

לדוגמא, בדוגמאות של סעיף א –

1. `str1 = "adbrcfa"`, `str2 = "abc"` , השיטה `strStep` תחזיר את הערך 2.
2. `str1 = "cbdadbrcf"`, `str2 = "abc"` , השיטה `strStep` תחזיר את הערך 1-.
3. `str1 = "adcfbaagcxabcd"`, `str2 = "abc"` , השיטה `strStep` תחזיר את הערך 4.
4. `str1 = "abcfbaagcxabcd"`, `str2 = "abc"` , השיטה `strStep` תחזיר את הערך 1. (שימו לב, יש כאן שני דילוגים, אבל 1 הוא הראשון ולכן הוא זה שיוחזר)

חתימת השיטה היא:

```
public static int strStep(String str1, String str2)
```

שימו לב, מותר לכם להשתמש בשיטה `isJump` מסעיף א, גם אם לא פתרם אותה.

השיטות מהמחלקה `String` בהן מותר לכם להשתמש הן:

- `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`.

כל השיטות שתכתבו צריכה להיות רקורסיבית 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 שהמחרוזות אינן `null` ואינן ריקות. אין צורך לבדוק זאת!
- 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 חיוביים ושליילים ללא אפס! (אין צורך לבדוק זאת). המערך אינו ממוין!

כתבו שיטה סטטית `longestSubarray` המקבלת את המערך כפרמטר ומחזירה את אורכו של התת-מערך (הרציף) הארוך ביותר שכל איבריו הם לסירוגין מספר חיובי ומספר שלילי. השיטה צריכה גם להדפיס מה האינדקס ההתחלתי ומה האינדקס הסופי של התת-מערך המבוקש. אם יש במערך יותר מתת-מערך אחד באורך מקסימלי, יודפסו האינדקסים של התת-מערך הראשון מביניהם.

לדוגמא:

- עבור המערך $\{-1, 1, -1, -5, 2, 2\}$ השיטה תחזיר 3 (האיברים של התת-מערך הזה מודגשים) ותדפיס משהו כעין זה:

Starting index = 0 Ending index = 2

- עבור המערך $\{3, 3, 2, -7, 2, 1, 1, -2, -2\}$ השיטה תחזיר 3 ותדפיס משהו כעין זה:

Starting index = 2 Ending index = 4

- עבור המערך $\{1, 2, 3, 4, 5, 4\}$ השיטה תחזיר 1 (אין תת-מערך באורך גדול מ-1 שיש בו איברים מתחלפים. לכן כל אחד מהאיברים במערך הוא תת-מערך באורך 1) ותדפיס משהו כעין זה:

Starting index = 0 Ending index = 0

- עבור המערך $\{1, -2, 3, -4, -5, 4, 2, -4, 6, -2\}$ השיטה תחזיר 4 (יש שני תת-מערכים באורך 4, אחד מאינדקס 0 עד אינדקס 3, והשני מאינדקס 6 עד אינדקס 9) ותדפיס משהו כעין זה:

Starting index = 0 Ending index = 3

חתימת השיטה היא:

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

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

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.

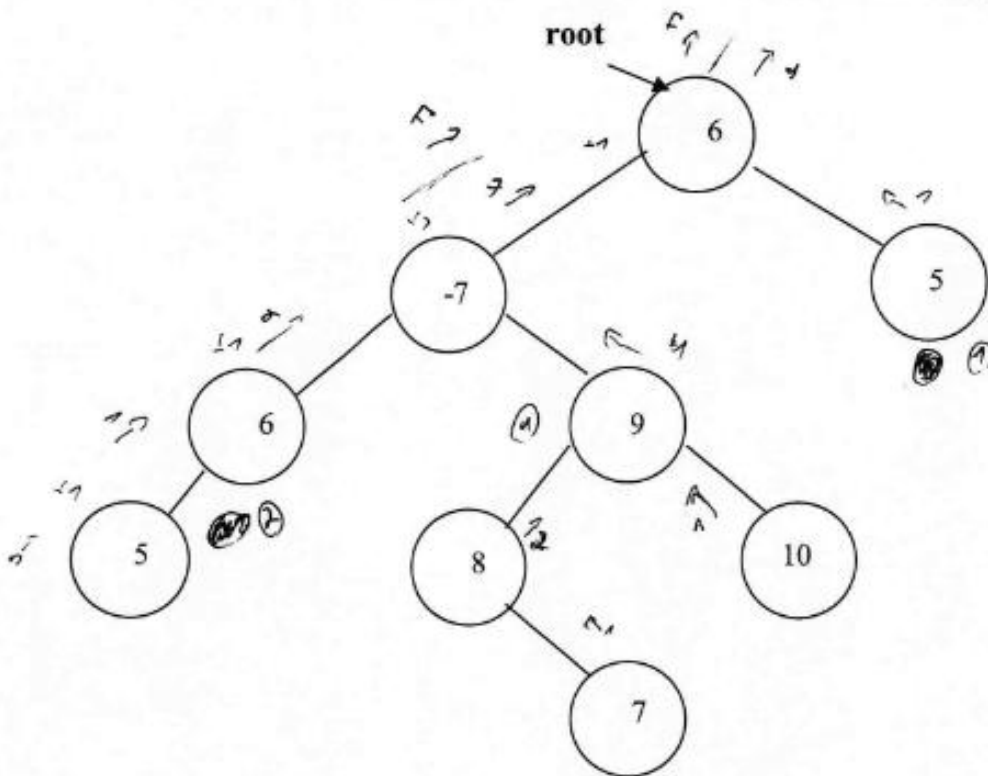
בין השיטות נתונות השיטות something, what ו-something, what - secret הבאות, שכל אחת מהן מקבלת שורש של עץ בינרי.

```
public static int what(Node root)
{
    if (root == null)
        return 0;
    return what(root.getLeftSon()) + 1 +
           what(root.getRightSon());
}

public static boolean something(Node node)
{
    if (node == null)
        return true;
    if (node.getNumber() <= 0)
        return false;
    return something(node.getLeftSon()) &&
           something(node.getRightSon());
}
```

```
public static int secret(Node root)
{
    if (something(root))
        return what(root);
    return Math.max(secret(root.getLeftSon()),
                    secret(root.getRightSon()));
}
```

נתון העץ הבינרי הבא, ששורשו הוא root



ענו על הסעיפים הבאים:

2) נקי (א) איזה ערך תחזיר השיטה **what** בעקבות הקריאה `BinaryTree.what(root)`?

התשובה היא:

9



3) נקי (ב) מה מבצעת השיטה **what** 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:

בסיסית, המטרה של `what` היא לחזיר את הערך המקסימלי
בצורה, אם באזורים null יחזיר אפס



3 נקי' (ג) איזה ערך תחזיר השיטה something בעקבות הקריאה

!BinaryTree.something(root)

התשובה היא:
False



4 נקי' (ד) מה מבצעת השיטה something באופן כללי כשהיא מקבלת כפרמטר שורש של עץ בינרי? 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:

טונקציה ג'נרלית שרשם על עץ בינרי ומחזירה קאמבאל א'בר'ן
מסר'ים ^{חלקי} ~~מסר'ים~~ או ~~אכס~~, ג'נרל ובעל ח'ק יחזיר True
-1

3 נקי' (ה) איזה ערך תחזיר השיטה secret בעקבות הקריאה !BinaryTree.secret(root)

התשובה היא:

4



5 נקי' (ו) מה מבצעת השיטה secret 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:

טונקציה ג'נרלית שרשם על עץ בינרי ומחזירה א'בר'ן
הא'בר'ן א' ג'נרל ע'ר הכי גדול ^{סבל} ~~מסר'ים~~ א'בר'ן ח'לקי או אכס
א'ר לא ג'נרל א'ר א'בר'ן ח'לקי ק'נר יחזיר אכס
שניאה נגרת



19
(3)

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

```
public class Animal
{
    public Animal()
    {
        System.out.println("new Animal");
    }
}

-----

public class Dog extends Animal
{
    protected String _name;
    public Dog(String n)
    {
        _name = n;
        System.out.println("new Dog");
    }
    public void printName()
    {
        System.out.println(_name);
    }
    public void compare(Dog d)
    {
        if (d._name.equals(_name))
            System.out.println("Dog - Yes");
        else
            System.out.println("Dog - No");
    }
}

-----

public class Poodle extends Dog
{
    public Poodle()
    {
        super("Kiki");
        System.out.println("new Poodle");
    }

    public void compare(Dog a)
    {
        System.out.print("Comparing...");
        compare((Animal) a);
    }

    public void compare(Animal a)
    {
        if ((a instanceof Poodle) ||
            (a instanceof Dog) && ((Dog)a)._name.equals("Kiki"))
            System.out.println("Poodle - Positive");
        else
            System.out.println("Poodle - Negative");
    }
}
```



```
public class Dalmatian extends Dog
{
    public Dalmatian()
    {
        super("Kiki");
        System.out.println("new Dalmatian");
    }

    public void compare(Dog a) {
        System.out.print("Comparing...");
        compare((Animal) a);
    }

    public void compare(Poodle a) {
        System.out.print("Comparing Poodle...");
        compare((Animal) a);
    }

    public void compare(Animal a) {
        if ((a instanceof Poodle) ||
            (a instanceof Dog) && ((Dog)a)._name.equals("Kiki"))
            System.out.println("Dalmatian Poodle - Positive");
        else
            System.out.println("Dalmatian Poodle - Negative");
    }

    public void compare(Dalmatian a) {
        if ((a instanceof Dalmatian) ||
            (a instanceof Dog) && ((Dog)a)._name.equals("Kiki"))
            System.out.println("Dalmatian Dal - Positive");
        else
            System.out.println("Dalmatian Dal - Negative");
    }
}
```

כמו כן נתונה המחלקה Driver (באותו פרויקט):

```
public class Driver
{
    public static void main(String [] args)
    {
        Animal a = new Animal();
        Poodle p1 = new Poodle();
        Dog d = new Dalmatian ();
        Animal d2= new Dog("Kiki");
        Dog d3 = new Dog("Max");
        Animal p2 = new Poodle();
        Dog p3 = new Poodle();
        Dalmatian d4 = new Dalmatian();

        // כאן יופיעו הפקודות שבסעיפים הבאים
    }
}
```

בסעיפים הבאים נوسیף פקודות לשיטה main במחלקה Driver. בכל סעיף הניחו שהפקודות החדשות נוספות לאלה שכבר הוספו בסעיפים הקודמים לה. כל סעיף 2 נקודות.
אין צורך לכתוב את מה שיודפס בעקבות הצהרות האובייקטים והקריאות לבנאים.
 בכל סעיף כתבו מה יהיה הפלט שיודפס בעקבות הוספת הפקודות החדשות. (התייחסו לפלט של הפקודות החדשות בלבד, אין צורך לרשום שוב ושוב את הפלט של הפקודות הקודמות). אם לדעתכם לא יודפס דבר, או תהיה שגיאה כלשהי – ציינו את השגיאה שנגרמה (קומפילציה/ריצה) וחשבירו את הסיבה לשגיאה.

1. מה יקרה בעקבות הוספת הפקודה הבאה?

p1.compare(a);

Poodle - Negative



2. מה יקרה בעקבות הוספת הפקודה הבאה?

p1.compare(d2);

Poodle - Positive



3. מה יקרה בעקבות הוספת הפקודה הבאה?

d3.compare(p1);

Dog - No



4. מה יקרה בעקבות הוספת הפקודה הבאה?

d.compare(d4);

~~comparing...~~ Dalmatian Poodle - Positive



5. מה יקרה בעקבות הוספת הפקודה הבאה?

d.compare(p1);

comparing... Dalmatian Poodle - Positive



6. מה יקרה בעקבות הוספת הפקודה הבאה?

((Dog)p2).compare(p3);

comparing... Poodle - Positive



7. מה יקרה בעקבות הוספת הפקודה הבאה?

((Dalmatian)d).compare(a);

~~Dalmatian Poodle - Negative~~
 Dalmatian Poodle - Negative



14
(4)

שאלה 5 (16 נקודות)

נתונה המחלקה Queue המממשת תור, המאחסן מספרים שלמים.

להלן ה-API של המחלקה Queue :

Constructor Summary

Constructors

Constructor and Description

Queue ()

Constructs an empty Queue object.

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method and Description
int	<div>dequeue ()</div> <div>Removes an item from the front of the queue.</div>
void	<div>enqueue (int num)</div> <div>Adds an item to the tail of the queue.</div>
boolean	<div>isEmpty ()</div> <div>Checks whether the queue is empty.</div>

נניח שנוספה למחלקה גם השיטה length() שמחזירה את אורכו של התור (כמה איברים יש בו).

הניחו שהשיטות לעיל ממומשות במחלקה Queue.

במחלקה Queue הוגדרה גם השיטה הסטטית what שמתוארת בעמוד הבא. השיטה מקבלת כפרמטר תור q שמלא במספרים שלמים ומחזירה את התור q.


```
public static Queue what(Queue q)
{
    int length = q.size();
    Queue q1 = new Queue();
    for (int i=0; i<length/2; i++)
    {
        int num = q.dequeue();
        q1.enqueue(num);
    }
    Queue q2 = new Queue();
    for (int i=0; i<length/2; i++)
    {
        int num = q1.dequeue();
        q2.enqueue(num);
        num = q.dequeue();
        q2.enqueue(num);
    }
    for (int i=0; i<length; i++)
    {
        int num = q2.dequeue();
        q.enqueue(num);
    }
    return q;
}
```

סעיף א (4 נק')

אם נפעיל את השיטה **what** על התור {1, 2, 3, 4, 5, 6} כאשר 1 נמצא בראש התור ו-6 בזנבו איך ייראה התור q המוחזר?
התשובה היא

2, 1, 4, 2, 5, 3, 6



סעיף ב (4 נק')

לאחר הפעלת השיטה **what** עם הפרמטר שהוא תור q שמלא במספרים שלמים, הוחזר התור q הבא:

{8, 4, 7, 3, 6, 2, 5, 1}

(כאשר 8 נמצא בראש התור ו-1 בזנבו).

מה היה התור q שהועבר כפרמטר לשיטה?

התשובה היא

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



סעיף ג (4 נק')

לאחר הפעלת השיטה **what** עם הפרמטר שהוא תור q שמלא במספרים שלמים, הוחזר התור q הבא:

$\{1, 0, 0, 1, 1, 0\}$

(כאשר 1 נמצא בראש התור ו-0 בזנבו).

מה היה התור q שהועבר כפרמטר לשיטה?

התשובה היא

2, 1, 0, 1, 0, 1, 0 ✓

סעיף ד (4 נק')

הציגו תור ובו לפחות 4 איברים שאם השיטה תופעל עליו - התוכנית תעוף בגלל שגיאת זמן ריצה, והסבירו מדוע השיטה עפה.

התשובה היא

2, 1, 2, 3, 4, 5, 6 כל תור שאינו מסביר איבר'ם לא
אז לא יאותרו יוצרים ~~מספר~~ ^{מספר} ובמיוחד מספרים (קב)
ע"א ריצה שניסיון להקים איבר מתוך ריק
 $q_2.length < q.length = length$ ✓

16
(5)

בהצלחה!