

## מבוא למדעי המחשב 202-1-101-1

סמסטר א' תשס"ד

בוהן שני

מר בועז בן-משה  
פרופ' דניאל ברנד  
פרופ' מיכאל קודיש  
דר' יצחק רוזן

משך הבוהן – שעתיים.  
חומר עזר - אסור.  
אין להשתמש במחשבון.

במבחן זה 5 שאלות, בניקוד המסתכם ב- 100 נקודות. ענו על כל השאלות.

אנא רשמו את תשובותיכם בדף התשובות בלבד. המחברת שקיבלתם היא **מחברת טיוטה והיא לא תימסר כלל לבדיקה**. בסיום הבחינה נשמור אך ורק את דף התשובות. כל שאר החומר יועבר לגריסה. הקפידו לרשום בדף התשובות גם את מספר הנבחן ומספר החדר שבו אתם נבחנים.

בשאלות התכנות, מספר השורות העומדות לרשותכם בדף התשובות רומז על אורך הקוד הנדרש. הקפידו על כתב יד ברור. תשובות מסורבלות או ארוכות מדי לא יזכו בניקוד מלא. **אין צורך להעתיק את שורות הקוד הנתונות בשאלון לדף התשובות.**

**בהצלחה !**

### שאלה 1: (25 נקודות)

נתונה המחלקה ContinuousFunction המייצגת פונקציה ממשיית רציפה (הקוד מצורף למטה). המחלקה בעלת בנאי ריק, ושיטה `public abstract double valueAt(double x)`, אשר מחזירה את ערך הפונקציה בנקודה `x`.

משימותיכם יהיו:

א. השלמת המחלקה LinearFunction המרחיבה את ContinuousFunction ומייצגת פונקציה ליניארית:  $y=ax+b$ . ראו התבנית שלמטה.

ב. השלמת המחלקה AbsValue המרחיבה את ContinuousFunction ומייצגת את פונקציית הערך המוחלט של פונקציה אחרת. ראו התבנית שלמטה.

ג. השלמת השיטות (במחלקה ContinuousFunction)

```
private double eq1(ContinuousFunction f, double x1, double x2)
private double eq2(ContinuousFunction f, double x1, double x2)
```

אשר מקבלות פונקציה רציפה `f` ושני מספרים `x1, x2`, כך שמתקיים:  
(`this.valueAt(x1) ≥ f.valueAt(x1)`) && (`this.valueAt(x2) ≤ f.valueAt(x2)`)

הפונקציה מחזירה נקודה `x` שהיא קירוב לנקודה בה שתי הפונקציות שוות. ליתר דיוק, הפונקציה מחזירה נקודה `x` כך ש  $|this.valueAt(x) - f.valueAt(x)| < EPS$  (הקבוע EPS מוגדר במחלקה ContinuousFunction).

במשימה זו `eq1` תמומש בצורה רקורסיבית ו-`eq2` תמומש בצורה איטרטיבית. ראו התבנית שלמטה.

התכנית הבאה מהווה דוגמא לשימוש בקוד שתכתבו:

```
public static void main(String[] a) {
    LinearFunction myLin = new LinearFunction(3,-5); // the function 3x-5
    AbsValue myAbsLin = new AbsValue(myLin); // the function |3x-5|
    System.out.println (myLin.valueAt(1)+" "+ myAbsLin.valueAt(1)); // prints "-2,2"
    LinearFunction myLin2 = new LinearFunction(2,7); // the function 2x+7
    System.out.println (myLin.eq1(myLin2,0,100)); // prints "12"
}
```

```

public class LinearFunction extends ContinuousFunction {
    private double _a, _b;
        // השלימו את המחלקה בדף התשובות סעיף א
    }

public class AbsValue extends ContinuousFunction {
    private ContinuousFunction _cf;
        // השלימו את המחלקה בדף התשובות סעיף ב
    }

public abstract class ContinuousFunction {
    public static final double EPS = 0.001;
    public ContinuousFunction ( ) { }
    public abstract double valueAt(double x);
    private double eq1(ContinuousFunction f, double x1, double x2) {
        // השלם את השיטה בדף התשובות סעיף ג1 (הגדרה רקורסיבית)
    }
    private double eq2(ContinuousFunction f, double x1, double x2) {
        // השלם את השיטה בדף התשובות סעיף ג2 (הגדרה איטרטיבית)
    }
}

```

## שאלה 2: (20 נקודות)

הגדרה: פירוק של מספר טבעי  $n$  הינו סדרה לא-עולה של מספרים טבעיים שסכומם  $n$ . לדוגמא, להלן הפירוקים של המספר 4:

```
4
3 1
2 2
2 1 1
1 1 1 1
```

הפונקציה `public static void partitions(int n)`, מדפיסה את כל הפירוקים של מספר טבעי  $n$  והינה מוגדרת בעזרת הפונקציה הסטטית הרקורסיבית הבאה אשר אותה יש להשלים:

```
public static void partitions(int n, int max, String prefix)
```

תפקידה של השיטה `partitions(int n, int max, String prefix)` הוא להדפיס את כל אותם הפירוקים של המספר הטבעי  $n$ , אשר בהם המספר הגדול ביותר הינו לכל היותר  $max$ , כאשר לפני כל פירוק מודפסת המחרוזת `prefix`.

רמז: שימו לב שניתן לחלק את הפירוקים של המספר 4 לקבוצות לפי המספר הראשון בסדרת הפירוק. דהיינו, לפי אלו שמתחילים ב- 4, 3, 2, וב-1.

```
public static void partitions(int n) {
    partitions(n,n,"");
}

public static void partitions(int n, int max, String prefix) {
    if(n == 0)
        System.out.println(prefix);
    else
        for(int i=Math.min(n,max);i>=1;i=i-1) {
            // השלימו בדף התשובות
        }
}
```

### שאלה 3: (25 נקודות)

בשאלה זו עליך להשלים את המחלקה ShapeContainer, המייצגת אוסף של צורות, שונות ביניהן.

הניחו שמוגדרת כבר המחלקה Shape, המייצגת צורה. למחלקה שיטת equals המשווה לוגית בין שתי צורות. דהיינו, השיטה מחזירה אמת אם הפרמטר מייצג אותה צורה כמו עצם המפתח (ושקר אחרת):

```
public boolean equals(Shape s);
```

במחלקה ShapeContainer מוגדרות כבר השיטות הבאות:

- בנאי ריק.
- השיטה public int size(), המחזירה את מספר הצורות באוסף.
- השיטה public Shape at(int index), המחזירה את הצורה הנמצאת במקום ה index בקבוצה.
- שיטה פרטית private void addP(Shape s), המוסיפה את הצורה s לאוסף – מבלי לורדא תחילה שהצורה איננה באוסף!

השלם את השיטות הבאות (ע"י שימוש בשיטות הנתונות):

3.1 כתוב שיטה member הבודקת האם צורה מסוימת שייכת לאוסף (הבדיקה צריכה להיעשות במובן של שוויון לוגי בין צורות). אם הצורה קיימת (או שערכה null) השיטה מחזירה אמת, אחרת - שקר.

3.2 כתוב שיטה ציבורית adjoin המקבלת צורה ומוסיפה אותה לאוסף רק אם איננה null ואינה נמצאת כבר באוסף.

3.3 כתוב שיטה ציבורית equals המקבלת אוסף של צורות (ShapeContainer) ובודקת האם הוא שווה לוגית לעצם המפתח. (שני אוספים של צורות שווים אם הם מכילים בדיוק אותם איברים - ללא תלות בסדר הופעתם. אם האוספים שווים מחזירה אמת, אחרת - שקר.

3.4 כתוב שיטה ציבורית intersection המקבלת אוסף נוסף של צורות (ShapeContainer) sc ומחזירה אוסף חדש שהוא החיתוך של עצם המפתח ו-sc (אם sc==null מחזיר אוסף ריק).

```
public class ShapeContainer {
    // לא רלוונטי! - private data ...

    public ShapeContainer() { ... } // יוצרת אוסף חדש (ריק) של צורות
    public int size() { ... } // מחזירה את מספר הצורות באוסף
    private void addP (Shape sh) { ... } // שיטה פרטית! מוסיפה צורה לאוסף

    //////////////////////////////////////////////////// Q3 ////////////////////////////////////////////
    // ** בודקת האם הצורה קיימת באוסף **
    public boolean member(Shape sh) { // 3.1 השלם את השיטה בדף התשובות }

    // ** מוסיפה צורה לאוסף בתנאי שהיא חוקית **
    public void adjoin (Shape sh) { // 3.2 השלם את השיטה בדף התשובות }

    // ** בודקת האם שני אוספים זהים לוגית **
    public boolean equals(ShapeContainer sc) { // 3.3 השלם את השיטה בדף התשובות }

    // ** מחשבת חיתוך של שתי קבוצות **
    public ShapeContainer intersection(ShapeContainer sc) { // 3.4 השלם את השיטה
    בדף התשובות }
} // class Shape Container.
```

```
class Q4{
    static int count = 0;

    public static void main(String[] args){
        int [] array = {4,3,0,3,2};
        System.out.println(what(array,0)); // #1
        System.out.println(what(array,2)); // #2
    }

    public static boolean what(int[] array,int i){
        if (i >= 0 && i < array.length)
            return what(array,i,i+1);
        return false;
    }

    public static boolean what(int[] array,int i,int j){
        if (i == array.length-1)
            return true;
        if (j == array.length)
            return what(array,i+1);
        if(array[i] == array[j])
            return false;
        count = count + 1;
        return what(array,i,j+1);
    }

    public static void countWhat(int[] array,int i){
        count = 0;
        boolean b = what(array, i);
        System.out.println(count);
    }
}
```

- א. מה תדפיס התוכנית בשורה #1?
- ב. מה תדפיס התוכנית בשורה #2?
- ג. מה הוא הערך המקסימלי שהפונקציה `countWhat(array,i)` יכולה להדפיס כאשר מספר האיברים במערך `array` הוא `n`? נסחו את תשובתכם כפונקציה של `n`.
- ד. ענו במשפט אחד: איזו תכונה של המערך `array` בודקת הפונקציה `what(array,i)` (תשובות של יותר ממשפט אחד לא תזכינה בנקודות!).

### שאלה 5: (15 נקודות)

התייחסו לפונקציה הסטטית הבאה:

```
public static int guess(int[][] matrix,int i,int j) {
    if(matrix[i][j]==0)
        return 1;
    else
        return(guess(matrix,i+1,j) + guess(matrix,i,j+1));
}
```

הניחו כי matrix בגודל  $m \times n$  ( $m$  ו- $n$  חיוביים), וכי בקריאה הראשונה לפונקציה `guess(matrix,i,j)`,  $0 \leq j < n-1$  ו- $0 \leq i < m$ .

להלן מספר משפטים. יש לסמן לכל סעיף "אמת" או "שקר". כל תשובה נכונה תזכה ב-2 נקודות. כל תשובה לא נכונה תוריד 2 נקודות. סעיף שלא עניתם עליו לא יוריד ולא יוסיף ניקוד.

- א. עלולה להיות שגיאת קומפילציה.
- ב. עלולה להיות שגיאת הרצה.
- ג. לכל מערך דו-מימדי התכנית תגיע לסיום תקני.
- ד. תנאי מספיק אך לא הכרחי לכך שהקריאה ל-`guess(myMatrix,0,0)`, עבור קלט תקין, תבצע ללא שגיאת הרצה הוא שהשורה האחרונה והטור האחרון במטריצה יהיו אפסים.
- ה. ייתכן שהקריאה ל-`guess(myMatrix,0,0)` תחזיר את הערך 43.