



מבוא לתכנות מונחה עצמים תשפ"ד - בחינת סיכום מועד א'

21/02/2024

- **שימו לב:** את התשובות יש לכתוב אך ורק במחברת הבחינה!
- **שימו לב:** יש להקפיד על תיכון (design) בסגנון מונחה עצמים. אין להסתפק ב"תוכנה שעובדת" בלבד!
- משך הבחינה 3 וחצי שעות.
- יש לפתור את כל השאלות.
- בחינה בחומר פתוח.
- בטרם תיגשו לפתרון ודאו כי הטופס מכיל 6 עמודים ו- 4 שאלות.
- מרצים: ד"ר קרן כהן, ד"ר אחמד ג'בארה.

בהצלחה!!!

שאלה מס' 1 (15 נקודות)

1.1 השיטה הבאה מקבלת מחרוזת ושני מספרים שלמים, ומחזירה מחרוזת חדשה.

```
public static String doSomething(String s,int a, int b) {  
    String s1=s.substring(0,a);  
    String s2=s.substring(b,s.length());  
    for(int i=b-1 ; i >=a ; i-- ) {  
        s1+=s.charAt(i);  
    }  
    return s1+s2;  
}
```

מה יהיה הפלט אם נפעיל את השיטה כך:

```
System.out.println(doSomething("abcmatsdefg",3,7));
```

1.2 הסבירו בכמה מילים מה עושה השיטה doSomething מהסעיף הקודם.

1.3 האם הפקודה הבאה חוקית:

```
doSomething( "abcdefghij" , Math.pow(2,2) , Math.pow(2,3) );
```

אם כן, כתבו מה יהיה הפלט. אם לא, הסבירו מהי השגיאה.

שאלה מס' 2 (20 נקודות)

2.1 כתבו שיטה סטטית שמקבלת מחרוזת (ניתן להניח שיש בה אותיות קטנות בלבד) ומחזירה אמת אם במחרוזת לא קיימים תווים סמוכים זהים.

דוגמאות:

אם המחרוזת היא "abacb" השיטה תחזיר אמת. כי אין שני תווים רצופים זהים.

אם המחרוזת היא "xyxxzx" השיטה תחזיר שקר. יש תו שחוזר לפחות פעמיים ברצף.

אם המחרוזת היא "abcc" השיטה תחזיר שקר. התו c מופיע פעמיים ברצף.

2.2 כתבו שיטה סטטית שמקבלת משפט s. השיטה תחזיר משפט שבו מילים מהמשפט s אך בכל מילה שבה יש רצף של

תווים סמוכים זהים, הרצף יקוצר כך שלא יהיו תווים רצופים זהים. למשל, אם מילה מתוך s נראית כך: "aabccc" אז אחרי

קיצורה היא תראה כך: "abc" (ראו דוגמאות למטה).

הערות: ניתן להניח שכל התווים הם אותיות קטנות בלבד, ושהמילים במחרוזת s מופרדות ברווח בודד.

ניתן לכתוב שיטות עזר.

דוגמאות:

אם המשפט הוא "abc aabc xxzzyy abcabc" השיטה תחזיר את המשפט "abc abc xzy abcabc"

אם המשפט הוא "a cccc abccaaa" השיטה תחזיר את המשפט "a c abca"

3.1.1 עקבו אחרי ביצוע התוכנית הבאה וציינו את הפלט שמתקבל

3.1.2 על סמך הפלט שהתקבל בסעיף 3.1.1 והבנתכם לתוכנית, ציינו מה מבצעת התוכנית באופן כללי

3.1.3 תנו דוגמה למערך דו-מימדי עבורו תוכנית זו לא תיפעל (תעבור קומפילציה אך תיפול בזמן הריצה).

```
public class Program {
    private static int[][] data = { { 1, 2, 3, 4 }, { 5, 6, 7, 8 },
                                     { 9, 10, 11, 12 }, { 13, 14, 15, 16 } };
    public static void main(String[] args) {
        int k = 0, j = 0, m, n;
        m = data.length;
        n = data[0].length;
        while (k < m && j < n) {
            for (int i = j; i < n; i++)
                System.out.print(data[k][i] + " ");
            k++;
            for (int i = k; i < m; i++)
                System.out.print(data[i][n - 1] + " ");
            n--;
            if (k < m) {
                for (int i = n - 1; i >= j; i--)
                    System.out.print(data[m - 1][i] + " ");
                m--;
            }
            if (j < n) {
                for (int i = m - 1; i >= k; i--)
                    System.out.print(data[i][j] + " ");
                j++;
            }
        }
    }
}
```

להלן המחלקות Point, Line, Polygon אשר מייצגות את המחלקות נקודה במישור, קו ישר ומצולע.

```

public class Point {
    private int x, y;
    public Point(int x, int y) {
        this.x = x;
        this.y = y;
    }
    public boolean unknown( 1 ) {
        return this.x == p.x && this.y == p.y;
    }
}

public class Line {
    private Point start, end;
    public Line(int a, int b, int c, int d) {
        2
    }
    public boolean unknown(Line ln) {
        return end.unknown(ln.start);
    }
}

public class Polygon {
    private Line[] lines;
    public Polygon(Line[] lines) {
        this.lines = lines;
    }
    public boolean unknown() {
        3
        return false;

        int i;
        for (i = 0; i < lines.length; i++)
            if (!lines[i].unknown(lines[(i + 1) % lines.length]))
                return false;

        return true;
    }
}

```

```

public class Program {
    public static void main(String[] args) {
        Polygon p = new Polygon(new Line[] { new Line(1, 4, 3, 4), new Line(3, 4, 5, 2),
                                                new Line(5, 2, 4, 1), new Line(4, 1, 1, 1),
                                                new Line(1, 1, 1, 4) });
        System.out.println(p.unknown());
    }
}

```

3.2.1 השלימו את הקוד החסר (מלבנים 1,2,3) כך שהקוד יתקמפל וירוץ ויהיה בעל משמעות ביחס לשאר הקוד. שימו לב לכך שמלבן יכול להיות ריק, שורת קוד אחת או מספר שורות קוד.

3.2.2 ציירו את העצמים שנוצרים בעת הרצת השיטה הראשית main וציינו את הפלט שמתקבל

3.2.3 תארו במלים שלכם את פעולתה של unknown אשר מוגדרת במחלקה Polygon.

שאלה מס' 4 (30 נקודות)

המחלקה Patient מייצגת חולה שהגיע לבית חולים לקבלת טיפול. חולה מיוצג על ידי שמו, גילו ומספר שמייצג את מידת הדחיפות לקבלת טיפול – ככל שמספר זה גדול יותר החולה מקבל עדיפות גבוהה יותר.

4.1 הגדירו מחלקה בשם Patient לייצוג חולה על פי התיאור הנ"ל. הגדרת המחלקה תכלול כותרת המחלקה, תכונותיה ובנאים מתאימים: בנאי ראשון מקבל את כל המידע והבנאי השני מקבל את כל המידע למעט "דחיפות הטיפול" שתאוחל ל 1. בנוסף, עליכם להשתמש בבנאי הראשון מתוך הבנאי השני ולא לבצע אתחול ישיר לתכונות בתוך הבנאי השני.

4.2 הגדירו שיטה בשם compare אשר קובעת לאיזה מבין שני חולים נתונים יש עדיפות גבוהה יותר בקבלת טיפול.

- אם מידת הדחיפות של החולה הנוכחי גדולה יותר ממידת הדחיפות של החולה האחר יוחזר ערך חיובי

- אם מידת הדחיפות של החולה הנוכחי קטנה יותר מאשר מידת הדחיפות של החולה האחר יוחזר ערך שלילי

- אם מידת הדחיפות שווה אצל שני החולים ההחלטה תתקבל על פי הגיל. החולה שגילו מבוגר יותר יקבל עדיפות. כלומר, אם הגיל של החולה הנוכחי גדול יותר יוחזר ערך חיובי ואם גילו קטן יותר יוחזר ערך שלילי.

- במידה ומידת הדחיפות של שני החולים שווה וגם הגילאים, יוחזר אפס.

4.3 המחלקה PQueue מייצגת תור חולים שהגיעו לקבלת טיפול. מספר החולים המקסימלי שיכולים לעמוד בתור נקבע בעת הקמת התור. הגדירו את המחלקה PQueue לפי התיאור הנ"ל.

4.4 ממשו שיטה בשם nextPatient אשר מוצאת את החולה הבא בתור שאמור לקבל טיפול. השיטה תמחוק את החולה מהתור ותצופף את תאי התור לאחר המחיקה. החולה שיימצא הוא החולה בעל העדיפות הגבוהה ביותר על פי ההגדרה שניתנה בסעיף 4.2. אם יש יותר מחולה אחד בעל עדיפות גבוהה אז החולה שהגיע קודם (על פי מיקומו בתור) הוא זה שיטופל. במידה והתור ריק השיטה תחזיר null.

4.5 כתבו שיטה בשם nextPatients שמחזירה תור חדש שמכיל את כל החולים בעלי עדיפות מקסימלית שווה לקבלת טיפול. אם אין כאלה (כי התור הקיים ריק) יוחזר תור חדש ריק. שימו לב שגודל התור החדש חייב להיות בדיוק כמספר החולים שעונים על התנאי.