

דף תשובות מבחן 2009 (בסיום נשמור רק את דף התשובות!)

מספר נבחן \_\_\_\_\_ מספר החדר \_\_\_\_\_

**שאלה 1 (20 נקודות)**

**סעיפים א'-ג**

```
public class numbers{

    public static void numbers(int n, int b){
        numbersRec(n, b, "");
    }

    public static void numbersRec(int n, int b, String acc) {
        if (n == 0) {
            System.out.println(acc);
        }
        else {
            for (int i = 0; i < b; i = i + 1)
                numbersRec(n-1, b, acc + i);
        }
    }

    public static void main(String[] args){
        numbers(2,3);
    }
}
```

## שאלה 2 (25 נק')

### סעיף א' (10 נקודות)

השיטה הבאה מתוספת למחלקה `BinaryNode`:

```
public boolean hasPathSum(int sum){
    boolean ans = false;

    if (data != null)
        sum = sum - ((Integer)data).intValue();
    if ((left == null) & (right == null) & (sum == 0))
        ans = true;
    if (left != null)
        ans = left.hasPathSum(sum);
    if ((ans == false) & (right != null))
        ans = right.hasPathSum(sum);

    return ans;
}
```

### סעיף ב' (15 נקודות)

ב1. השיטה הבאה מתוספת למחלקה `BinaryTree`:

```
public void printPaths() {
    root.printPaths("");
}
```

ב2. השיטה הבאה מתוספת למחלקה `BinaryNode`:

```
public void printPaths(String soFar){

    if (data != null)
        soFar = soFar + data + " ";
    if ((left == null) & (right == null))
        System.out.println(soFar);
    else {
        if (left != null)
            left.printPaths(soFar);
        if (right != null)
            right.printPaths(soFar);
    }
}
```

```
class StringTokenizer implements Iterator{
    private String str;
    private int currentIndex;

    public StringTokenizer(String str){
        this.str = str;
        currentIndex = 0;
    }

    public boolean hasNext(){
        return (currentIndex < str.length());
    }

    public Object next(){
        if (!hasNext())
            throw new NoSuchElementException();
        String ans = "";
        while ((currentIndex < str.length()) &&
            (str.charAt(currentIndex) != ' ')) {
            ans = ans + str.charAt(currentIndex);
            currentIndex = currentIndex + 1;
        }
        // move the currentIndex after the space
        currentIndex = currentIndex + 1;
        return ans;
    }
}
```

```
class StringTokenReverseIterator implements Iterator {
    private Stack st;

    public StringTokenReverseIterator (String str){
        st = new StackAsArray();
        StringTokenIterator it = new StringTokenIterator(str);
        while (it.hasNext())
            st.push(it.next());
    }

    public boolean hasNext(){
        return !(st.isEmpty());
    }

    public Object next(){
        if (!hasNext())
            throw new NoSuchElementException();
        return st.pop();
    }
}
```

```
public static int maxNumDigits(int[] arr){
    int max = arr[0];
    for (int i = 1; i < arr.length; i = i + 1) {
        if (arr[i] > max)
            max = arr[i];
    }
    String str = "" + max;
    return str.length();
}
```

סעיף ב (5 נקודות)

```
public static boolean greaterAtPosition(int x, int y, int i){
    int numPosition = (int) Math.pow(10, i - 1);
    int xDigit = (x / (numPosition)) % 10;
    int yDigit = (y / (numPosition)) % 10;
    return xDigit > yDigit;
}
```

סעיף ג (5 נקודות)

```
public static void sort(int [] arr, int i) {
    boolean isSorted = false;
    for (int bbl = 0; !isSorted && bbl < arr.length-1; bbl = bbl + 1) {
        isSorted = true;
        for (int index = 0; index < arr.length-1-bbl; index = index + 1) {
            if (greaterAtPosition(arr[index], arr[index + 1], i)) {
                tmp = arr[index];
                arr[index] = arr[index + 1];
                arr[index + 1] = tmp;
                isSorted = false;
            }
        }
    }
}
```

סעיף ד (5 נקודות)

```
public static void sortByPosition(int [] arr){
    int maxDigits = maxNumDigits(arr);
    for (int i = 1; i <= maxDigits; i = i + 1)
        sort(arr, i);
}
```

## שאלה 5 (15 נק')

בשאלה זו 4 סעיפים (א-ד). בסעיפים א'-ג' עליכם לבחור תשובה אחת נכונה מבין ארבע האפשרויות. עליכם להקיף בעיגול את התשובה שבחרתם.

סעיף א' (תשובה נכונה: 3 נקודות, תשובה לא נכונה: 1- נקודות, לא כתבת כלום: 0 נקודות).

הניחו כי `<a>` מוחלף בקוד הבא:

```
System.out.println(shraga.getInfo());
```

מה יודפס למסך?

1. Swallow speed: 15, coconut weight: 100.0
2. **Swallow speed: 15, coconut weight: 200.0**
3. Swallow speed: 20, coconut weight: 200.0
4. Swallow speed: 20, coconut weight: 100.0

סעיף ב' (תשובה נכונה: 3 נקודות, תשובה לא נכונה: 1- נקודות, לא כתבת כלום: 0 נקודות).

הניחו כי `<a>` מוחלף בקוד הבא:

```
System.out.println(shraga.toString());
```

מה יודפס למסך?

1. Hello. I'm a European swallow. How are you?
2. **Hello. I'm a European swallow. How do you do?**
3. Good morning. I'm a European swallow. How are you?
4. Good morning. I'm a European swallow. How do you do?

סעיף ג' (תשובה נכונה: 3 נקודות, תשובה לא נכונה: 1- נקודות, לא כתבת כלום: 0 נקודות).

הניחו כי `<a>` מוחלף בקוד הבא:

```
/*1*/ Swallow bigBird;
```

```
/*2*/ bigBird = new Swallow(20, coco);
```

```
/*3*/ System.out.println(bigBird.getType());
```

אזי

- (1) ישנה שגיאת קומפילציה בשורה `/*1*/`.
- (2) ישנה שגיאת קומפילציה בשורה `/*2*/`.
- (3) אין שגיאות קומפילציה בקטע קוד זה, אך השורה `/*3*/` השלישית תגרום לשגיאה בזמן ריצה.
- (4) הקוד יתקמפל וירוץ ללא שגיאות.

## סעיף ד' (4 נקודות)

הקיפו בעיגול "נכון" או "לא נכון" על כל אחת מהטענות הבאות (תשובה נכונה: 1 נקודות, תשובה לא נכונה: 0 נקודות, לא כתבת כלום: 0 נקודות).

\*הערה: בביטוי "ClassName1 obj = new ClassName2()", obj הוא מצביע לאובייקט הנוצר ע"י הבנאי ClassName2(). טיפוס המצביע הוא ClassName1, ואילו טיפוס האובייקט הקונקרטי הוא ClassName2.

(1) טיפוס של מצביע לאובייקט יכול להיות (בין השאר) מחלקה או ממשק (\*ראו הערה בראשית השאלה).

נכון \ לא נכון

(2) טיפוס של אובייקט קונקרטי יכול להיות (בין השאר) מחלקה או מחלקה אבסטרקטית (\*ראו הערה בראשית השאלה).

נכון \ לא נכון

(3) ערך הביטוי הבוליאני `x instanceof Type` נקבע בזמן ריצה לפי טיפוס האובייקט הקונקרטי עליו `x` מצביע (\*ראו הערה בראשית השאלה).

נכון \ לא נכון

(4) פעולה של פתיחה או סגירה של קובץ עלולה לזרוק `RuntimeException`.

נכון \ לא נכון

(5) חריגות מסוג `RuntimeException` יש חובה לתפוס (try ו-catch) או להעביר הלאה (throws).

נכון \ לא נכון

(6) בלוק של try יכול להיות מלווה ביותר מ-catch יחיד.

נכון \ לא נכון

# בהצלחה!