אוסף התרגילים הגדול

בשפת ג'אווה

שיתבור

אוסף התרגילים הגדול בשפת ג׳אווה

מהדורה ראשונה, 2019
יצירת קשר – shay.tavor@gmail.com

תוכן עניינים

הקדמה

הספר שאתם מחזיקים בידיכם הוא תוצר של שנים רבות של עבודה. במשך שנים רבות לימדתי סטודנטים לתכנת. בקורסים אקדמים, קורסים מקצועיים, קורסים פרטיים ועוד. במהלך התקופה הארוכה הזאת הרגשתי שאין מספיק מקורות לתרגול מסיבי ואינטנסיבי של החומר. אני מאמין גדול בתרגול, ואני חושב שכמו כל מיומנות טכנית, גם תכנות לומדים הכי טוב כשכותבים, והרבה. כמו שבשביל לתרגל פתרון משוואות ריבועיות במתמטיקה צריך לפתור כמה שיותר משוואות כאלה, ככה בשביל לתרגל תנאים, לולאות, מערכים או כל דבר אחר, צריך לכתוב כמה שיותר.

במהלך השנים צברתי תרגילים רבים שהשתמשתי בהם לאורך הזמן. התחלתי לאסוף אותם והם מרכיבים את האוסף שאתם מחזיקים עכשיו.

התרגילים מספקים תרגול של כל היבט תכנותי ביסודות השפה. הם יסודיים, מתחילים מרמה קלה וממשיכים לרמה גבוהה יותר. חלקם דורשים בעיקר טכניקה, חלקם דורשים מחשבה מעמיקה. אבל כולם מספקים את התרגול האינטנסיבי שכל מי שמתחיל לתכנת צריך.

הספר מחולק לפי נושאים, לכן אפשר לתרגל ישירות את הנושא הרצוי. כמובן שהנושאים תלויים אחד בשני, ואי אפשר לתרגל, למשל, לולאות לפני שיודעים תנאים, אבל מעבר לזה הקשר בין הפרקים רופף.

לסטודנטים ולתלמידים – אני ממליץ להתחיל לתרגל מהתרגילים הראשונים. לא לדלג על אף שלב ולפתור גם את אילו שנראים טריוויאלים. עצם הכתיבה של קוד תחייב אתכם להתמודד עם ההיבטים הטכניים, עם שגיאות קומפילציה ובאופן כללי, תחדד את המיומנות שלכם.

למרצים והמורים – האוסף הזה יכול לשמש בסיס טוב לתרגול בכיתה או לתרגילי בית. בהרבה תרגילים ניתן לעשות וריאציות שונות על אותו תרגיל, ולקבל גרסאות אחרות.

לסיום, אני רוצה להודות לכל אותם סטודנטים שעזרו לי במהלך הדרך בזה שהם ניסו ופתרו בעצמם את התרגילים בספר, תיקנו שגיאות והציעו הצעות לשיפור הניסוח והטקסט. בזכותם האוסף הזה מכיל תרגילים מעניינים, חשובים וברמה גבוהה.

אשמח לשמוע הערות ותיקונים בכתובת shay.tavor@gmail.com

אני מאחל לכם תרגול מועיל ומהנה,

שי תבור

קלט, פלט ואריתמטיקה

קובץ תרגילים זה מכיל תרגילים בנושא יסודות שפת ג'אווה – הגדרת משתנים, קלט, פלט, ואריתמטיקה. מומלץ לנסות ולפתור את כל התרגילים, לכתוב אותם בעצמכם בסביבת העבודה ולהריץ אותם. זאת הדרך הטובה ביותר ללמוד.

שאלה 1

בהנתן התכנית הבאה, סמנו עבור כל משתנה האם הגדרתו חוקית, אינה חוקית או חוקית אבל מסוכנת.

```
public class Tester {
    public static void main(String[] args) {
        int carNumber, 2lines, distanceToCenter;
        double int, Int, int2;
        long card#1, age;
        float age, void;
    }
}
```

שאלה 2

נתונה מערכת ממוחשבת להזמנת כרטיסים לסרט באינטרנט. בכל אחד מהסעיפים הבאים ישנו נתון שהמערכת שומרת. עבור כל אחד מהנתונים רשמו משתנה שיחזיק את הנתון – חשבו על טיפוס המשתנה ועל שמו.

א. מחיר כרטיס לסרט.
ב. מספר האנשים שקנו כרטיס לסרט.
ג. האם קונה הכרטיס הוא גבר או אישה.
ד. שם קונה הכרטיס.
ה. משך זמן הסרט בדקות.
ו. משך זמן הסרט במילי שניות (בשניה אחת יש 1000 מילי שניות).
ז. גיל מינימלי לצפיה בסרט.
ח. האם הסרט הוא קומדיה.

שאלה 3

התכנית הבאה עוברת קומפילציה, אולם כתובה בצורה מבלבלת ולא אלגנטית. כתבו את התכנית מחדש כך שתהיה יותר ברורה לקריאה – שמרו על רווחים, הזחות וכו'.

```
public class Tester { public static void main(String[] args) {
int x;long y; x=0;y=0;
System.out.
println("Welcome!")
;
x    ++;
y=x+5;x--;System.out
.println
("now x=" + x+" And y="+y
);
System.out.println("Goodbye")
;}}
```


shay.tavor@gmail.com

שאלה 4

בתכניות הבאות נפלו מספר שגיאות קומפילציה. תקנו אותן והריצו את התכניות. בכל סעיף אמור לצאת פלט מסויים. אם תצליחו להריץ את התכנית כהלכה, תקבלו את הפלט הצפוי.

א. הפלט הצפוי:

If you see this line, you've compiled it!

```
public class Tester {
    public static void main(String[] args) {
      system.out.println("Welcome...")
      System.out.prinln(I wonder if it compiles...);
      System.out.printLN("If you see this line, you've compiled it!);
    }
}
```

ב. הפלט הצפוי:

Values are: 1, 2, 3

```
public class Tester {
    public static void main(String[] args) {
        int x, y;
        y = x + 1;
        z = y + 1;
        System.out.println("Values are: " + x ", " + y + ", " + z);
    }
}
```

ג. הפלט הצפוי:

Result: 2

```
public class Tester {
    public static void main(String[] args) {
        double num = 62.0;
        int d1, d2;
        d1 = num / 10.0;
        d2 = num - num * 10;
        System.out.prinln("Result: " + d2);
    }
}
```

שאלה 5

א. כתבו מחלקה בשם Welcome. המחלקה תכיל תכנית (main) שתדפיס על המסך את המילה Welcome ובשורה מתחתיה את היום והתאריך. למשל, הפלט האפשרי:

Welcome!
Today is Monday, 28.3.2011.

ב. כתבו מחלקה בשם MyName שתכיל תכנית (main) שתדפיס על המסך את שמכם בתוך מלבן של כוכביות. למשל, הפלט האפשרי:

```
***************
*   David Cohen   *
***************
```

ג. כתבו מחלקה בשם Days שתכיל תכנית (main) שתדפיס על המסך את ימי השבוע, כל יום בשורה נפרדת. הפלט צריך להיות:

Days of the week:
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday

שאלה 6

כתבו מחלקה בשם Echo שתכיל תכנית. התוכנית תקלוט מהמשתמש מספר שלם, ותדפיס את המספר על המסך. לדוגמא, להלן אופן פעולת התכנית (השורות המודגשות הן הקלט מהמשתמש):

Please enter a number:
34
Your number is 34.

ועוד דוגמא:
Please enter a number:
398
Your number is 398.

שאלה 7

כתבו מחלקה בשם SquareAndCube שתכיל תכנית. התוכנית תקלוט מספר ממשי מהמשתמש, ותדפיס על המסך את המספר, את המספר בחזקת 2 ואת המספר בחזקת 3. לדוגמא, להלן אופן פעולת התכנית:

Please enter a number:
3
Number: 3.0, Square: 9.0, Cube: 27.0

ועוד דוגמא:

Please enter a number:
4.8
Number: 4.8, Square: 23.04, Cube: 110.592

שאלה 8

א. כתבו מחלקה בשם Calculations1 שתכיל תכנית. התכנית תקלוט שני מספרים ממשיים, ותדפיס על המסך את סכומם, הפרשם, מכפלתם ומנתם. בכל החישובים המספר הראשון יהיה הארגומנט השמאלי של פעולת החשבון. לדוגמא:

Please enter two numbers:
8 4
8.0 + 4.0 = 12.0
8.0 – 4.0 = 4.0
8.0 * 4.0 = 32.0
8.0 / 4.0 = 2.0

shay.tavor@gmail.com

ב. כתבו מחלקה בשם Calculations2 שתכיל תכנית. התוכנית תקלוט מספר ממשי ותדפיס על המסך את המספר ההופכי ואת המספר הנגדי למספר שנקלט. המספר ההופכי של x הוא 1/x. המספר הנגדי של x הוא x*1-. לדוגמא:

Please enter a number:
4
Inverse of 4.0 is 0.25, opposite of 4.0 is 0.25

ג. כתבו מחלקה בשם Calculations3 שתכיל תכנית. התוכנית תקלוט מספר ממשי, ותדפיס על המסך את ערכו המוחלט ואת השורש הריבועי שלו. כדי לחשב שורש ריבועי ניתן להשתמש בשיטה Math.sqrt. השיטה מקבלת מספר ממשי ומחזירה את השורש הריבועי שלו. למשל –

```
double x = Math.sqrt(9);        // x = 3.0
```

ניתן להשתמש בשיטה Math.abs כדי לחשב ערך מוחלט. למשל –

```
double y = Math.abs(-9.7);      // y = 9.7
```

שאלה 9

א. כתבו מחלקה בשם Rectangle שתכיל תכנית. התוכנית תקלוט מהמשתמש שני מספרים ממשיים המהווים רוחב וגובה של מלבן. התוכנית תדפיס על המסך את שטחו והיקפו של המלבן.
תזכורת – שטח המלבן = אורך * גובה
היקף המלבן = (אורך + גובה) * 2

ב. כתבו מחלקה בשם Circle שתכיל תכנית. התכנית תקלוט מהמשתמש מספר ממשי המהווה רדיוס של מעגל. התכנית תדפיס על המסך את שטחו והיקפו של המעגל.
תזכורת – שטח המעגל = 3.14 * רדיוס * רדיוס
היקף המעגל = 3.14 * 2 * רדיוס.
המספר 3.14 הוא הקבוע פאי, שמכיל אינסוף ספרות אחרי הנקודה. הגדירו את המספר כקבוע והשתמשו בו בתכנית.

שאלה 10

א. כתבו מחלקה בשם Numbers1 שתכיל תכנית. התכנית תקלוט מהמשתמש מספר חיובי שלם בן שלוש ספרות. התכנית תדפיס על המסך את ספרת האחדות, העשרות והמאות. לדוגמא:

Please enter a 3 digits number:
538
Ones: 8, Tens: 3, Hundreds: 5

ב. כתבו מחלקה בשם Numbers2 שתכיל תכנית. התכנית תקלוט מהמשתמש מספר חיובי שלם בן שלוש ספרות. התכנית תדפיס על המסך את סכום הספרות, ואת הפרש הספרות. לדוגמא:

Please enter a 3 digits number:
491
4 + 9 + 1 = 14
4 – 9 – 1 = -6

ג. כתבו מחלקה בשם Numbers3 שתכיל תכנית. התכנית תקלוט מהמשתמש מספר חיובי שלם בן ארבע ספרות. התכנית תהפוך את סדר הספרות במספר, ותדפיס על המסך את המספר ההפוך (כלומר, ספרת האחדות תהפוך להיות ספרת האלפים, ספרת העשרות תהיה ספרת המאות וכן הלאה). לדוגמא:

Please enter a 4 digits number:
8254
The reversed number is: 4528
שימו לב – יש ליצור מספר חדש שייצג את המספר ההפוך, ולאחסן אותו במשתנה.

ד. כתבו מחלקה בשם Numbers4 שתכיל תכנית. התכנית תקלוט מהמשתמש שלוש ספרות, תבנה מהן מספר תלת ספרתי, ותדפיס אותו על המסך. למשל:
Enter first digit:
4
Enter second digit:
9
Enter third digit:
2
The number is 492

ה. כתבו מחלקה בשם Numbers5 שתכיל תכנית. התכנית תקלוט מהמשתמש שלוש ספרות ותבנה מהן מספר ממשי בצורה הבאה:
Enter first digit:
4
Enter second digit:
9
Enter third digit:
2
The number is 0.492

שאלה 11

כתבו מחלקה בשם VAT שתכיל תכנית. התכנית תקלוט מהמשתמש מחיר, ותדפיס על המסך את המחיר לאחר מע"מ, ואת המע"מ ששולם. הניחו שאחוז המע"מ הוא 16.5%. למשל:
Please enter a price:
100
The price after VAT is 116.5, VAT is 16.5

שאלה 12

כתבו מחלקה בשם DollarsToShekels שתכיל תכנית. התכנית תקלוט מהמשתמש סכום בדולרים, ותדפיס על המסך את הסכום בשקלים. הניחו שדולר אחד שווה 3.5 שקלים. למשל:
Please enter a price:
240
240$ are 840.0 NIS.

שאלה 13

נגדיר **הזזה של מספר ימינה** בצורה הבאה – בהינתן מספר, הזזה של המספר ימינה תשנה את מיקומה של כל ספרה מיקום אחד ימינה. הספרה הימנית ביותר תהפוך לשמאלית ביותר. למשל, הזזה של המספר 578 תיתן את המספר 857. הזזה של המספר 934 תיתן את המספר 493.
כתבו מחלקה בשם Shift שתכיל תכנית. התכנית תקלוט מהמשתמש מספר בן שלוש ספרות ותדפיס על המסך את המספר המוזז ימינה. למשל:

Please enter a 3 digits number:
613
The shifted number is: 361

שאלה 14

בקורס מסויים מגישים הסטודנטים שלושה תרגילי בית, וכן נבחנים במבחן הסופי. משקל כל תרגיל בית הוא 5% מהציון הסופי, ומשקל המבחן הוא 85% מהציון הסופי. חישוב הציון הסופי נעשה לפי הנוסחה הבאה – נניח שסטודנט קיבל 90 בתרגיל 1, 80 בתרגיל 2 ו-100 בתרגיל 3, וכן 80 במבחן. ציונו הסופי יהיה:

90*0.05 + 80*0.05 + 100*0.05 + 80*0.85 = 81.5

כתבו מחלקה בשם FinalGrade שתכיל תכנית. התכנית תקלוט מהמשתמש ציוני שלושה תרגילים וכן ציון בחינה, ותדפיס על המסך את הציון הסופי. למשל:

Enter EX1 grade:
90
Enter EX2 grade:
80
Enter EX3 grade:
100
Enter final exam grade:
80
Your final grade is: 81.5

שאלה 15

כתבו מחלקה בשם Money שתכיל תכנית. התכנית תקלוט מהמשתמש מספר ממשי המייצג סכום כסף. התכנית תפרט את הסכום לפי מספר השטרות של 200, 100, 50 ו-20 שמרכיבם את הסכום, וכן מספר המטבעות של 10, 5 ו-1, וכן ממספר המטבעות של חצי שקל ועשר אגורות שמרכיבים את הסכום. למשל, הסכום 359.70 מורכב מהפירוט הבא:
שטר אחד של 200, שטר אחד של 100, שטר אחד של 50, מטבע אחד של 5, 4 מטבעות של 1, מטבע אחד של חצי שקל ושני מטבעות של 10 אגורות.
הדרכה – שימו לב שפעולת השארית (מודולו) עובדת נכון רק על מספרים שלמים. בנוסף, שימו לב שיש להפריד את חישוב המספר השלם מחישוב האגורות, ולכן עליכם למצוא דרך להפוך את מספר האגורות למספר שלם. למשל, בסכום 359.70 יש לחשב תחילה את המרכיבים של 359, ואז את המרכיבים של 70 אגורות.

shay.tavor@gmail.com

משפטי תנאי

קובץ זה מכיל תרגילים בנושא משפטי תנאי בשפת ג'אווה. התרגילים מסודרים מהקל לקשה, כאשר בתחילת הקובץ ישנם תרגילים קצרים להבנת המבנה התחבירי של הפקודות, ובהמשך ישנם תרגילי תכנות להרצה. מומלץ לפתור את כל התרגילים, לכתוב אותם בעצמכם ולהריץ אותם בסביבת העבודה. זאת הדרך הטובה ביותר להבין ולהפנים את החומר.

שאלה 1

נתונות הגדרות המשתנים הבאות:

```
int age;
double productPrice;
int day;
```

המשתנה age מייצג גיל של בן אדם, המשתנה productPrice מייצג מחיר של מוצר והמשתנה day מכיל יום בשבוע (1 הוא יום ראשון, 7 הוא יום שבת).

עבור כל אחד מהסעיפים הבאים כתבו משפט תנאי מתאים שיבצע את הבדיקה הרצויה:

א. האם הגיל הוא מעל 18.
ב. האם מחיר המוצר הוא פחות מ-100.
ג. האם היום יום שני.
ד. האם מחיר המוצר חוקי (מחיר חוקי הוא מחיר חיובי).
ה. האם היום יום זוגי (שני, רביעי או שישי).
ו. האם האדם אינו קטין (מעל גיל 18) וגם מתחת לגיל הפנסיה (65).

שאלה 2

נתונה התכנית הבאה:

```
import java.util.Scanner;
public class Tester {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int x;
        x = scan.nextInt();
        // ...
    }
}
```

בכל אחד מהסעיפים נתונות שורות קוד שיכנסו במקום ההערה בתכנית שלעיל. עבור כל אחד מהסעיפים תנו דוגמא עבור ערך של x שעבורו התנאי יתקיים, וערך של x שעבורו התנאי לא יתקיים.

א.

```
if(x > 0)
    System.out.println("OK");
```

ב.

```
if(x == 1 || x == -1)
    System.out.println("OK");
```

ג.

```
if(x >= 3 && x <= 20 && x != 10)
    System.out.println("OK");
```


shay.tavor@gmail.com

ד.

```
if(x % 2 == 0)
    System.out.println("OK");
```

ה.

```
if((x > 10 && x < 20) || (x < -10 && x > -20))
    System.out.println("OK");
```

שאלה 3

מה מבצעת התכנית הבאה? נסחו מה יהיה פלט התכנית.

```
public class Tester {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int x, y, res = 0;
        x = scan.nextInt();
        y = scan.nextInt();

        if(x > y)
            res = x - y;
        else
            res = y - x;
        System.out.println("Result is: " + res);
    }
}
```

שאלה 4

נתונה התכנית הבאה. התכנית עוברת קומפילציה ורצה נכון, אולם היא מכילה שכפולי קוד. כתבו את התכנית מחדש כך שתבצע את אותן פעולות כמו התכנית המקורית, אבל לא תכיל שכפולי קוד.

```
public class Tester {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int x, y, res = 0;
        x = scan.nextInt();
        y = scan.nextInt();

        if(x > y && y != 0) {
            System.out.println("Calculating....");
            res = (x - y) / y;
            System.out.println("The result is: " + res);
        }
        else if(x < y && y != 0){
            System.out.println("Calculating....");
            res = (y - x) / y;
            System.out.println("The result is: " + res);
        }
        else
            System.out.println("Invalid input");
    }
}
```

שאלה 5

כתבו מחלקה בשם Absolute שתכיל main. התכנית תקבל כקלט מהמשתמש מספר, ותדפיס על המסך את ערכו המוחלט של המספר. בשאלה זו אין צורך להשתמש ב-else.

תזכורת – ערך מוחלט של מספר הוא המספר בלי סימן. למשל, הערך המוחלט של 8 הוא 8, והערך המוחלט של 4- הוא 4. להלן דוגמא להרצת התכנית:

Please enter a number: **3**
Absolute value is 3
ועוד דוגמא:
Please enter a number: **-5**
Absolute value is: 5

שאלה 6

כתבו תכנית בשם Salary. התכנית תקלוט מהמשתמש גיל, ומשכורת חודשית. אם גילו של המשתמש מעל 18, התכנית תנקה משכרו 5% למיסי בריאות. התכנית תדפיס את המשכורת הסופית.
להלן דוגמא להרצת התכנית:

Please enter your age: **20**
Please enter your salary: **3000**
Your salary is: 2850

ועוד דוגמא:

Please enter your age: **16**
Please enter your salary: **1500**
Your salary is: 1500

שאלה 7

א. כתבו מחלקה בשם BigSmall1 שתכיל main. התכנית תקלוט מהמשתמש שני מספרים ותדפיס את המספר הגדול ואת המספר הקטן. להלן דוגמא להרצת התכנית:

Please enter first number: **4**
Please enter second number: **8**
The bigger number is: 8

ב. כתבו מחלקה בשם BigSmall2 שתכיל main. התכנית תקלוט מהמשתמש שלושה מספרים ותדפיס אותם בסדר עולה, כלומר מהקטן לגדול. להלן דוגמא להרצת התכנית:

Please enter first number: **10**
Please enter second number: **8**
Please enter third number: **2**
Sorted numbers are: 2, 8, 10

שאלה 8

כתבו מחלקה בשם Rectangle, שתכיל main. התכנית תקלוט מהמשתמש אורך ורוחב של מלבן, ותדפיס על המסך את שטחו והיקפו של המלבן. אם המלבן הוא ריבוע, התכנית תדפיס על המסך שזהו ריבוע (ראו דוגמא). אם אחד הפרמטרים או שניהם אינו חוקי התכנית תוציא הודעת שגיאה.
להלן דוגמא להרצת התכנית:

Please enter rectangle's width: **5**
Please enter Rectangle's length: **4**
Rectangle's area: 20, Rectangle's perimeter: 18

ועוד דוגמא:

Please enter rectangle's width: **5**
Please enter Rectangle's length: **5**
Square's area: 25, Square's perimeter: 20

עוד דוגמא:

Please enter rectangle's width: **5**
Please enter Rectangle's length:- **4**
Invalid input.

שאלה 9

מספר בן 3 ספרות יקרא **מספר אקסטרים** אם הספרה האמצעית גדולה משתי הספרות האחרות, או קטנה משתי הספרות האחרות. למשל, המספרים הבאים הם מספרי אקסטרים: 132, 683, 739, 929.
המספרים הבאים **אינם** מספרי אקסטרים: 111, 123, 942.
כתבו מחלקה בשם ExtremeNumber שתכיל main. התכנית תקלוט מהמשתמש מספר בן שלוש ספרות. אם המספר אינו חוקי (כלומר אינו בן שלוש ספרות) התכנית תדפיס הודעת שגיאה. אם המספר חוקי, התכנית תבדוק אם הוא מספר אקסטרים ותדפיס זאת על המסך. להלן דוגמא להרצת התכנית:

Please enter 3 digits number: **547**
547 is an extreme number.

ועוד דוגמא:

Please enter 3 digits number: **543**
543 is not an extreme number.

ועוד דוגמא:

Please enter 3 digits number: **5467**
Invalid input.

שאלה 10

כתבו מחלקה בשם Sequence שתכיל main. התכנית תקלוט מהמשתמש מספר בן שלוש ספרות, ותבדוק האם המספר מהווה סדרה עוקבת של מספרים. כלומר שהספרה האמצעית עוקבת לספרה השמאלית, והספרה הימנית עוקבת לספרה האמצעית. למשל, המספרים הבאים מהווים סדרה עוקבת: 123, 567, 345. המספרים הבאים **אינם** מהווים סדרה עוקבת: 135, 646, 222.
אם הקלט אינו מספר בן שלוש ספרות, התכנית תדפיס הודעת שגיאה למשתמש.

שאלה 11

כתבו מחלקה בשם Hotel שתכיל main. התכנית תדמה מערכת הזמנת חדרים במלון. במלון ישנם שלושה סוגי חדרים – חדר רגיל, חדר כפול וסוויטה. מחיר לילה בחדר רגיל הוא 250 ש"ח, בחדר כפול המחיר הוא 400 שח"ח ובסוויטה המחיר הוא 600 ש"ח. בנוסף, ניתן לבקש גם ארוחת בוקר, שמחירה 50 שקל, בכל סוג חדר.
התכנית תקרין על המסך תפריט המבקש מהמשתמש לבחור את סוג החדר, ולאחר מכן את מספר הלילות, ולאחר מכן האם הוא רוצה ארוחת בוקר. בסיום התהליך התכנית תדפיס על המסך את פרטי ההזמנה – סוג החדר, מספר הלילות, אם או בלי ארוחת בוקר והמחיר הכולל.

בכל אחד משלבי הקלט, אם הקלט אינו חוקי (למשל, מספר לילות שלילי, או מספר חדר לא חוקי) התכנית תדפיס הודעת שגיאה מתאימה ותסיים.
להלן דוגמא להרצת התכנית:

Welcome to our hotel. In which room do you want to stay?
1. Regular (250 nis per night)
2. Double (400 nis per night)
3. Suite (600 nis per night)
1
How many nights do you want to stay?
3
Do you want breakfast included (50 nis per night)? (press 1 for yes, 2 for no)
1
Your reservation for regular roo, nights with breakfast will cost total of 900 nis.

ועוד דוגמא:

Welcome to our hotel. In which room do you want to stay?
1. Regular (250 nis per night)
2. Double (400 nis per night)
3. Suite (600 nis per night)
4
Invalid room type.

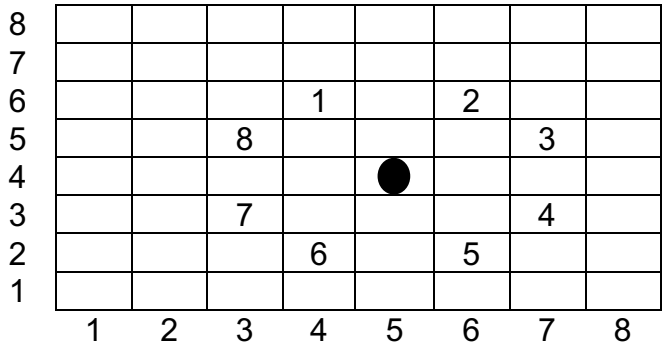
ועוד דוגמא:

Welcome to our hotel. In which room do you want to stay?
1. Regular (250 nis per night)
2. Double (400 nis per night)
3. Suite (600 nis per night)
1
How many nights do you want to stay?
-4
Invalid nights number.

שאלה 12

במשחק השחמט ישנו לוח של 8X8 משבצות. כלי המשחק יכולים לנוע בכיוונים שונים, בהתאם לסוג הכלי. הפרש (knight) במשחק נע בצורה הבאה – שתי משבצות בכיוון אחד, ומשבצת אחת בכיוון המאונך. למשל, שתי משבצות קדימה ומשבצת אחת שמאלה, או שתי משבצות שמאלה ומשבצת אחת למטה.
כתבו מחלקה בשם KnightMoves שתכיל main. התכנית תקלוט מהמשתמש מספר שורה ומספר עמודה של משבצת בה נמצא הפרש. התכנית תדפיס על המסך את קואורדינטות כל המשבצות אליהן יכול הפרש להגיע מהמשבצת ההתחלתית. אם פרטי הקלט אינם חוקיים (מספר שקטן מאחד או גדול משמונה) התכנית תדפיס הודעת שגיאה ותסיים.
למשל, האיור הבא מראה את לוח השחמט ובו פרש בנקודה (4, 5) (שורה 4 עמודה 5) ואת המשבצות אליהן יכול הפרש להגיע מנקודה זו (מסומנות במספרים 1 עד 8).

shay.tavor@gmail.com



להלן דוגמא להרצת התכנית:

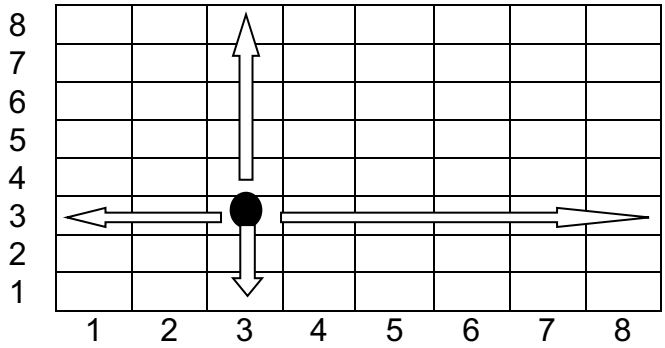
Please enter the knight's row: **4**
Please enter the knight's column: **5**
Possible knight's moves are:
(6, 4)
(6, 6)
(5, 7)
(3, 7)
(2, 6)
(2, 4)
(3, 3)
(5, 3)

ועוד דוגמא:

Please enter the knight's row: **9**
Please enter the knight's column: **5**
Invalid position.

שאלה 13

בלוח השחמט, הצריח יכול לנוע רק קדימה, אחורה, ימינה או שמאלה, בקו ישר. האיור הבא מראה צריח שממוקם בנקודה (3, 3). החיצים מראים את כל המשבצות אליהם יכול הצריח להגיע מנקודה זו.



כתבו מחלקה בשם RookAndKnight שתכיל main. התכנית תקלוט מהמשתמש מיקום של פרש (שורה ועמודה) ומיקום של צריח (שורה ועמודה) על לוח השחמט. התכנית תדפיס האם הצריח מאיים על הפרש, או שהפרש מאיים על הצריח או שאין איום.

shay.tavor@gmail.com

למשל, באיור שלעיל, אם היינו ממקמים את הפרש על אחת מהמשבצות שנמצאות על החיצים, אזי הצריח היה מאיים על הפרש. אם היינו ממקמים את הפרש בנקודה (1,4) למשל, הפרש היה מאיים על הצריח, ואם היינו ממקמים את הפרש בנקודה (7,5) אז לא היה איום.

שאלה 14

כתבו מחלקה בשם TimeDiff שתכיל main. התכנית תקלוט מהמשתמש פרטי שעות בפורמט הבא – שעות ודקות – עבור שני זמנים. התכנית תדפיס על המסך איזה מהזמנים קודם ואיזה מאוחר יותר.

להלן דוגמא להרצת התכנית:
Please enter first time's hour: **20**
Please enter first time's minutes: **14**
Please enter second time's hour: **13**
Please enter second time's minutes: **0**
13:00 is earlier than 20:14

שימו לב לפורמט הפלט – השעות והדקות תמיד צריכות להיות בפורמט של שתי ספרות. כך שהשעה שמונה ושתי דקות בבוקר תודפס כך 08:02.

שאלה 15

נתונה קטע הקוד הבא. החליפו את משפטי התנאי במשפט switch כך שקטע הקוד יהיה בעל אותה משמעות.

```
int num, x;
if(num == 1)
    System.out.println("One");
else if(num == 2 || num == 3)
{
    x = num * 2;
    System.out.println("x is: " + x);
}
else if(num == 3 || num == 4 || num == 5)
{
    x = num * 10;
    System.out.println("x is: " + x);
}
else
    System.out.println("Other...");
```

שאלה 16

כתבו מחלקה בשם Months שתכיל main. התכנית תקלוט מהמשתמש מספר חודש (בין 1 ל-12) ותדפיס על המסך את שם החודש הלועזי (ינואר, פברואר וכו'). אם מספר החודש שנקלט אינו חוקי, התכנית תדפיס הודעת שגיאה. ממשו את התכנית באמצעות switch.

שאלה 17

כתבו מחלקה בשם Calculator שתכיל main. התכנית תקלוט מהמשתמש שני מספרים חיוביים. אם הקלט אינו חוקי (מספר לא חיובי) התכנית תדפיס הודעת שגיאה ותסיים. אם הקלט חוקי התכנית תקרין על המסך תפריט אפשרויות שמאפשר למשתמש לבחור אם לחבר בין שני המספרים, להפחית בניהם או להכפיל בניהם. עבור כל אופציה התכנית תדפיס את תוצאת הפעולה. אם המשתמש הזין אפשרות לא קיימת, התכנית תדפיס הודעת שגיאה ותסיים.
ממשו את התפריט באמצעות switch.

מחלקות ואובייטים חלק א'

קובץ תרגילים זה מכיל שאלות בנושא מחלקות, אובייקטים ושיטות בשפת ג'אווה. חלק מהשאלות הן תאורטיות ונועדו לעזור לקורא לבחון את הידע שלו בנושא. רוב השאלות הן שאלות כתיבת קוד, ומסודרות מהקל לקשה. כדאי מאוד להריץ את התשובות (גם את הקלות שבהן) ולוודא שאתם מבינים איך הקוד עובד.

שאלה 1

ענו על השאלות הבאות. השתדלו לנמק בצורה מלאה ולא מעורפלת. זה אמנם לא קל אך משתלם בהמשך!

1. מה זאת מחלקה? מה זה אובייקט? מה ההבדל בניהם?
2. כמה אובייקטים ניתן ליצור ממחלקה?
3. מה משותף לאובייקטים שנוצרו מאותה מחלקה?
4. מה זאת תכונה של מחלקה? איך מגדירים תכונה?
5. מה זאת שיטה? מה זאת חתימה של שיטה?
6. מהו ערך חוזר משיטה? איך מבצעים החזרת ערך משיטה?
7. מהם הפרמטרים הפורמלים של השיטה?
8. מה זה מאפיין גישה (Access Modifier)? מה ההבדל בין public ל-private?
9. מה זה בנאי (Constructor)? מתי מופעל הבנאי של האובייקט? לשם מה נרצה לכתוב בנאי במחלקה?
10. מה זה בנאי דיפולטיבי (בנאי ריק – Empty Constructor)?
11. מה זה ריבוי בנאים? לשם מה נרצה להשתמש בריבוי בנאים במחלקה?
12. איך הקומפיילר יודע להחליט באיזה בנאי להשתמש בזמן יצירת האובייקט?
13. האם בנאי יכול להשתמש בתכונות הפרטיות של המחלקה בה הוא מוגדר?
14. מהן שיטות get ו-set? לשם מה נצטרך אותן?
15. האם ניתן להגדיר שיטה פרטית (private)? לשם מה להגדיר שיטה כזאת?
16. מהי השיטה toString? מה מטרתה ואיך משתמשים בה?

שאלה 2

א. נתונה המחלקה Car המייצגת מכונית (בעמוד הבא). למכונית יש מספר, שם בעלי המכונית, ומספר הקילומטרים שנסעה המכונית. סמנו במחלקה את המרכיבים הבאים:

1. שם המחלקה.
2. התכונות של המחלקה.
3. הבנאי הריק של המחלקה.
4. הבנאים של המחלקה.
5. השיטות של המחלקה.
6. שיטות gettrs/setters של המחלקה.

shay.tavor@gmail.com

```java
public class Car
{
    private int carNum;
    private String owner;
    private int km;

    public Car()
    {
        carNum = 111;
        owner = "No Owner Yet";
        km = 0;
    }

    public Car(int num, String carOwner)
    {
        carNum = num;
        owner = carOwner;
        km = 0;
    }

    public int getCarNum()
    {
        return carNum;
    }

    public String getOwner()
    {
        return owner;
    }

    public void setOwner(String carOwner)
    {
        owner = carOwner;
    }

    public int getKm()
    {
        return km;
    }

    public void setKm(int carKm)
    {
        if(carKm > km)
            km = carKm;
    }

    public void drive(int distance)
    {
        km += distance;
    }

    public String toString()
    {
        return "Car num: " + carNum + ", km: " + km +
                                    ", belongs to " + owner;
    }
}
```

ב. התייחסו למחלקה Car וענו על השאלות הבאות:

shay.tavor@gmail.com

1. איזו שיטה אחראית לקביעת ערך מספר הרכב?
2. האם אפשר לשנות את ערך מספר הרכב אחרי שהאובייקט נוצר?
3. האם אפשר לשנות את ערך בעלי הרכב אחרי שהאובייקט נוצר?
4. האם יתכן מצב שבו יווצרו שני אובייקטים מהמחלקה שלשניהם אותו מספר רכב?
5. האם ניתן לגרום לכך שערך התכונה km יהיה שלילי? אם לא הסבירו למה לא, ואם כן הדגימו כיצד ניתן לבצע זאת.
6. התייחסו לקוד הבא הרשום ב-main :

```
Car c1 = new Car();
Car c2 = new Car(222, "David");
System.out.println(c1);
System.out.println(c2);
```

מה יהיה פלט הקוד?

7. כתבו שורת קוד שיוצרת אובייקט מסוג Car בשם myCar שמספרו 1234 ושם בעלי הרכב הוא "Yossi". שנו את מספר הקילומטרים ל-100 והדפיסו את פרטי האובייקט.
8. בשורה הבאה הכתובה ב-main ישנה שגיאת קומפילציה. הסבירו מהי.

```
Car c3 = new Car("Dafna", 345);
```

שאלה 3

נתון קוד ה-main הבא שעושה שימוש במחלקה Unknown. נתון שהקוד עובר קומפילציה ורץ.

```
public class Tester
{
    public static void main(String[] args)
    {
        Unknown a1 = new Unknown(1, 4);
        Unknown a2 = new Unknown();
        Unknown a3 - new Unknown(3, 8.9);

        int x = a1.get1();
        int y = a2.get1();
        if(a3.f(x, y))
            System.out.println("OK");
        else
            System.out.println("Wrong");
    }
}
```

ענו על השאלות הבאות:

1. כמה בנאים יש למחלקה Unknown? מה חתימות הבנאים?
2. מהי חתימת השיטה get1 במחלקה Unknown?
3. מהי חתימת השיטה f במחלקה Unknown?

שאלה 4

נתונה המחלקה הבאה:

```
public class MyClass {
    public int f() {
        return 3;
    }
    public int g(int x) {
        return x + 2;
    }
    public int h(int x) {
        return x + 5;
    }
}
```

נתון ה-main הבא העושה שימוש במחלקה MyClass. עבור כל שורה ב-main הסבירו מה עושה השורה, וציינו מה הפלט אם יש:

```
public class Tester {
    public static void main(String[] args) {
        MyClass b = new MyClass();
        int y = b.f();
        System.out.println(y);
        System.out.println(b.f());
        int x = b.g(y);
        System.out.println(x);
        System.out.println(b.g(b.f()));
        System.out.pritln(b.h(b.g(b.f())));
        System.out.println(b.h(b.g(b.f() - 1) + 7));
    }
}
```

שאלה 5

נתונה המחלקה Address המייצגת כתובת מגורים:

```
public class Address
{
    private String city, street;
    private int number, zip;
}
```

כתובת מיוצגת ע"י שם העיר ושם הרחוב (מסוג String) ומספר הבית והמיקוד (מסוג int).

1. הוסיפו למחלקה בנאי שחתימתו:

   public Address(String c, String s, int n, int z)

   הבנאי יקבל כפרמטרים את נתוני הכתובת ויציב אותם לתכונות.
2. הוסיפו למחלקה בנאי שחתימתו:

   public Address(String c, String s, int n)

   הבנאי יקבל כפרמטרים את שם העיר, הרחוב ומספר הבית ויציב אותם לתכונות. בתכונת המיקוד הבנאי יציב 0.
3. הוסיפו שיטות get/set עבור כל אחת מהתכונות.
4. הוסיפו שיטת toString שתחזיר את פרטי הכתובת בפורמט הבא:

   Address: [street] st., [number], [city], [zip]

   למשל, אם הכתובת היא רחוב הרצל 34 ירושלים, מיקוד 778345, הפעלת השיטה תחזיר את המחרוזת:

   Address: Herzl st., 34, Jerusalem, 778345

   שימו לב, אם אין מיקוד בכתובת (כלומר המיקוד שווה לאפס) השיטה לא תחזיר את המיקוד, כלומר אחרי שם העיר לא יופיע פסיק ומיקוד.

5. כתבו מחלקה בשם AddressTester שתכיל main. צרו ב-main שני אובייקטים מסוג Address. הראשון יהיה בכתובת רחוב וייצמן 59 ת"א, מיקוד 23432. השני יהיה בכתובת רחוב בן גוריון 12 ר"ג, ללא מיקוד. הדפיסו את שני האובייקטים. הפעילו את שיטת setZip עבור הכתובת השניה והציבו מיקוד 45456. הדפיסו שוב את הכתובות.

שאלה 6

כתבו מחלקה בשם Book המייצגת ספר. המחלקה תכיל את התכונות הבאות:
name – שם הספר, מסוג String.
author – שם הסופר, מסוג String.
numberOfPages – מספר העמודים בספר, מסוג int.
הוסיפו את המרכיבים הבאים למחלקה:

1. בנאי שמקבל כפרמטרים את שם הספר, שם הסופר ומספר העמודים ומציב אותם לתכונות. אם מספר העמודים הוא שלילי, הבנאי יציב בתכונה את הערך 1.
2. שיטות get עבור התכונות.
3. שיטת toString שתחזיר מחרוזת המייצגת את פרטי הספר בפורמט הבא. למשל, אם שם הספר הוא Oliver Twist שנכתב ע"י Charles Dikens והוא מכיל 200 עמודים, השיטה תחזיר את המחרוזת הבאה:
   Oliver Twist by Charles Dikens, 200 pages.

תוכלו לבדוק את המחלקה שכתבתם באמצעות ה-main הבא:

```
public class Tester
{
    public static void main(String[] args)
    {
        Book b1 = new Book("Oliver Twist", "Charles Dikens", 200);

        System.out.println(b1);
        System.out.println("Book name: " + b1.getName());
        System.out.println("Book author: " + b1.getAuthor());
        System.out.println("Book pages: " + b1.getnumberOfPages());
    }
}
```

שאלה 7

כתבו מחלקה בשם Course שתייצג קורס באוניברסיטה. לקורס יהיו התכונות הבאות:
courseName – שם הקורס, מסוג String.
courseNum – מספר הקורס, מסוג int.
points – מספר נקודות הזכות שהקורס מעניק, מסוג double.
הוסיפו למחלקה בנאי שיקבל את שלושת הנתונים ויציב אותם לתכונות.
הוסיפו שיטות get/set עבור כל אחת מהתכונות.
הוסיפו את השיטה toString שתחזיר מחרוזת המייצגת את פרטי הקורס, מופרדים בפסיקים.
באוניברסיטה הקורסים מסווגים לפי שלושה סוגים – קורס פתיחה, קורס רגיל, קורס מתקדם. הסיווג נעשה עפ"י מספר נקודות הזכות שמעניק הקורס – קורס פתיחה מעניק 2.5 או פחות נקודות. קורס רגיל מעניק בין 3 ל-5 נקודות זכות וקורס מתקדם מעניק 6 נקודות.
הוסיפו למחלקה את השיטות הבאות:
public boolean isIntroCourse() – השיטה תחזיר true אם הקורס הוא קורס פתיחה, ו-false אם לא.
public boolean isRegularCourse() – השיטה תחזיר true אם הקורס הוא קורס רגיל, ו-false אם לא.
public boolean isAdvancedCourse() – השיטה תחזיר true אם הקורס הוא קורס מתקדם, ו-false אם לא.

שאלה 8

כתבו מחלקה בשם Word שתייצג מילה בעברית ובאנגלית. המחלקה תכיל את התכונות:
hebrew – תכיל את המילה בעברית, מסוג String.
english – תכיל את המילה באנגלית, מסוג String.
הוסיפו למחלקה בנאי שחתימתו:
public Word(String heb, String eng) – הבנאי יקבל מילה בעברית ובאנגלית ויציב אותן לתכונות.
הוסיפו שיטות get/set עבור התכונות.
הוסיפו את השיטה toString שתחזיר מחרוזת המייצגת את המילה באנגלית, נקודותיים, ואת המילה בעברית. למשל, עבור המילה "כלב" ו-"dog" השיטה תחזיר את המחרוזת:: כלב : dog

שאלה 9

כתבו מחלקה בשם Circle שתייצג מעגל. מעגל ייוצג ע"י התכונות:
radius – רדיוס המעגל, מסוג double.
centerX – קואורדינטת ה-x של מרכז המעגל (מסוג double).
centerY – קואורדינטת ה-y של מרכז המעגל (מסוג double).
כתבו בנאי שמקבל את שלושת הפרמטרים של התכונות ומציב אותם לתכונות. אם ערך הרדיוס קטן או שווה לאפס, הבנאי יציב 1 ברדיוס.
כתבו שיטת toString שתחזיר את פרטי המעגל בפורמט הבא: אם המעגל נמצא למשל בנקודה (1.0, 2.0) והרדיוס שלו הוא 5.6 השיטה תחזיר את המחרוזת:
Circle at (1.0, 2.0) radius 5.6
הוסיפו שיטות get בלבד עבור התכונות.
הוסיפו את השיטות הבאות למחלקה:
public double area() – השיטה תחזיר את שטח המעגל. להזכירכם, שטח המעגל הוא pi*r*r כאשר pi הוא קבוע מספרי. ניתן להשתמש בקבוע Math.PI.
public double perimeter() – השיטה תחזיר את היקף המעגל. היקף המעגל הוא 2*pi*r.
public double diameter() – השיטה תחזיר את קוטר המעגל. קוטר המעגל הוא פעמיים הרדיוס.
public void moveTo(int x, int y) – השיטה תקבל קואורדינטות של נקודה, ותזיז את המעגל לנקודה זו.
public void resize(double factor) – השיטה תשנה את רדיוס המעגל פי factor. אם factor קטן או שווה לאפס, השיטה לא תשנה את רדיוס המעגל.
public boolean isIn(int x, int y) – השיטה תקבל כפרמטרים קואורדינטות של נקודה, ותחזיר true אם הנקודה נמצאת בתוך שטח המעגל או על ההיקף שלו, ו-false אם לא.

שאלה 10

כתבו מחלקה בשם BankAccount המייצגת חשבון בנק. לחשבון יהיו התכונות הבאות:
num – מספר החשבון, מסוג int.
balance – היתרה בחשבון, מסוג double.
overdraftLimit – תקרת משיכת היתר, מסוג double.
הוסיפו למחלקה בנאי שחתימתו:
public BankAccount(int num) – הבנאי יקבל כפרמטר את מספר החשבון, יציב אותו לתכונה, וכן יציב 0 בשתי התכונות האחרות.
הוסיפו שיטות get עבור התכונות.
הוסיפו את השיטה toString שתחזיר מחרוזת המייצגת את פרטי החשבון. למשל, אם מספר החשבון הוא 111, יתרתו היא 3000 ותקרת משיכת היתר היא 1000- השיטה תחזיר את המחרוזת:
Num: 111, Balance: 3000.0, Limit: -1000.0

הוסיפו למחלקה את השיטות הבאות:
public boolean transaction(double amount) – השיטה תבצע פעולה על החשבון – משיכה או הפקדה של סכום כסף. אם הפרמטר amount הוא חיובי, מדובר בהפקדה. השיטה תוסיף אותו ליתרה ותחזיר true.
אם הפרמטר amount הוא שלילי מדובר במשיכה. אם כתוצאה מהמשיכה ערך היתרה יהיה קטן יותר מערך תקרת משיכת היתר, המשיכה לא תתבצע והשיטה תחזיר false. אחרת השיטה תוריד את הסכום מהיתרה ותחזיר true.
public boolean setOverdraftLimit(double limit) – השיטה תקבל תקרת משיכת יתר ותציב אותה בתכונה. אם הפרמטר limit הוא חיובי, או ששינוי התקרה יגרום ליתרה להיות קטנה יותר מהתקרה, השינוי לא יתבצע והשיטה תחזיר false. אחרת השיטה תשנה את תקרת משיכת היתר לערך limit ותחזיר true.

שאלה 11

כתבו מחלקה בשם Time המייצגת זמן. הזמן ייוצג ע"י התכונות הבאות:
hours – השעות, מסוג int.
minutes – הדקות, מסוג int.
is24 – תכונה מסוג boolean המסמנת האם השעה מיוצגת בפורמט של 12 שעות או של 24 שעות. אם השעה היא בפורמט של 24 שעות, ערך התכונה יהיה true, ואם השעה מיוצגת בפורמט של 12 שעות ערך התכונה יהיה false.
isAM – האם השעה היא בחצי הראשון של היום (am) או בחצי השני (pm). תקף עבור שעות בפורמט של 12 שעות. התכונה מסוג boolean.
הוסיפו למחלקה בנאי שחתימתו:
public Time(int h, int m) – הבנאי יקבל את ערכי השעה והדקה ויציב אותן בתכונות. כמו כן הבנאי יגדיר את הזמן בפורמט של 24 שעות. אם ערך השעה או הדקה אינו חוקי, הבנאי יציב בתוכן 0.
הוסיפו את השיטות get עבור התכונות.
הוסיפו את השיטה toString. השיטה תחזיר מחרוזת המייצגת את הזמן. למשל, אם השעה היא אחת חמישים וארבע בצהריים, השיטה תחזיר את המחרוזת: 13:54. שימו לב שהשעות תמיד יופיעו בשתי ספרות וכך גם הדקות. כלומר, אם השעה היתה שמונה ודקה בבוקר, השיטה תחזיר את המחרוזת: 08:01. אם הזמן מיוצג כזמן של 12 שעות, השיטה תוסיף גם am או pm בהתאמה. למשל, אם השעה היתה שמונה וחמש דקות בבוקר, השיטה תחזיר 08:01 am ואם השעה היתה תשע בערב השיטה תחזיר 09:00 pm.
הוסיפו את השיטות הבאות למחלקה:
public void setHours(int h) – השיטה תקבל כפרמטר מספר שעה ותציב אותו לתכונה. אם המספר אינו מייצג שעה חוקית, השיטה תציב בתכונה את 12 % h או 24 % h בהתאם לסוג הזמן (12 או 24 שעות). למשל, אם השיטה תופעל עם הפרמטר 25 בשעה של 24 שעות, התוצאה תהיה 2, ואם יועבר הפרמטר 13 בשעה של 12 שעות, התוצאה תהיה 1. שימו לב שבשעה של 12 שעות, תוצאת ההצבה הלא חוקית תמיד תהיה בזמן am.
public void setMinutes(int m) – השיטה תקבל כפרמטר מספר שעה ותציב אותו לתכונה. אם המספר אינו מייצג דקות חוקיות, השיטה תציב את הערך 60 % m.
public void switchFormat() – השיטה תשנה את פורמט השעה. אם השעה היא בפורמט 24 שעות, היא תשתנה ל-12 שעות וההיפך.
public boolean isMorning() – השיטה תחזיר true אם הזמן המיוצג ע"י האובייקט הוא זמן בוקר. בוקר מוגדר כשעות שבין 6:00 ל-11:00 בבוקר.

מחלקות ואובייקטים חלק ב'

קובץ תרגילים זה מכיל שאלות בנושא מחלקות, אובייקטים ושיטות בשפת ג'אווה. חלק מהשאלות הן תאורטיות ונועדו לעזור לקורא לבחון את הידע שלו בנושא. רוב השאלות הן שאלות כתיבת קוד, ומסודרות מהקל לקשה. כדאי מאוד להריץ את התשובות (גם את הקלות שבהן) ולוודא שאתם מבינים איך הקוד עובד.

שאלה 1

ענו על השאלות הבאות. השתדלו לנמק בצורה מלאה ולא מעורפלת:

1. מה זה Aliasing (הצבעה כפולה)?
2. מה זה Composition (הרכבה)?
3. מה זה בנאי העתקה (Copy Constructor)? לאיזה צורך נשתמש בו?
4. מתי נרצה להמנע מ-aliasing?
5. האם שיטה יכולה לקבל כפרמטר אובייקט?
6. האם שיטה יכולה להחזיר אובייקט באמצעות הערך המוחזר שלה?
7. מה משמעות המילה this?

שאלה 2

נתונה המחלקה A:

```
public class A {
    private int x;

    public int getX() { return x; }
}
```

ונתונה המחלקה B שמכילה אובייקט מסוג A כתכונה:

```
public class B {
    private A a1;
    public B() {
        a1 = new A();
    }
    // ...
}
```

בכל אחד מהסעיפים הבאים כתובה שיטה שתכתב במחלקה B אחרי הקוד המסומן ב- ...// . עבור כל שיטה כתבו האם היא עוברת קומפילציה או לא. אם השיטה לא עוברת קומפילציה, כתבו מה הסיבה.

א.

```
public void f() {
    System.out.println(x);
}
```

ב.

```
public void f() {
    System.out.println(a1.x);
}
```

ג.

```
public void f() {
    System.out.println(this.a1.x);
}
```


shay.tavor@gmail.com

ד.

```
public void f() {
    System.out.println(a1.getX());
}
```

ה.

```
public void f() {
    System.out.println(this.getX());
}
```

ו.

```
public void f() {
    System.out.println(this.a1.getX());
}
```

ז.

```
public void f(B b1) {
    this.a1 = b1.a1;
}
```

שאלה 3

נתונה המחלקה Car המייצגת מכונית. למכונית יש את התכונות הבאות:
מספר הרכב, שם בעלי הרכב, מספר הקילומטרים שעבר הרכב.

```
public class Car {
    private long carID;
    private String ownerName;
    private int km;
}
```

א. הוסיפו למחלקה בנאי העתקה. הבנאי יקבל אובייקט מסוג Car ויעתיק את נתוניו לאובייקט this.

ב. הוסיפו למחלקה שיטה שחתימתה –

public boolean equals(Car other)

השיטה תקבל אובייקט מסוג Car ותחזיר true אם הוא שווה ל-this ו-false אם לא. שתי מכוניות יקראו שוות אם יש להן את אותו מספר רכב.

שאלה 4

נתונה המחלקה Point המייצגת נקודה במישור:

```
public class Point {
    private int x, y;

    public Point(int x, int y) {
        this.x = x;
        this.y = y;
    }
    public Point(Point p) {
        x = p.x;
        y = p.y;
    }

    public int getX() { return x; }
    public void setX(int x) { this.x = x; }
    public int getY() { return y; }
    public void setY(int y) { this.y = y; }
```

```
    public boolean equals(Point p) {
        if(x == p.x && y == p.y)
            return true;
        return false;
    }
    public String toString() {
        return "(" + x + ", " + y + ")";
    }
}
```

נקודה מיוצגת ע"י קואורדינות ה-x וה-y שלה (מספרים שלמים). המחלקה מכילה בנאי שמקבל את ערכי הקואורדינטות ומציב אותם לתכונות, ובנאי העתקה. המחלקה מכילה שיטות get/set עבור התכונות. כמו כן מכילה המחלקה את השיטה equals שמשווה בין שתי נקודות, ואת השיטה toString שמחזירה את פרטי הנקודה במבנה של מחרוזת.

נתון ה-main הבא העושה שימוש במחלקה Point:

```
public class PointTester {
    public static void main(String[] args) {
        Point p1 = new Point((int)(Math.random()*10),
                             (int)(Math.random()*10));
        // ...

        System.out.println(p1 == p2);           // 1
        System.out.println(p1.equals(p2));      // 2
        p1 = p2;
        System.out.println(p1 == p2);           // 3
        p1.setX(20);
        System.out.println("X:" + p2.getX());   // 4
        p1 = null;
        System.out.println("X:" + p2.getX());   // 5
        System.out.println("X:" + p1.getX());   // 6
    }
}
```

התוכנית יוצרת אובייקט מסוג Point בשם p1 ומאתחלת את הקואורדינטות שלו לערכים אקראיים.

א. צרו ב-main אחרי השורה שמסומנת בהערה אובייקט חדש מסוג Point בשם p2 שיכיל את אותם ערכי קואורדינטות כמו p1. השתמשו בבנאי ההעתקה של Point.

ב. צרו אובייקט חדש מסוג Point בשם p3 שיכיל את אותם ערכי קואורדינטות כמו p1. השתמשו בשיטות get.

ג. עבור כל אחת מהשורות המסומנות 1 עד 6 כתבו מה יהיה הפלט והסבירו את תשובותיכם. שימו לב, השורות מהוות רצף בתוכנית, ולכן כל שורה משפיעה על השורה שאחריה, התייחסו לזה.

ד. הוסיפו למחלקה Point שיטה שחתימתה public void move(int dx, int dy). השיטה תקבל כפרמטרים ערכי x ו-y **ותוסיף** אותם לערכי x, y של האובייקט.

ה. הוסיפו למחלקה Point שיטה שחתימתה public double distance(Point p). השיטה תקבל כפרמטר אובייקט מסוג Point ותחזיר את מרחק הנקודה p מהנקודה this. חישוב המרחק נעשה ע"י הנוסחה:

$$d = \sqrt{(x1-x2)^2 + (y1-y2)^2}$$

כאשר x1, y1 הן הקואורדינטות של נקודה אחת ו-x2, y2 הן הקואורדינטות של הנקודה השניה.

ו. הוסיפו למחלקה Point שיטה שחתימתה public double distanceFromBase(). השיטה תחזיר את מרחק הנקודה מנקודת ראשית הצירים (0, 0). רמז – רצוי להשתמש בשיטה מהסעיף הקודם, חשבו כיצד.

שאלה 6

א. כתבו מחלקה בשם Adress שתייצג כתובת[1]. כתובת מיוצגת ע"י שם העיר והרחוב (מסוג String) ומספר הבית והמיקוד (מסוג int).

6. הוסיפו למחלקה בנאי שחתימתו:

public Address(String c, String s, int n, int z)

הבנאי יקבל כפרמטרים את נתוני הכתובת ויציב אותם לתכונות.

7. הוסיפו למחלקה בנאי שחתימתו:

public Address(String c, String s, int n)

הבנאי יקבל כפרמטרים את שם העיר, הרחוב ומספר הבית ויציב אותם לתכונות. בתכונת המיקוד הבנאי יציב 0.

8. הוסיפו שיטות get/set עבור כל אחת מהתכונות.
9. הוסיפו שיטת toString שתחזיר את פרטי הכתובת בפורמט הבא:

Address: [street] st., [number], [city], [zip]

למשל, אם הכתובת היא רחוב הרצל 34 ירושלים, מיקוד 778345, הפעלת השיטה תחזיר את המחרוזת:

Address: Herzl st., 34, Jerusalem, 778345

שימו לב, אם אין מיקוד בכתובת (כלומר המיקוד שווה לאפס) השיטה לא תחזיר את המיקוד, כלומר אחרי שם העיר לא יופיע פסיק ומיקוד.

10. הוסיפו למחלקה שיטה שחתימתה:

public boolean equals(Address other)

השיטה תקבל כפרמטר אובייקט מסוג Address ותשווה אותו לאובייקט this. שתי הכתובות יהיו שוות אם הן מכילות את אותם נתונים (עיר, רחוב, מספר ומיקוד).

ב. כתבו מחלקה בשם Person שתייצג בן אדם. לבן האדם יהיו התכונות הבאות:

name – שם האדם, מסוג String.

pAddress – כתובת האדם, מסוג Address.

הוסיפו למחלקה שלושה בנאים:

public Person(String name, String city, String street, int number, int zip) – הבנאי מקבל כפרמטרים את שם האדם ואת פרטי הכתובת שלו, ומאתחל בהם את התכונות.

public Person(String name, Address a) – הבנאי מקבל כפרמטרים את שם האדם, ואובייקט מסוג Address המייצג את כתובתו, ומאתחל בהם את התכונות.

public Person(Person other) – בנאי העתקה.

הוסיפו שיטת get עבור השם, ושיטות set/get עבור הכתובת. **המנעו מ-aliasing בשיטות אלו.**

הוסיפו למחלקה את השיטות הבאות:

public String toString() – השיטה תחזיר את פרטי האדם כמחרוזת בפורמט הבא: למשל, אם שם האדם הוא יוסי כהן, והוא גר ברחוב ויצמן 53 תל אביב, 6758, אזי השיטה תחזיר את המחרוזת:

Yossi Cohen, Address: Weizman st. 53, Tel Aviv, 6758

public boolean sameAddress(Person p) – השיטה מקבלת כפרמטר אובייקט מסוג Person ומחזירה true אם לשני בני האדם (p והאובייקט this) יש את אותה הכתובת.

שאלה 7

בשאלה זו נשתמש במחלקה Point משאלה 4.

כתבו מחלקה בשם Line שתייצג קו ישר במישור. הקו מיוצג ע"י שתי נקודות בניהן עובר הקו.

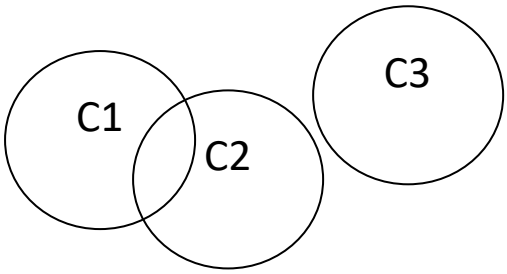
המחלקה תכיל שלושה בנאים:

[1] המחלקה Address נכתבה בשאלה 4 בקובץ "מחלקות ואובייקטים". למען הנוחות היא מובאת פה שוב.

public Line(int x1, int y1, int x2, int y2) – הבנאי יקבל את קואורדינטות נקודות הקצה של הקו ויאתחל בהן את התכונות. שימו לב, אם שתי הנקודות נמצאות בדיוק באותו מקום, יש לשנות אחת מהן כרצונכם לערך דיפולטיבי כלשהו.
public Line(Point p1, Point p2) – הבנאי יקבל שני אובייקטים מסוג Point שמייצגים את נקודות הקצה של הקו, ויאתחל בהן את התכונות. שימו לב, אם שתי הנקודות זהות, יש לשנות את מיקום אחת מהן כרצונכם לערך דיפולטיבי כלשהו. המנעו מ-aliasing!
public Line(Line other) – בנאי העתקה.
הוסיפו למחלקה את השיטות:
public Point getP1() – השיטה מחזירה את אובייקט אחת הנקודות של הישר.
public Point getP2() – השיטה מחזירה את אובייקט אחת הנקודות של הישר.
public void setP1(Point p1) – השיטה מעדכנת את ערך אחת הנקודות של הישר.
public void setP2(Point p2) – השיטה מעדכנת את ערך אחת הנקודות של הישר.
בכל השיטות האלו המנעו מ-aliasing.
הוסיפו למחלקה את השיטות הבאות:
public String toString() – השיטה תחזיר את פרטי הקו בצורת מחרוזת. למשל, אם הקו נמצא בין הנקודות (1, 1) ו-(4, 5) קריאה לשיטה תחזיר את המחרוזת:
"Line between: (1, 1) and (4, 5)"
public double length() – השיטה תחזיר את אורך הקו.

שאלה 8

בשאלה זו נשתמש במחלקה Point משאלה 4.
כתבו מחלקה בשם Circle שתייצג מעגל. מעגל ייוצג ע"י הרדיוס שלו (מסוג double) ונקודת המרכז שלו (מסוג Point).
הוסיפו למחלקה את הבנאים הבאים:
public Circle(double r, int x, int y) – הבנאי יקבל את ערכי הרדיוס וקואורדינטות נקודת המרכז, ויאתחל בהם את התכונות. אם הרדיוס אינו חיובי, הבנאי יציב 1 בתכונת הרדיוס.
public Circle(double r, Point center) – הבנאי יקבל את ערך הרדיוס ואובייקט מסוג Point שמייצג את נקודת מרכז המעגל. אם הרדיוס אינו חיובי, הבנאי יציב 1 בתכונת הרדיוס.
public Circle(Circle other) – בנאי העתקה.
הוסיפו למחלקה שיטות get/set עבור כל אחת מהתכונות (המנעו מ-aliasing).
הוסיפו למחלקה את השיטות הבאות:
public String toString() – השיטה תחזיר את פרטי המעגל במחרוזת בפורמט הבא: למשל, אם המעגל ברדיוס 4 ומרכזו בנקודה (1, 2) השיטה תחזיר את המחרוזת:
Circle at: (1, 2) radius 4.
public move(int dx, int dy) – השיטה תקבל כפרמטרים ערכי תזוזה על ציר ה-x ועל ציר ה-y ותזיז את המעגל בערכים אלו.
public boolean intersect(Circle c) – השיטה תקבל כפרמטר אובייקט מסוג Circle ותחזיר true אם שני המעגלים (c ו-this) נחתכים, ו-false אם לא. הציור הבא מראה שלושה מעגלים, c1, c2, c3. אם נפעיל את השיטה על c1 ונעביר כפרמטר את c2 נקבל true. אם נפעיל את השיטה על c1 ונעביר את c3 כפרמטר נקבל false.



שאלה 9

א. כתבו מחלקה בשם Date שתייצג תאריך. התאריך ייוצג ע"י שלוש תכונות – יום, חודש ושנה (כל אחת מהתכונות מסוג int).
הוסיפו למחלקה את הבנאים הבאים:
public Date(int day, int month, int year) – הבנאי יקבל כפרמטרים את ערכי היום, חודש והשנה, ויציב אותם לתכונות. אם אחד מהפרמטרים אינו חוקי, הבנאי יציב במקומו 1.
public Date(Date other) – בנאי העתקה.
הוסיפו למחלקה שיטת get/set עבור כל אחת מהתכונות. שימו לב שבשיטות ה-set אסור לאפשר הכנסה של ערכים לא חוקיים. אם המשתמש ינסה להכניס, למשל, יום שערכו 89, השיטה לא תשנה את ערך היום.
הוסיפו למחלקה את השיטות הבאות:
public String toString() – השיטה תחזיר את פרטי התאריך כמחרוזת בפורמט הבא: אם למשל התאריך הוא 2 לינואר 2011, השיטה תחזיר את המחרוזת:
2/1/2011
public boolean equals(Date d) – השיטה תקבל כפרמטר אובייקט מסוג Date ותחזיר true אם הוא שווה ל-this ו-false אחרת.
public boolean before(Date d) – השיטה תקבל כפרמטר אובייקט מסוג Date ותחזיר true אם התאריך המיוצג ע"י this הוא **לפני (כרונולוגית)** התאריך המיוצג ע"י d. למשל, אם האובייקט d מכיל את התאריך 2/4/2011 והאובייקט this מכיל את התאריך 1/1/2011 השיטה תחזיר true.
public boolean after(Date d) – השיטה תקבל כפרמטר אובייקט מסוג Date ותחזיר true אם האובייקט המיוצג ע"י this הוא **אחרי (כרונולוגית)** לתאריך המיוצג ע"י d.
ב. כתבו מחלקה בשם Exam המייצגת בחינה בקורס. המחלקה תכיל את התכונות הבאות:
courseName – שם הקורס בו נערכת הבחינה, מסוג String.
examDate – התאריך בו נערכך הבחינה, מסוג Date.
הוסיפו למחלקה את הבנאים הבאים:
public Exam(String name, Date d) – הבנאי יקבל כפרמטרים את שם הקורס ותאריך הבחינה ויציב אותם לתכונות.
public Exam(String name, int day, int month, int hour) – הבנאי יקבל כפרמטרים את שם הקורס, ואת תאריך הבחינה לפי יום, חודש ושנה, ויציב אותם לתכונות. רמז – יש להפוך את פרמטרי התאריך לאובייקט מסוג Date.
public Exam(Exam other) – בנאי העתקה.
הוסיפו שיטות get/set עבור כל אחת מהתכונות.
הוסיפו למחלקה את השיטות הבאות:
public String toString() – השיטה תחזיר מחרוזת המייצגת את פרטי הבחינה – שם הקורס והתאריך. למשל, אם הבחינה היא בקורס ששמו OOP בתאריך 1/1/2011 השיטה תחזיר את המחרוזת:
Exam in course OOP at 1/1/2011
public boolean sameDate(Exam other) – השיטה תקבל כפרמטר אובייקט מסוג Exam ותחזיר true אם תאריך הבחינה של this שווה לתאריך הבחינה של other.
public boolean before(Exam other) – השיטה תקבל כפרמטר אובייקט מסוג Exam ותחזיר true אם תאריך הבחינה של this קודם לתאריך הבחינה של other.
public boolean differentSemesters(Exam other) – השיטה תקבל כפרמטר אובייקט מסוג Exam ותחזיר true אם גם other וגם this מייצגים בחינות באותו הקורס (כלומר, שם הקורס זהה) אבל בסמסטרים שונים (כלומר, השנה שונה).

לולאות

קובץ תרגילים זה מכיל שאלות בנושא לולאות בשפת ג'אווה. חלק מהשאלות הן תאורטיות ונועדו לעזור לקורא לבחון את הידע שלו בנושא. רוב השאלות הן שאלות כתיבת קוד, ומסודרות מהקל לקשה. כדאי מאוד להריץ את התשובות (גם את הקלות שבהן) ולוודא שאתם מבינים איך הקוד עובד.

שאלה 1

ענו על השאלות הבאות. השתדלו לנמק בצורה ברורה ולא מעורפלת:

1. מה זאת לולאה? לשם מה נצטרך להשתמש בלולאה?
2. מה ההבדל בין לולאת for ללולאת while? לאילו צרכים נשתמש בכל אחת מהן?
3. מה ההבדל בין לולאת while ללולאת do while? לאילו צרכים נשתמש בכל אחת מהן?
4. מהי פקודת break? לשם מה נשתמש בפקודה זו?

שאלה 2

בכל אחד מהסעיפים הבאים נתונה שיטה. הסבירו מה עושה כל שיטה. השתדלו לנסח את תשובותיכם בצורה ברורה ולא מעורפלת. הדרכה – כדאי מאוד לכתוב על נייר את המשתנים המשתתפים בלולאה, ולעקוב אחרי השינויים שלהם במהלך הלולאה. כדי להסביר מה השיטה עושה נסו להבין מה פלט השיטה – אם השיטה מדפיסה משהו – מה היא מדפיסה בסופו של דבר? אם השיטה מחזירה ערך, מה משמעותו של הערך החוזר?

א.

```
public void f1()
{
    for(int i = 1; i <= 10; i++)
        System.out.println("*");
}
```

ב.

```
public int f2(int x)
{
    int c = 0;
    for(int i = 0; i <= x; i++)
        if(i % 3 == 0)
            c++;
    return c;
}
```

ג.

```
public int f3(int x)
{
    int c = 0;
    while(x > 0)
    {
        x /= 10;
        c++;
    }
    return c;
}
```


shay.tavor@gmail.com

ד.

```
public void f4(int x)
{
    int i = 1;
    boolean flip = true;
    while(i <= x)
    {
        char c;
        if(flip)
            c = 'X';
        else
            c = 'O';
        flip = !flip;
        System.out.println(c);
        i++;
    }
}
```

ה.

```
public void f5(int x)
{
    int i = 0;
    do {
        System.out.print("*");
        i++;
    }while(i < x);

    System.out.println();
    for(int j = 1; j < x; j++)
    {
        System.out.print("*");
        int k = 1;
        while(k < x - 1)
        {
            System.out.print(" ");  // printing space
            k++;
        }
        System.out.print("*");
    }

    System.out.println();
    for(int j = 0; j < x; j++)
        System.out.print("*");
}
```

שאלה 3

א. כתבו מחדש את השיטה f1 משאלה 2 סעיף א', תוך שימוש בלולאת while.
ב. כתבו מחדש את השיטה f2 משאלה 2 סעיף ב' תוך שימוש בלולאת do while.
ג. כתבו מחדש את השיטה f3 משאלה 2 סעיף ג' תוך שימוש בלולאת for.
ד. התבוננו בשיטה f3 משאלה 2 סעיף ג' – מה היה קורה אם היינו כותבים את התנאי בלולאה בצורה הזאת:

while(x >= 0)

shay.tavor@gmail.com

שאלה 4

נתונה השיטה הבאה:

```
public void g()
{
    for(int i = 0; i < 10; i++)
        System.out.print(i + "\t");
    System.out.println("Last value: " + i);
}
```

שיטה זו לא עוברת קומפילציה. הסבירו מה בעית הקומפילציה ותקנו את השיטה.

****

כתבו מחלקה בשם LoopsMethods. את התשובות לשאלות הבאות כתבו כשיטות בתוך מחלקה זו.

שאלה 5

כתבו שיטה שחתימתה:

public int factorial(int n)

השיטה תקבל כפרמטר מספר שלם n ותחזיר את העצרת של n. עצרת מוגדרת כמכפלת כל המספרים מ-1 עד n. למשל,

4! = 1*2*3*4 = 24

5! = 1*2*3*4*5 = 120

ובאופן כללי:

n! = 1*2*3*...*n

אם הפרמטר n אינו חיובי, השיטה תחזיר 1-.

שאלה 6

כתבו שיטה שחתימתה:

public double average()

השיטה תקלוט מהמשתמש סדרה של מספרים חיוביים ממשיים, עד המספר 1-, ותחזיר את ממוצע המספרים. תזכורת – ממוצע המספרים הוא סכום המספרים חלקי כמות המספרים.

למשל, עבור הקלט הבא:

3.2 5.6 1.0 7.9 -1

יוחזר הממוצע: 4.425

שימו לב שהמספר 1- אינו מהווה חלק מהסדרה, ומשמש רק לסמן לתוכנית שהמשתמש סיים להקליד את הנתונים.

שאלה 7

כתבו שיטה שחתימתה:

public int howManyEven(int num)

השיטה תקבל כפרמטר מספר שלם, ותחזיר כמה ספרות במספר הן זוגיות. למשל, עבור המספר 25643 השיטה תחזיר 3 כיוון שיש 3 ספרות זוגיות במספר (2, 6, 4).

שאלה 8

כתבו שיטה שחתימתה:

public boolean isAscending(int num)

השיטה תקבל כפרמטר מספר שלם, ותחזיר true אם הספרות במספר מסודרות בסדר עולה ממש, משמאל לימין. למשל, המספרים הבאים הם עולים: 1234, 25689, 267, 3. המספרים הבאים אינם מספרים עולים: 264, 2556, 7654.

שאלה 9

כתבו שיטה שחתימתה:

public int maxDigit(int num)

השיטה תקבל כפרמטר מספר שלם, ותחזיר את הספרה המקסימלית במספר. למשל, עבור המספר 352875 השיטה תחזיר 8.

שאלה 10

כתבו שיטה שחתימתה:

public int howManyOccurences(int num, int d)

השיטה תקבל כפרמטרים מספר שלם num וספרה d. השיטה תחזיר כמה פעמים מופיעה הספרה d במספר num.

למשל, אם המספר הוא 1434624 ו-d = 4 השיטה תחזיר 3. אם עבור אותו מספר d = 3 השיטה תחזיר 1, ואם d = 9 השיטה תחזיר 0.

שאלה 11

כתבו שיטה שחתימתה:

public int filter(int num, int d)

השיטה תקבל כפרמטרים מספר שלם num וספרה d. השיטה תחזיר מספר חדש שמכיל את כל הספרות מ-num שגדולות או שוות ל-d.

למשל, אם num = 316462 ו- d = 3 השיטה תחזיר את המספר 3646.

שאלה 12

נגדיר – **מספר עולה יורד** הוא מספר שלם שהספרות בו משמאל לימין ממויינות בסדר עולה עד ספרה מסויימת, ואז ממויינות בסדר יורד עד סוף המספר. למשל, המספרים הבאים הם עולים יורדים: 12343, 15543, 1567, 7654 (שימו לב שגם מספר שהוא רק עולה או רק יורד נחשב לעולה יורד).

א. כתבו שיטה שחתימתה:

public boolean isUpdownNumber(int num)

השיטה תקבל כפרמטר מספר שלם, ותחזיר true אם המספר עולה יורד, ו-false אם לא.

ב. נגדיר – **השיא** של מספר עולה יורד הוא המיקום במספר בו נמצאת הספרה הגדולה ביותר (כלומר, שעד אליה הספרות עולות, וממנה הספרות יורדות). כתבו שיטה שחתימתה:

public int peak(int num)

השיטה תקבל כפרמטר מספר, ותחזיר את השיא שלו כמתואר לעיל. אם המספר אינו עולה יורד, השיטה תחזיר 1-. אם המספר עולה יורד, השיטה תחזיר את מיקום הספרה שמהווה את השיא.

שימו לב, הספרה השמאלית ביותר נספרת כספרה מספר 1, הספרה שאחריה נספרת כספרה מספר 2 וכו'. למשל, עבור המספר 13432 השיטה תחזיר 3 (הספרה 4 היא השיא, והיא נמצאת במיקום 3 במספר). עבור המספר 16432 השיטה תחזיר 2. עבור המספר 1578 השיטה תחזיר 4. עבור המספר 12324 השיטה תחזיר 1-.

שאלה 13

כתבו שיטה שחתימתה:

public int reverse(int num)

השיטה תקבל כפרמטר מספר שלם ותחזיר את המספר ההפוך (כלומר, מספר שסדר ספרותיו הפוך מהמספר המקורי). למשל, אם num = 3628 השיטה תחזיר 8263. אם num = 4 השיטה תחזיר 4.

שאלה 14

כתבו שיטה שחתימתה:

public void printMultiply()

השיטה תדפיס על המסך את לוח הכפל של 10X10. ההדפסה צריכה להיות מורכבת מעשר שורות ועשר עמודות, כאשר בכל תא נמצאת תוצאת המכפלה של השורה והעמודה הרלבנטיים:

```
1     2     3     ...     10
2     4     6     ...     20
3     6     9     ...     30
...
10    20    30    ...     100
```

השתמשו בסימן "\t" כדי להדפיס טאב, על מנת שההדפסות יהיו מיושרות.

שאלה 15

א. כתבו שיטה שחתימתה:

public void printSquare(int n)

השיטה תקבל כפרמטר מספר שלם n ותדפיס ריבוע מלא בכוכביות בגודל nXn. למשל, אם n=4 השיטה תדפיס על המסך את הפלט:

```
****
****
****
****
```

אם הפרמטר n קטן או שווה לאפס, השיטה תדפיס על המסך את הפלט "Invalid Input".

ב. כתבו שיטה שחתימתה:

public void triangle1(int n)

השיטה תקבל כפרמטר מספר שלם n ותדפיס משולש ישר זווית של כוכביות. למשל, אם n=4 השיטה תדפיס על המסך את הפלט:

```
*
**
***
****
```

אם הפרמטר n קטן או שווה לאפס, השיטה תדפיס על המסך את הפלט "Invalid Input".

ג. כתבו שיטה שחתימתה:

public void triangle2(int n)

השיטה תקבל כפרמטר מספר שלם n ותדפיס משולש ישר זווית של כוכביות. למשל, אם n=4 השיטה תדפיס על המסך את הפלט:

```
****
***
**
*
```

אם הפרמטר n קטן או שווה לאפס, השיטה תדפיס על המסך את הפלט "Invalid Input".

ד. כתבו שיטה שחתימתה:

public void triangle3(int n)

השיטה תקבל כפרמטר מספר שלם n ותדפיס את הצורה הבאה: למשל, אם n=4 השיטה תדפיס על המסך את הפלט:

```
*
**
***
****
****
***
**
*
```

אם הפרמטר n קטן או שווה לאפס, השיטה תדפיס על המסך את הפלט "Invalid Input".

ה. כתבו שיטה שחתימתה:

public void line1(int n)

השיטה תקבל כפרמטר מספר שלם n ותדפיס על המסך קו אלכסוני של n כוכביות משמאל לימין.
למשל, אם n=4 השיטה תדפיס את הפלט:

```
*
        *
                *
                        *
```

השתמשו בסימן "\t" כדי ליישר את הרווחים.
אם הפרמטר n קטן או שווה לאפס, השיטה תדפיס על המסך את הפלט "Invalid Input".

ו. כתבו שיטה שחתימתה:

public void line2(int n)

השיטה תקבל כפרמטר מספר שלם n ותדפיס על המסך קו אלכסוני של n כוכביות מימין לשמאל.
למשל, אם n=4 השיטה תדפיס את הפלט:

```
                        *
                *
        *
*
```

השתמשו בסימן "\t" כדי ליישר את הרווחים.
אם הפרמטר n קטן או שווה לאפס, השיטה תדפיס על המסך את הפלט "Invalid Input".

שאלה 16

כתבו שיטה שחתימתה:

public int power(int n, int exp)

השיטה תקבל כפרמטר מספר שלם n ומספר שלם exp ותחזיר את התוצאה של n בחזקת exp.
למשל, אם n=2 ו-exp=3 השיטה תחזיר את המספר 8, כלומר, 2 בחזקת 3.

שאלה 17

סדרת פיבונאצ'י היא סדרת מספרים שמוגדרת באופן הבא:

A0 = 0
A1 = 1
A2 = A1 + A0 = 1
A3 = A2 + A1 = 2
A4 = A3 + A2 = 3
A5 = A4 + A3 = 5

כלומר, האיבר הראשון בסדרה (מסומן כ A0) שווה לאפס, והאיבר השני (מסומן כ-A1) שווה לאחד.
כל איבר אחר בסדרה שווה לסכום שני האיברים הקודמים לו. באופן כללי, האיבר ה-n בסדרה הוא:
An = An-1 + An-2

כתבו שיטה שחתימתה:

```
public int fibo(int n)
```

השיטה תקבל כפרמטר מספר שלם n ותחזיר את האיבר ה-n בסדרת פיבונאצ'י.

שאלה 18

מספר ראשוני הוא מספר שלם שמתחלק ללא שארית רק בעצמו ובאחד. למשל, המספרים 3, 5, 7, 13 הם מספרים ראשוניים. המספרים 6, 9, 21 הם לא ראשוניים.

א. כתבו שיטה שחתימתה:

```
public boolean isPrime(int n)
```

השיטה תקבל כפרמטר מספר שלם, ותחזיר true אם המספר הוא ראשוני, ו-false אם לא.

ב. כתבו שיטה שחתימתה:

```
public void printAllPrimes(int a, int b)
```

השיטה תקבל כפרמטרים שני מספרים a, b ותדפיס על המסך את כל המספרים הראשוניים בין a ל-b כולל. למשל, אם a = 3 ו- b = 10 השיטה תדפיס את המספרים 3, 5, 7.

ג. כתבו שיטה שחתימתה:

```
public int differencePrimes(int a, int b)
```

השיטה תקבל כפרמטרים שני מספרים a, b ותחזיר את ההפרש בין המספר הראשוני הגדול ביותר בניהם והמספר הראשוני הקטן ביותר בניהם. למשל, אם a = 4 ו- b = 20 השיטה תחזיר את המספר 14, כיוון שהמספר הראשוני הגדול ביותר בתחום הוא 19, והקטן ביותר הוא 5.

שאלה 19

כתבו שיטה שחתימתה:

```
public int formula1(int n)
```

השיטה תקבל כפרמטר מספר שלם n ותחזיר את תוצאת הנוסחה הבאה:

$$n + n^2 + n^3 + \ldots + n^n$$

למשל, אם n=4 השיטה תחזיר את המספר 340, שכן: $4 + 4^2 + 4^3 + 4^4 = 340$

שאלה 20

כתבו שיטה שחתימתה:

```
public int formula2(int n)
```

השיטה תקבל כפרמטר מספר שלם n ותחזיר את סכום הסדרה הבאה:

1 – 2 + 3 – 4 + 5 - … + n

למשל, אם n = 5 השיטה תחזיר 3 שכן:

1 – 2 + 3 – 4 + 5 = 3

שאלה 21

כתבו שיטה שחתימתה:

public boolean uniqueDigits(int n)

השיטה תקבל כפרמטר מספר שלם ותחזיר true אם כל ספרה במספר מופיעה רק פעם אחת ו-false אחרת. למשל, אם n = 24198 השיטה תחזיר true ואם n = 1951 השיטה תחזיר false.

שאלה 22

מספר ארמסטרונג הוא מספר בן שלוש ספרות, שסכום החזקות השלישיות של כל ספרה שלו שווה למספר עצמו. למשל, המספר 153 הוא מספר ארמסטרונג כיוון ש-

$153 = 1^3 + 5^3 + 3^3$

א. כתבו שיטה שחתימתה –

public boolean isArmstrong(int n)

השיטה תקבל כפרמטר מספר שלם n ותחזיר true אם הוא מספר ארמסטרונג ו-false אם לא.

ב. כתבו שיטה שחתימתה –

public void printAllArmstrongs()

השיטה תדפיס על המסך את כל מספרי הארמסטרונג הקיימים.

shay.tavor@gmail.com

מערכים

קובץ תרגילים זה מכיל שאלות בנושא מערכים - חד ממדיים, דו ממדיים ומערכים של אובייקטים בשפת ג'אווה. חלק מהשאלות הן תאורטיות ונועדו לעזור לקורא לבחון את הידע שלו בנושא. רוב השאלות הן שאלות כתיבת קוד, ומסודרות מהקל לקשה. כדאי מאוד להריץ את התשובות (גם את הקלות שבהן) ולוודא שאתם מבינים איך הקוד עובד.

שאלה 1

ענו על השאלות הבאות. השתדלו לנמק בצורה ברורה ולא מעורפלת:

5. מה זה מערך? לשם מה נצטרך להשתמש במערך?
6. איך יוצרים מערך?
7. איך ניגשים למשתנים השמורים במערך?
8. מהו הערך length?
9. כמה תאים יש במערך?

שאלה 2

נתון הקוד הבא:

```
import java.util.Scanner;
public class Tester {
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        int[] a = new int[4];

        // …
    }
}
```

הוסיפו לתוכנית, אחרי השורה המסומנת בהערה, קוד העונה לכל אחד מהסעיפים הבאים.

א. הציבו את המספר 3 בתא הראשון במערך.

ב. הציבו בתא השני מהמערך מספר שנקלט מהמשתמש (השתמשו באובייקט scan שהוגדר בתוכנית).

ג. הציבו בתא השלישי במערך את סכום התא הראשון והתא השני.

ד. הציבו בתא הרביעי במערך את מכפלת שלושת המספרים שבתאים הראשונים.

ה. הדפיסו על המסך את המספרים שבמערך, מהמספר הראשון ועד האחרון, כאשר המספרים מופרדים בפסיקים אחד מהשני.

שאלה 3

בכל אחד מהסעיפים ישנו קטע קוד שעובר קומפילציה אולם עף על שגיאת ריצה. הסבירו מה השגיאה והציעו דרך לתקן אותה.

א.

```
int[] arr = new int[5];
for(int i = 0; i <= 5; i++)
    arr[i] = 2;
```

ב.

```
int[] b;
b[0] = 1;
```

ג.

```
double[] m = new double[5];
for(int i = m.length; i >= 0; i--)
    m[i] = 8.0;
```

****

כתבו מחלקה בשם OneDArraysMethods. את התשובות לשאלות 4 עד 21 כתבו כשיטות בתוך מחלקה זו.

שאלה 4

כתבו שיטה שחתימתה:

public void inputArray(int[] a)

השיטה תקבל כפרמטר מערך של מספרים שלמים a ותקלוט מהמשתמש נתונים לתוך מערך זה. שימו לב, השיטה צריכה לבקש מהמשתמש להזין בדיוק את כמות הנתונים הדרושה כדי למלא את המערך.

שאלה 5

כתבו שיטה שחתימתה:

public void printArray(int[] a)

השיטה תקבל כפרמטר מערך של מספרים שלמים a ותדפיס על המסך את איברי המערך מהראשון לאחרון בשורה אחת, כאשר האיברים יופרדו בטאבים (השתמשו בסימן "\t" כדי להדפיס טאב) אחד מהשני.

שאלה 6

כתבו שיטה שחתימתה:

public void printBack(int[] a)

השיטה תקבל כפרמטר מערך של מספרים שלמים a ותדפיס על המסך את איברי המערך מהסוף להתחלה בשורה אחת, כאשר האיברים יופרדו בטאבים (השתמשו בסימן "\t" כדי להדפיס טאב) אחד מהשני.

שאלה 7

כתבו שיטה שחתימתה:

public double average(double[] dArr)

השיטה תקבל כפרמטר מערך של מספרים ממשיים ותחזיר את ממוצע האיברים במערך.

שאלה 8

כתבו שיטה שחתימתה:

public int maxInArray(int[] arr)

השיטה תקבל כפרמטר מערך של מספרים שלמים, ותחזיר את ערך האיבר המקסימלי במערך.

שאלה 9

כתבו שיטה שחתימתה:

public int difference(int[] arr)

השיטה תקבל כפרמטר מערך של מספרים שלמים, ותחזיר את ההפרש (תוצאת פעולת החיסור) בין האיבר המקסימלי לאיבר המינימלי במערך.

שאלה 10

כתבו שיטה שחתימתה:

public boolean isSorted(int[] a)

השיטה תקבל כפרמטר מערך של מספרים שלמים, ותחזיר true אם המערך ממויין בסדר עולה, ו-false אם לא.

למשל, עבור המערך {1, 3, 3, 5, 7} (משמאל לימין) השיטה תחזיר true ועבור המערך {4, 6, 7, 3, 8} השיטה תחזיר false.

שאלה 11

כתבו שיטה שחתימתה:

public int countNums(double[] dArr, double num)

השיטה תקבל כפרמטרים מערך של מספרים ממשיים ומספר num. השיטה תחזיר כמה איברים במערך גדולים או שווים ל-num. למשל, אם dArr = {2, 5, 1, 8, 4, 5} ו- num = 5 השיטה תחזיר 3.

שאלה 12

כתבו שיטה שחתימתה:

public int countOdd(int[] nums)

השיטה תקבל כפרמטר מערך של מספרים שלמים, ותחזיר כמה מספרים אי זוגיים יש במערך.

למשל, אם nums = {3, 5, 2, 7, 8} השיטה תחזיר 3.

שאלה 13

כתבו שיטה שחתימתה:

public boolean isPalindrom(int[] a)

השיטה תקבל כפרמטר מערך של מספרים שלמים, ותחזיר true אם המערך הוא פאלינדרום, ו-false אם לא. מערך פאלינדרום הוא מערך שאפשר לקרוא אותו מההתחלה לסוף ומהסוף להתחלה באותה דרך. למשל, המערך הבא הוא פאלינדרום – {1, 5, 3, 7, 3, 5, 1}.

שאלה 14

א. כתבו שיטה שחתימתה:

public int[] createArray1(int size)

השיטה תקבל כפרמטר מספר חיובי שלם size, ותחזיר מערך של int בגודל זה.

ב. כתבו שיטה שחתימתה:

public int[] createArray2(int size, int num)

השיטה תקבל כפרמטרים מספרים חיוביים שלמים size ו-num. השיטה תחזיר מערך של int בגודל size כאשר תאיו מכילים את הכפולות של num מ-1 ועד size.

למשל, אם size = 5 ו- num = 3 השיטה תחזיר את המערך {3, 6, 9, 12, 15}.

ג. כתבו שיטה שחתימתה:

public int[] createArray3(int[] a)

השיטה תקבל כפרמטר מערך של מספרים שלמים, ותחזיר מערך שגודלו פי שניים מ-a ומכיל את ערכי התאים של a בחצי הראשון שלו, ואת ערכי התאים של a בסדר הפוך, בחציו השני.

למשל, אם a = {1, 5, 3, 8} השיטה תחזיר את המערך {1, 5, 3, 8, 8, 3, 5, 1}.

שאלה 15

כתבו שיטה שחתימתה:

public void arrayShift(int[] a)

השיטה תקבל כפרמטר מערך של מספרים שלמים a ותזיז את כל אחד מאיבריו צעד אחד ימינה. כלומר, המספר שהיה בתא מספר 0 יעבור לתא מספר 1, המספר בתא מספר 1 יעבור לתא מספר 2 וכן הלאה. המספר שהיה בתא האחרון, יעבור לתא 0.

למשל, אם a = {7, 4, 9, 1, 5} אחרי פעולת השיטה המערך יראה כך: {5, 7, 4, 9, 1}.

שאלה 16

כתבו שיטה שחתימתה:

public int[] merge(int[] a, int[] b)

השיטה תקבל כפרמטרים שני מערכים **ממויינים בסדר עולה** (כלומר מהקטן לגדול). השיטה תחזיר מערך ממויין בסדר עולה שמכיל את כל איברי a ו-b. שימו לב, המערכים a ו-b לא בהכרח באותו הגודל.

למשל, אם a = {1, 5, 7, 9} ו- b = {2, 3, 7, 10, 11} השיטה תחזיר את המערך:

{1, 2, 3, 5, 7, 7, 9, 10, 11}

שאלה 17

א. כתבו שיטה שחתימתה:

public boolean allDifferent(int[] a)

השיטה תקבל כפרמטר מערך של מספרים שלמים, ותחזיר true אם אין במערך מספרים שחוזרים על עצמם. למשל, אם a = {3, 1, 6, 8} השיטה תחזיר true, ואם a = {4, 1, 4, 5} השיטה תחזיר false.

ב. כתבו שיטה שחתימתה:

public int mostPopular(int[] a)

השיטה תקבל כפרמטר מערך של מספרים שלמים a ותחזיר את המספר שמופיע הכי הרבה פעמים במערך. אם יש כמה מספרים שמופיעים אותה כמות של פעמים, השיטה תחזיר אחד מהם לבחירתכם. למשל, אם a = {2, 1, 6, 1, 2, 7, 6, 1} השיטה תחזיר 1 (כי הוא מופיע 3 פעמים).

שאלה 18

א. כתבו שיטה שחתימתה:

public int[] unify(int[] a, int[] b)

השיטה מקבלת כפרמטר שני מערכים של מספרים שלמים a, b. נתון שבכל מערך כל מספר מופיע רק פעם אחת (כלומר המספרים לא חוזרים אחד על השני במערך מסויים). השיטה תחזיר מערך שמכיל את כל האיברים שמופיעים בשני המערכים, כל מספר רק פעם אחת. כלומר המערך החוזר יכיל את האיברים של a וגם את האיברים של b.

למשל, אם a = {4, 1, 8, 5, 7} ו-b = {7, 2, 10, 5} השיטה תחזיר את המערך {4, 1, 8, 5, 7, 2, 10}.

שימו לב – המערך החוזר לא חייב להיות בדיוק בגודל של מספר האיברים, הוא יכול להיות גם גדול יותר ממספר האיברים שבאמת יהיו בתוכו.

ב. כתבו שיטה שחתימתה:

public int[] intersect(int[] a, int[] b)

השיטה מקבלת כפרמטר שני מערכים של מספרים שלמים a, b. נתון שבכל מערך כל מספר מופיע רק פעם אחת (כלומר המספרים לא חוזרים אחד על השני במערך מסויים). השיטה תחזיר מערך שמכיל את כל האיברים שמופיעים בשני המערכים ביחד. כלומר מספר שמופיע במערך a ולא מופיע במערך b, למשל, לא יהיה במערך החוזר.

למשל, אם a = {4, 1, 8, 5, 7} ו-b = {7, 2, 10, 5} השיטה תחזיר את המערך {7, 5}.

ג. כתבו שיטה שחתימתה:

public int[] minus(int[] a, int[] b)

השיטה מקבלת כפרמטר שני מערכים של מספרים שלמים a, b. נתון שבכל מערך כל מספר מופיע רק פעם אחת (כלומר המספרים לא חוזרים אחד על השני במערך מסויים). השיטה תחזיר מערך שמכיל את כל האיברים שמופיעים בשני במערך a ולא מופיעים במערך b.

למשל, אם a = {4, 1, 8, 5, 7} ו-b = {7, 2, 10, 5} השיטה תחזיר את המערך {4, 1, 8}.

שאלה 19

כתבו שיטה שחתימתה:

public boolean equals(int[] a, int[] b)

השיטה תקבל שני מערכים של מספרים שלמים a, b ותחזיר true אם הם **שווים לחלוטין**. כלומר, מכילים את אותם איברים ובאותו סדר בדיוק. שימו לב שהמערכים לא חייבים להיות באותו גודל. במקרה והמערכים אינם שווים בגודלם, השיטה תחזיר false.

למשל, אם a = {1, 2, 3, 4}, b = {1, 2, 3, 4} השיטה תחזיר true.

אם a = {1, 2, 3, 4}, b = {3, 1, 2, 4} השיטה תחזיר false.

אם a = {1, 2, 3, 4}, b = {1, 2, 3} השיטה תחזיר false.

שאלה 20

כתבו שיטה שחתימתה:

public boolean twoNumbers(int[] a, int num)

השיטה תקבל כפרמטרים מערך של מספרים שלמים a ומספר שלם num. השיטה תחזיר true אם קיימים במערך זוג איברים כלשהם שסכומם שווה ל-num.

למשל, אם a = {5, 1, 7, 9, 4, 2} ו- num = 9 השיטה תחזיר true (כי למשל 4+5=9) שימו לב, גם אם יש יותר מזוג איברים אחד שסכומו הוא num השיטה תחזיר true.

שאלה 21

א. כתבו שיטה שחתימתה:

public int findNum(int[] a, int num)

השיטה תקבל כפרמטרים מערך של מספרים שלמים a ומספר שלם num. אם המספר num נמצא במערך, השיטה תחזיר את מיקומו במערך (האינדקס שלו). אם המספר אינו קיים, השיטה תחזיר 1-.

למשל, אם a = {4, 1, 8, 9, 14} ו-num = 8 השיטה תחזיר 2 (המספר 8 נמצא באינדקס 2 במערך).

אם num = 20 השיטה תחזיר 1- (המספר 20 אינו קיים במערך).

ב. **חיפוש בינארי** הוא אלגוריתם שמחפש מספר num בתוך מערך ממויין בסדר עולה. האלגוריתם מבצע את הפעולות הבאות:

1. מצא את האינדקס האמצעי במערך.
2. אם המספר num גדול מהמספר שבאמצע המערך, חפש את המספר בחצי העליון.
3. אם המספר num קטן מהמספר שבאמצע המערך, חפש את המספר בחצי התחתון.
4. אם המספר num שווה למספר שבאמצע המערך, החזר את האינדקס שלו.

בכל איטרציה מחפשים את המספר בחלק קטן והולך של המערך, בכל איטרציה מקטינים את המערך פי 2.

כתבו שיטה שחתימתה:

public int binarySearch(int[] a, int num)

השיטה מקבלת כפרמטר מערך **ממויין בסדר עולה** ומספר num. אם המספר קיים במערך, השיטה תחזיר את האינדקס שלו. אם המספר אינו קיים, השיטה תחזיר 1-. השיטה תבצע את החיפוש לפי אלגוריתם החיפוש הבינארי המתואר לעיל.
הדרכה – שימו לב שאין אפשרות פרקטית להקטין את המערך באופן פיסי. ולכן רצוי לשמור שני אינדקסים שיסמנו את גבולות המערך. את החיפוש נבצע בין האינדקסים האלו, כאשר בכל איטרציה אחד מהאינדקסים מתקדם לקראת השני.
ג. **חיפוש טרינארי** במערך ממויין בסדר עולה הוא אלגוריתם לחיפוש מספר במערך, שפועל כמו חיפוש בינארי. אולם במקום לחלק את המערך לשני חלקים, האלגוריתם מחלק את המערך לשלושה חלקים. כמו בחיפוש בינארי, אנו יודעים איפה להמשיך לחפש את המספר לפי השוואה למספר שנמצא בשליש המערך הראשון, והשוואה למספר שנמצא בשליש המערך השני.

8	20
7	18
6	15
5	11
4	10
3	8
2	7
1	5
0	3

נניח שבמערך זה נרצה לחפש את המספר 10. נחלק את המערך לשלושה חלקים. נשווה את המספר 10 למספר שנמצא בסוף השליש הראשון – 7. כיוון ש-10 גדול ממנו, נשווה את 10 למספר שבסוף השליש השני – 11. כיוון ש-10 קטן ממנו, אנו יודעים ש-10 אמור להמצא בשליש השני. כעת נחלק גם אותו לשני שלישים ונמשיך באותה דרך.

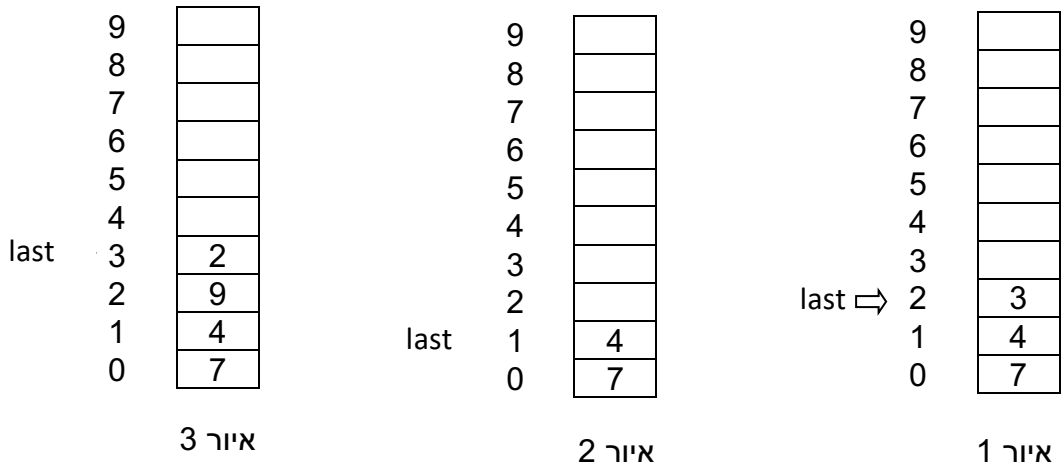
כתבו שיטה שחתימתה:

```
public int trenarySearch(int[] a, int num)
```

השיטה מקבלת כפרמטר מערך **ממויין בסדר עולה** ומספר num. אם המספר קיים במערך, השיטה תחזיר את האינדקס שלו. אם המספר אינו קיים, השיטה תחזיר 1-. השיטה תבצע את החיפוש לפי אלגוריתם החיפוש הטרינארי המתואר לעיל.

שאלה 22

מחסנית (stack) היא מבנה נתונים המאפשר לשמור ולהוציא נתונים בשיטת LIFO – Last In First Out כלומר הנתונים נכנסים למחסנית אחד אחרי השני, ויוצאים בסדר הפוך.
נוכל לייצג מחסנית של מספרים שלמים ע"י מערך חד ממדי של int. כתבו מחלקה בשם Stack שתייצג מחסנית.
המחלקה תכיל את התכונות הבאות:
elements – מסוג int[], מערך שיכיל את איברי המחסנית.
last – מסוג int, משתנה שישמור את מיקום האינדקס של האיבר האחרון במחסנית.
המחלקה תכיל את הבנאי:
public Stack(int size) – הבנאי יקבל מספר שלם, ויאתחל את המערך לגודל size ואת last ל 1-.
המחלקה תכיל את השיטות הבאות:
public boolean push(int x) – השיטה תקבל מספר שלם ותוסיף אותו למחסנית. אם המחסנית לא מלאה עדיין השיטה תוסיף את x למקום הפנוי הבא, ותחזיר true. אם המחסנית מלאה, השיטה לא תוסיף את המספר ותחזיר false.
public int pop() – השיטה תוציא את האיבר האחרון מהמחסנית ותחזיר אותו. אם המחסנית ריקה השיטה תעוף על שגיאת ריצה.
public boolean isEmpty() – השיטה תחזיר true אם המחסנית ריקה ו-false אם לא.
למשל, איור 1 מראה את מחסנית בגודל 10 לאחר הכנסת המספרים 7 ואחריו 4 ואחריו 3 (שימו לב למיקום המשתנה last). איור 2 מראה את המחסנית לאחר ביצוע פעולת pop אחת. איור 3 מראה את המחסנית לאחר הוספת המספרים 9 ואחריו 2.



שאלה 22

נתונה התכנית הבאה:

```
public class Tester
{
    public static void main(String[] args)
    {
        int[][] a = new int[4][5];
        // ...
    }
}
```

האיור הבא מציג את המערך שנוצר:

	0	1	2	3	4
0					
1					
2					
3					

כתבו שורות קוד מתאימות עבור כל אחד מהסעיפים הבאים. רשמו את התשובות בתכנית הנתונה, אחרי השורה המסומנת בהערה.

א. כתבו שורת קוד המציבה בתא הנמצא בשורה הראשונה ובעמודה הראשונה את המספר 2.

ב. כתבו שורת קוד המציבה בתא הנמצא בשורה האחרונה ובעמודה האחרונה את המספר 5.

ג. כתבו לולאה שמציבה בכל איברי השורה השניה את המספר 3.

ד. כתבו לולאה שמציבה בכל איברי העמודה השלישית את המספר 7.

שאלה 23

הוסיפו למחלקה Tester משאלה 22 קוד שיוצר מערך דו ממדי של int בעל 3 שורות ו-4 עמודות.
הוסיפו קוד שקולט מהמשתמש נתונים למערך זה, ומציב אותם לפי שורות – כלומר הנתון הראשון יכנס לאיבר הראשון בשורה הראשונה, הנתון השני יכנס לאיבר השני בשורה הראשונה וכן הלאה.

***

shay.tavor@gmail.com

כתבו מחלקה בשם TwoDArraysMethods. את התשובות לכל השאלות הבאות כתבו במחלקה זו.

שאלה 24

כתבו שיטה שחתימתה:

public void printArray(int[][] a)

השיטה תקבל כפרמטר מערך דו ממדי, ותדפיס את איבריו על המסך. כל שורה במערך תודפס כשורה על המסך, כאשר בין איבר לאיבר באותה שורה יפריד טאב.

שאלה 25

א. כתבו שיטה שחתימתה:

public int maxInArray(int[][] a)

השיטה תקבל כפרמטר מערך דו ממדי ותחזיר את האיבר המקסימלי במערך.

ב. כתבו שיטה שחתימתה:

public int minInArray(int[][] a)

השיטה תקבל כפרמטר מערך דו ממדי ותחזיר את האיבר המינימלי במערך.

שאלה 26

א. כתבו שיטה שחתימתה:

public int sumRow(int[][] a, int row)

השיטה תקבל כפרמטר מערך דו ממדי ואינדקס של שורה. השיטה תחזיר את סכום האיברים הנמצאים בשורה row.

ב. כתבו שיטה שחתימתה:

public int sumCol(int[][] a, int col)

השיטה תקבל כפרמטר מערך דו ממדי ואינדקס של עמודה. השיטה תחזיר את סכום האיברים הנמצאים בעמודה col.

שאלה 27

כתבו שיטה שחתימתה:

public boolean allPositive(double[][] arr)

השיטה תקבל כפרמטר מערך דו ממדי, ותחזיר true אם המערך מכיל אך ורק מספרים חיוביים, ו- false אחרת.

שאלה 28

א. כתבו שיטה שחתימתה:

public boolean isRowSorted(int[][] a. int row)

השיטה תקבל מערך דו ממדי ואינדקס של שורה. השיטה תחזיר true אם השורה מספר row ממויינת בסדר עולה ממש, כלומר שכל מספר בשורה גדול ולא שווה מהמספר שלפניו בשורה.

ב. כתבו שיטה שחתימתה:

public boolean allRowsSorted(int[][] a)

השיטה תקבל מערך דו ממדי ותחזיר true אם כל אחת מהשורות במערך ממויינת בסדר עולה ממש.

שאלה 29

כתבו שיטה שחתימתה:

public boolean special(int[][] a)

השיטה תקבל כפרמטר מערך דו ממדי, ותחזיר true אם כל האיברים בשורה ה-i קטנים ממש מכל האיברים בשורה ה-i+1. כלומר, כל האיברים בשורה ה-0 קטנים ממש מכל האיברים בשורה ה-1. כל האיברים בשורה ה-1 קטנים ממש מכל האיברים בשורה ה-2 וכן הלאה.
במושג "קטנים ממש" הכוונה היא שהאיברים לא שווים אחד לשני.

שאלה 30

נגדיר – מערך דו ממדי ריבועי יקרא **מערך משופע** אם הוא מקיים את התנאים הבאים:

1. כל איברי האלכסון הראשי שווים ל-0.
2. כל האיברים שמעל לאלכסון הראשי גדולים מ-0.
3. כל האיברים שמתחת לאלכסון הראשי קטנים מ-0.

	0	1	2	3	4
0	0	7	5	9	3
1	-1	0	4	7	6
2	-3	-2	0	2	3
3	-7	-9	-2	0	4
4	-1	-8	-5	-8	0

	0	1	2	3	4
0	0	7	5	9	3
1	-1	0	4	7	6
2	-3	-2	0	2	3
3	-7	-9	-2	0	4
4	-1	-8	-5	-8	0

המערך מצד ימין הוא מערך שיפועי, והמערך השמאלי אינו שיפועי.
כתבו שיטה שחתימתה:

public boolean isSloping(int[][] a)

השיטה תקבל כפרמטר מערך דו ממדי ריבועי ותחזיר true אם המערך הוא שיפועי לפי ההגדרה שלעיל, ו-false אם לא.

שאלה 31

כתבו שיטה שחתימתה:

public int sumDiagonal(int[][] a)

השיטה תקבל כפרמטר מערך דו ממדי ריבועי ותחזיר את סכום איברי האלכסון הראשי שלו. למשל, עבור המערך הבא:

	0	1	2	3
2	3	4	8	7
1	-3	6	-8	0
2	15	-4	-5	3
3	1	2	3	4

השיטה תחזיר 8. איברי האלכסון הראשי צבועים בכחול.

שאלה 32

א. כתבו שיטה שחתימתה:

public void switchRows(int[][] a, int r1, int r2)

השיטה תקבל כפרמטרים מערך דו ממדי, ואינדקסים של שתי שורות, r1, r2. השיטה תחליף בין האיברים של שתי השורות. כלומר כל איברי השורה r1 יהיו בשורה r2 ואיברי השורה r2 יהיו בשורה r1.

ב. כתבו שיטה שחתימתה:

public void switchCols(int[][] a, int c1, int c2)

השיטה תקבל כפרמטרים מערך דו ממדי, ואינדקסים של שתי עמודות c1, c2. השיטה תחליף בין האיברים של שתי העמודות. כלומר כל איברי העמודה c1 יהיו בעמודה c2 ואיברי העמודה c2 יהיו בעמודה c1.

שאלה 33

בהינתן מערך דו ממדי ריבועי, נגדיר **רצועה i** במערך ככל האיברים שנמצאים בשורות 0 עד i ובעמודה i, וכן כל האיברים שנמצאים בשורה i ובעמודות 0 עד i.

באיור הבא מוצג מערך דו ממדי ריבועי בגודל 4X4, כאשר הרצועות השונות צבועות בצבעים שונים.

	0	1	2	3
2	-3	4	8	11
1	-1	6	9	12
2	10	8	7	31
3	11	21	13	14

רצועה 0 צבועה בתכלת, רצועה 1 צבועה בירוק, רצועה 2 צבועה בכתום ורצועה 3 צבועה בלבן.

א. כתבו שיטה שחתימתה:

public int stripeNo(int[][] a, int x)

השיטה תקבל כפרמטרים מערך דו ממדי ריבועי ומספר x. השיטה תחזיר את מספר הרצועה בה נמצא x, או 1- אם x לא נמצא כלל במערך. למשל, עבור המערך הנתון לעיל והמספר 8 השיטה תחזיר 2 שכן 8 נמצא ברצועה 2. עבור המספר 14 השיטה תחזיר 3. עבור המספר 2 השיטה תחזיר 1- כיוון ש-2 אינו נמצא כלל במערך.

ב. נגדיר – מערך דו ממדי ריבועי יקרה **ממויין רצועות** אם עבור כל i בין 1 ל-a.length-1 כל האיברים ברצועה i גדולים ממש (כלומר, גדולים ולא שווים) מכל האיברים ברצועות 0 עד i-1. למשל, המערך הנתון לעיל הוא ממויין רצועות – אפשר לראות שכל האיברים ברצועה 1 גדולים ממש מכל האיברים ברצועה 0. כל האיברים ברצועה 2 גדולים ממש מכל האיברים ברצועה 1 וכן הלאה.

כתבו שיטה שחתימתה:

public boolean isStripesSorted(int[][] a)

השיטה תקבל כפרמטר מערך דו ממדי ריבועי ותחזיר true אם המערך הוא ממויין רצועות כמתואר לעיל, ו-false אם לא.

שאלה 34

בהנתן מערך דו ממדי שכל איבריו הם 0 או 1, נגדיר **בור** במערך כתא בשורה i ועמודה j כך שכל האיברים בשורה ה-i שווים ל-1 (חוץ מהאיבר ה i,j) וכל האיברים בעמודה ה-j שווים לאחד (חוץ מהאיבר ה-i, j).

לדוגמא, נתונים המערכים הבאים. במערך הימני קיים בור בתא [1,2] (מסומן בתכלת) ובמערך השמאלי לא קיים.

	0	1	2	3	4
0	0	1	1	0	0
1	0	1	0	0	1
2	0	0	0	0	0
3	0	1	0	1	1

	0	1	2	3	4
0	0	0	1	0	0
1	1	1	0	1	1
2	0	0	1	0	0
3	0	1	1	0	1

כתבו שיטה שחתימתה:

public boolean isSinkExist(int[][] a)

השיטה תקבל כפרמטר מערך דו ממדי המכיל אפסים ואחדים, ותחזיר true אם קיים בור במערך ו-false אם לא קיים.

שאלה 35

בהנתן מערך בעל N שורות ו-M עמודות, נגדיר **תת מערך תחתון i,j** כמרובע שכולל את כל התאים בין השורות i ו-N-1 ובין העמודות j ו-M-1. לדוגמא, באיור הבא מוגדר מערך בגודל 3X4 ובתכלת צבוע תת המערך התחתון 2, 3.

נגדיר **תת מערך עליון i, j** כמרובע שכולל את כל התאים בין השורות 0 ו-i-1 ובין העמודות 0 ו-j-1.

באיור הבא צבוע בירוק תת המערך העליון 2, 3.

	0	1	2	3	4
0	7	0	-1	6	2
1	2	11	4	-5	1
2	3	-9	1	8	10
3	6	5	2	7	4

א. כתבו שיטה שחתימתה:

public int sumLowerSub(int[][] a, int row, int col)

השיטה תקבל כפרמטרים מערך דו ממדי, ואינדקסים של שורה ועמודה. השיטה תחזיר את סכום תת המערך התחתון row, col במערך. למשל, עבור המערך שלעיל, ועבור row=2, col=3 השיטה תחזיר את הסכום 29.

ב. כתבו שיטה שחתימתה:

public int sumUpperSub(int[][] a, int row, int col)

השיטה תקבל כפרמטרים מערך דו ממדי, ואינדקסים של שורה ועמודה. השיטה תחזיר את סכום תת המערך העליון row, col במערך. למשל, עבור המערך שלעיל, ועבור row=2, col=3 השיטה תחזיר את הסכום 23.

ג. כתבו שיטה שחתימתה:

public boolean isEqualSubArrays(int[][] a)

השיטה תקבל כפרמטר מערך דו ממדי, ותחזיר true אם קיימים שני אינדקסים row, col כך שסכום תת המערך התחתון row, col שווה לסכום תת המערך העליון row, col. אם לא קיימים אינדקסים כאלה במערך, השיטה תחזיר false.

שאלה 36

כתבו שיטה שחתימתה:

public boolean allExist(int[][] a)

השיטה תקבל כפרמטר מערך דו ממדי בעל N שורות ו-M עמודות ותחזיר true אם כל המספרים בתחום 1 עד NXM קיימים במערך, ו-false אם לא.

לדוגמא, נתונים המערכים הבאים בגודל 3X4. ניתן לראות שבמערך הימני כל המספרים בין 1 ל-12 נמצאים במערך, ולכן השיטה תחזיר עליו true. לעומת זאת, במערך השמאלי ניתן לראות שלא כל המספרים בין 1 ל-12 נמצאים במערך (המספר 3 חסר) ולכן השיטה תחזיר עליו false.

	0	1	2	3
0	6	1	2	10
1	12	11	7	6
2	4	5	8	9

	0	1	2	3
0	2	1	10	6
1	11	12	8	5
2	3	4	9	7

השאלות הבאות עוסקות במערכים של אובייקטים

שאלה 37

נתונה המחלקה הבאה:

```
public class A
{
    private int x;
    public A(int x)
    {
        this.x = x;
    }
    public A(A other)
    {
        this.x = other.x;
    }
    public String toString()
    {
        return "X = " + x;
    }
    public boolean equals(A other)
    {
        return x == other.x;
    }
}
```

ונתונה התוכנית הבאה, העושה שימוש במחלקה A.

א. התייחסו לכל שורה בתוכנית והסבירו – מה עושה שורת הקוד? מה הפלט (אם יש)? שימו לב שהשורות קשורות זו לזו, התייחסו לכך.

```
public class Tester
{
    public static void main(String[] args)
    {
        A[] arr = new A[3];
        arr[0] = new A(5);
        arr[1] = new A(7);
        arr[2] = new A(arr[1]);
        for(int i = 0; i < arr.length; i++)
            System.out.println(arr[i]);
        System.out.println(arr[0].equals(arr[1]));
        A a1 = new A(5);
        System.out.println(a1.equals(arr[0]));
        a1 = arr[2];
    }
}
```

ב. מה יקרה אם בסוף התוכנית נוסיף את הפקודה:

arr = null;

shay.tavor@gmail.com

שאלה 38

נתונה המחלקה הבאה המייצגת ספר:

```
public class Book
{
    private String name, author;
    private int pages;

    public Book(String n, String a, int p)
    {
        name = n;
        author = a;
        pages = p;
    }
    public Book(Book other)
    {
        name = other.name;
        author = other.author;
        pages = other.pages;
    }
    public String getName() { return name; }
    public String getAuthor() { return author; }
    public int getPages() { return pages; }
    public String toString()
    {
        return name + ", by " + author + ", " + pages;
    }
}
```

לכל ספר יש תכונות שמייצגות את שמו, שם הסופר ומספר העמודים. במחלקה מוגדר בנאי המקבל את שלושת התכונות ומאתחל אותן, וכן בנאי העתקה. כמו כן מוגדרות שיטות get עבור כל אחת מהתכונות, ושיטת toString.

נתונה המחלקה Library שמייצגת ספריה של ספרים. המחלקה שומרת את הספרים במערך של אובייקטים מסוג Book. גודל המערך המירבי הוא 100. כמו כן המחלקה שומרת תכונה בשם numOfBooks שמייצגת את כמות הספרים הנוכחית במערך. השלימו את השיטות במחלקה. תיאור השיטות נמצא בסעיפים הבאים:

א. השיטה addBook תקבל אובייקט מסוג Book כפרמטר, ותוסיף **עותק** שלו לספריה, במקום הפנוי הבא במערך. אם אין מקום במערך, השיטה תחזיר false. אם הספר התווסף בהצלחה, השיטה תחזיר true.

ב. השיטה getByBookName תקבל כפרמטר מחרוזת המייצגת שם של ספר. השיטה תחזיר את אובייקט הספר שנמצא במערך ויש לו את אותו השם (השיטה תחזיר עותק של האובייקט). אם לא קיים במערך ספר בעל שם זה, השיטה תחזיר null.

ג. השיטה countAuthor תקבל כפרמטר מחרוזת המייצגת שם של סופר. השיטה תחזיר כמה ספרים בספריה נכתבו ע"י סופר זה.

ד. השיטה maxPagesBook תחזיר עותק של אובייקט Book מהספריה המכיל את מספר העמודים הרב ביותר.

ה. השיטה toString תחזיר מחרוזת המייצגת את הספריה. בתחילה תופיע המחרוזת:
Total books in library: [numOfBooks]
ואחריה פירוט הספרים, כל ספר בשורה נפרדת.

```java
public class Library
{
    private Book[] books;
    private int numOfBooks;
    private final int MAX_BOOKS = 100;

    public Library()
    {
        books = new Book[MAX_BOOKS];
        numOfBooks = 0;
    }

    public int getNumOfBooks() { return numOfBooks; }

    public boolean addBook(Book b)
    {
        // ...
    }
    public Book getByBookName(String bName)
    {
        // ...
    }
    public int countAuthor(String author)
    {
        // ...
    }
    public Book maxPagesBook()
    {
        // ...
    }
    public String toString()
    {
        // ...
    }
}
```

שאלה 39

נתונה המחלקה Student המייצגת סטודנט (בעמוד הבא):
לכל סטודנט יש תכונות של מספר ת.ז., שם, גיל ושם המחלקה בה הוא לומד. המחלקה מכילה בנאי המקבל את התכונות ומציב אותן, ובנאי העתקה. כמו כן, המחלקה מכילה שיטות get עבור כל אחת מהתכונות, ושיטת toString שמחזירה את פרטי הסטודנט בפורמט של מחרוזת.

```java
public class Student
{
    private String name, department;
    private int id, age;

    public Student(int id, String name, String department, int age)
    {
        this.id = id;
        this.name = name;
        this.department = department;
        this.age = age;
    }
    public Student(Student other)
    {
        other.id = id;
        name = other.name;
        department = other.department;
        age = other.age;
    }

    public int getId() { return id; }
    public String getName() { return name; }
    public String getDepartment() { return department; }
    public int getAge() { return age; }

    public String toString()
    {
        return "Student " + id + ", " + name +
                ", age: " + age + ", department: " + department;
    }
}
```

כתבו מחלקה בשם University שתייצג אוניברסיטה. המחלקה תכיל את התכונות הבאות:
students – תכונה מסוג Student[] שתכיל את הסטודנטים באוניברסיטה.
numOfStudents – תכונה מסוג int שתכיל את מספר הסטודנטים בפועל במערך.
הגדירו קבוע בשם MAX_STUDENTS שיהיה שווה ל-1000 וייצג את גודל המערך המירבי (כלומר, את מספר הסטודנטים המקסימלי).
הוסיפו למחלקה בנאי ריק. הבנאי יאתחל את המערך לגודל MAX_STUDENTS ויציב 0 בתכונה numOfStudents.
הוסיפו למחלקה את השיטות הבאות:
public int getNumOfStudents() – השיטה תחזיר את מספר הסטודנטים הנוכחי במערך.
public String toString() – השיטה תחזיר את פרטי הסטודנטים בצורת מחרוזת, כל סטודנט בשורה נפרדת.
public boolean addStudent(Student s) – השיטה תקבל כפרמטר אובייקט מסוג Student ותוסיף עותק שלו למערך. אם ההוספה הסתיימה בהצלחה, השיטה תחזיר true. אם אין מקום במערך לסטודנט נוסף, השיטה תחזיר false.
public Student find(int id) – השיטה תקבל כפרמטר מספר ת.ז. של סטודנט ותחזיר עותק של הסטודנט בעל מספר ת.ז. זה. אם לא קיים סטודנט כזה השיטה תחזיר null.
public Student oldestStudent() – השיטה תחזיר עותק של הסטודנט המבוגר ביותר במערך.
public int howManyFromDepartment(String dep) – השיטה תקבל כפרמטר של של מחלקת לימודים, ותחזיר את מספר הסטודנטים באוניברסיטה שלומדים במחלקה זו.
public Student[] sameName(String name) – השיטה תקבל כפרמטר שם של סטודנט, ותחזיר מערך מסוג Student המכיל עותקים של כל הסטודנטים בעלי שם זה. אם אין אף סטודנט בעל שם זה, השיטה תחזיר null.

שאלה 40

א. כתבו מחלקה בשם Date שתייצג תאריך. התאריך ייוצג ע"י שלוש תכונות – יום, חודש ושנה (כל אחת מהתכונות מסוג int). (המחלקה Date ניתנה כתרגיל בקובץ "אובייקטים ומחלקות חלק ב'" והיא מובאת פה למען הנוחות).
הוסיפו למחלקה את הבנאים הבאים:
public Date(int day, int month, int year) – הבנאי יקבל כפרמטרים את ערכי היום, חודש והשנה, ויציב אותם לתכונות. אם אחד מהפרמטרים אינו חוקי, הבנאי יציב במקומו 1.
public Date(Date other) – בנאי העתקה.
הוסיפו למחלקה שיטת get/set עבור כל אחת מהתכונות. שימו לב שבשיטות ה-set אסור לאפשר הכנסה של ערכים לא חוקיים. אם המשתמש ינסה להכניס, למשל, יום שערכו 89, השיטה לא תשנה את ערך היום.
הוסיפו למחלקה את השיטות הבאות:
public String toString() – השיטה תחזיר את פרטי התאריך כמחרוזת בפורמט הבא: אם למשל התאריך הוא 2 לינואר 2011, השיטה תחזיר את המחרוזת:
2/1/2011
public boolean equals(Date d) – השיטה תקבל כפרמטר אובייקט מסוג Date ותחזיר true אם הוא שווה ל-this ו-false אחרת.
public boolean before(Date d) – השיטה תקבל כפרמטר אובייקט מסוג Date ותחזיר true אם התאריך המיוצג ע"י this הוא **לפני (כרונולוגית)** התאריך המיוצג ע"י d. למשל, אם האובייקט d מכיל את התאריך 2/4/2011 והאובייקט this מכיל את התאריך 1/1/2011 השיטה תחזיר true.
public boolean after(Date d) – השיטה תקבל כפרמטר אובייקט מסוג Date ותחזיר true אם האובייקט המיוצג ע"י this הוא **אחרי (כרונולוגית)** לתאריך המיוצג ע"י d.

ב. כתבו מחלקה בשם Worker שתייצג עובד במפעל. המחלקה תכיל את התכונות הבאות:
name – שם העובד, מסוג String.
salary – משכורת העובד, מסוג double.
hireDate – תאריך תחילת העבודה, מסוג Date.
הוסיפו למחלקה את הבנאים הבאים:
public Worker(String name, double salary, Date hire) – הבנאי יקבל את שם העובד, משכורתו ואובייקט שמייצג את תאריך תחילת העבודה, ויאתחל אותם בתכונות.
public Worker(Worker other) – בנאי העתקה.
הוסיפו שיטות get עבור כל אחת מהתכונות (המנעו מ-aliasing בשיטת getHireDate).
הוסיפו למחלקה את השיטות הבאות:
public String toString() – השיטה תחזיר את פרטי העובד כמחרוזת, בפורמט הבא. אם למשל שם העובד הוא יוסי, משכורתו 5000 והוא התחיל לעבוד ב- 1/1/2011 השיטה תחזיר את המחרוזת:
Yossi, 5000 nis, starts at 1/1/2011
public boolean earnsMoreThan(Worker other) – השיטה תקבל אובייקט מסוג Worker ותחזיר true אם העובד שמיוצג ע"י this מרוויח יותר מהעובד המיוצג ע"י other , ו-false אם לא.

ג. כתבו מחלקה בשם Factory שתייצג מפעל. המחלקה תכיל את התכונות הבאות:
workers – מערך של העובדים במפעל, מסוג []Worker.
numOfWorkers – מספר העובדים בפועל במפעל, מסוג int.
הוסיפו למחלקה קבוע בשם MAX_WORKERS שיהיה שווה ל-100 וייצג את מספר העובדים המקסימלי שיכולים להיות במפעל.
הוסיפו למחלקה בנאי ריק שיאתחל את המערך לגודל MAX_WORKERS ואת מספר העובדים לאפס.
הוסיפו למחלקה את השיטות הבאות:
public String toString() – השיטה תחזיר מחרוזת המכילה את פרטי העובדים במפעל, כל עובד בשורה חדשה.

public boolean addWorker(Worker w) – השיטה תקבל אובייקט מסוג Worker ותוסיף אותו למערך העובדים במקום הפנוי הבא. אם ההוספה הצליחה, השיטה תחזיר true. אם אין מקום במערך לעובד נוסף, השיטה תחזיר false.
public Worker minSalary() – השיטה תחזיר עותק של אובייקט העובד שמרוויח את המשכורת הנמוכה ביותר.
public Date getWorkerDate(String name) – השיטה תקבל כפרמטר שם של עובד, ותחזיר עותק של אובייקט תאריך תחילת עבודתו. אם עובד בשם זה אינו קיים במערך, השיטה תחזיר null.
public String firstHired() – השיטה תחזיר את שמו של העובד שנשכר הכי מוקדם לעבודה (כלומר, תאריך תחילת העבודה שלו הוא המוקדם ביותר מכל העובדים).

שאלה 41

א. כתבו מחלקה בשם Course שתייצג קורס באוניברסיטה. המחלקה תכיל את התכונות הבאות:
name – שם הקורס, מסוג String.
day – היום בו נלמד הקורס, מסוג int, כאשר 1 מייצג את יום ראשון, ו-5 את יום חמישי.
beginHour – שעת תחילת הקורס, מסוג int. ניתן להניח שהקורסים נלמדים בין השעות 8:00 בבוקר ל-20:00 בערב, ושכל קורס נמשך שעתיים בדיוק.
הוסיפו למחלקה את הבנאים הבאים:
public Course(String name, int day, int beginHour) – הבנאי יקבל את ערכי התכונות כפרמטרים ויציב אותם לתכונות. אם היום אינו חוקי, הבנאי יציב בתכונת היום 1, ואם השעה אינה חוקית, הבנאי יציב בתכונת השעה 8.
public Course(Course other) – בנאי העתקה.
הוסיפו שיטות get עבור כל אחת מהתכונות.
הוסיפו למחלקה את השיטות הבאות:
public String toString() – השיטה תחזיר את פרטי הקורס בפורמט של מחרוזת. הפרטים יופרדו ביניהם בפסיקים.

ב. כתבו מחלקה בשם Schedule שתייצג את מערכת השעות של הקורסים. המחלקה תכיל את התכונות:
courses – מערך שיכיל את הקורסים, מסוג []Course.
numOfCourses – תכונה שתצייג את מספר הקורסים בפועל, מסוג int.
הוסיפו למחלקה קבוע בשם MAX_COURSES שיהיה שווה ל-20 ויציין את מספר הקורסים המקסימלי שיש במערכת.
הוסיפו בנאי ריק למחלקה, שיאתחל את המערך לגודל MAX_COURSES ויציב אפס במספר הקורסים.
הוסיפו למחלקה את השיטות הבאות:
public boolean addCourse(Course c) – השיטה תקבל אובייקט מסוג Course ותוסיף עותק שלו למערך הקורסים. אם ההוספה הצליחה, השיטה תחזיר true. אם אין מקום במערך לקורס נוסף, השיטה תחזיר false.
public String toString() – השיטה תחזיר מחרוזת המכילה את פרטי הקורסים במערכת, כל קורס בשורה נפרדת.
public Course getFirstInDay(int day) – השיטה תקבל מספר יום (בין 1 ל-5) ותחזיר עותק של אובייקט הקורס שנלמד בשעה המוקדמת ביותר באותו היום. אם באותו היום לא נלמד אף קורס, השיטה תחזיר null.
public Course getNextInDay(Course c) – השיטה תקבל אובייקט מסוג Course. השיטה תחזיר עותק של אובייקט Course שהוא הראשון שנלמד אחרי c באותו היום. למשל, אם הקורס c נלמד ביום ראשון, בשעה 12, אזי השיטה צריכה להחזיר את אובייקט הקורס שנלמד גם הוא ביום ראשון,

בשעה הקרובה ביותר ל-12. כלומר, אם יש קורס אחד שנלמד בשעה 14 וקורס שנלמד בשעה 16, השיטה תחזיר את הקורס של השעה 14. אם הקורס c הוא האחרון ביום, השיטה תחזיר null.
public String allCoursesInDay(int day) – השיטה תקבל מספר יום (בין 1 ל-5) ותחזיר מחרוזת שמייצגת את פרטי כל הקורסים שנלמדים באותו היום. אם באותו היום לא נלמד אף קורס, השיטה תחזיר null.
public void removeCourse(String cName) – השיטה תקבל כפרמטר שם של קורס, ותסיר מהמערך את אובייקט הקורס בעל שם זה. אם לא קיים קורס בעל שם זה, השיטה לא תעשה דבר.
שימו לב – מערך הקורסים חייב להמשיך להכיל קורסים ברצף, גם אחרי ההוספה.

שאלה 42

א. כתבו מחלקה בשם Apartment המייצגת דירה. למחלקה יהיו התכונות הבאות:
familyName – שם המשפחה שגרה בדירה, מסוג String. אם הדירה ריקה, כלומר לא גרה בה אף משפחה, התכונה תהיה שווה למחרוזת ריקה "".
rent – שכר הדירה. מסוג int.
isRented – תכונה מסוג boolean שמסמנת האם הדירה מושכרת (שווה ל-true) או לא (שווה ל-false).
הוסיפו למחלקה את הבנאים הבאים:
public Apartment(int rent) – הבנאי יקבל את שכר הדירה ויציב אותו בתכונה. הבנאי יציב false בתכונה הבוליאנית, ומחרוזת ריקה בשם המשפחה.
public Apartment(String familyName, int rent) – הבנאי יקבל כפרמטרים את שם המשפחה שגרה בדירה ואת שכר הדירה ויציב אותם בתכונות. הבנאי יציב true בתכונה הבוליאנית.
public Apartment(Apartment other) – בנאי העתקה.
הוסיפו שיטות get/set עבור כל אחת מהתכונות.
הוסיפו את השיטות הבאות למחלקה:
public String toString() – השיטה תחזיר את פרטי הדירה כמחרוזת בפורמט הבא: למשל, אם קיימת דירה ששכר הדירה בה הוא 4000 שקל והיא לא מושכרת, השיטה תחזיר את המחרוזת:
Apartment rent: 4000, not rented yet.
ואם הדירה מושכרת למשפחת כהן השיטה תחזיר את המחרוזת:
Apartment rent: 4000, rented to Cohen.

ב. כתבו מחלקה בשם Building שתייצג בניין דירות. המחלקה תכיל את התכונות הבאות:
apartments – מערך מסוג []Apartment שישמור את הדירות בבניין.
numOfApartments – משתנה מסוג int שישמור את מספר הדירות שיש בפועל במערך.
הוסיפו למחלקה קבוע בשם MAX_APARTMENTS שיהיה שווה ל-20 ויציין את מספר הדירות המקסימלי שיכולות להיות בבניין.
הוסיפו למחלקה בנאי ריק שיאתחל את המערך ויציב 0 במספר הדירות.
הוסיפו למחלקה את השיטות הבאות:
public boolean addApartment(Apartment a) – השיטה תקבל כפרמטר אובייקט מסוג Apartment ותוסיף אותו למערך. אם יש מקום במערך לדירה נוספת, השיטה תוסיף את האובייקט ותחזיר true. אחרת השיטה לא תשנה את המערך ותחזיר false.
public void raiseRent(int amount) – השיטה תקבל כפרמטר סכום כסף, ותעלה את שכר הדירה של כל הדירות בבניין בסכום זה.
public int howManyNotRented() – השיטה תחזיר את מספר הדירות בבניין שלא מושכרות.
public int howManyPays(String familyName) – השיטה תקבל כפרמטר שם משפחה, ותחזיר את הסכום שמשלמת משפחה זו כשכר דירה. אם משפחה זו אינה גרה בבניין, השיטה תחזיר 1-.
public Apartment maxRent() – השיטה תחזיר אובייקט מסוג Apartment שמייצג את הדירה ששכר הדירה שלה הוא הגבוה ביותר בבניין. אם אין דירות בבניין השיטה תחזיר null.

public String toString() – השיטה תחזיר את פרטי הדירות בבניין בצורת מחרוזת בפורמט הבא:
בתחילת המחרוזת יופיע המשפט –
This building has [numOfApartments] apartments
כאשר במקום [numOfApartments] יופיע מספר הדירות שיש בפועל בבניין. לאחר משפט זה יופיעו פרטי כל הדירות, כל דירה בשורה נפרדת.

תווים ומחרוזות

קובץ זה מכיל תרגילים בנושא תווים ומחרוזות בשפת ג'אווה. חלק מהשאלות תאורטיות ורובן שאלות כתיבת קוד. השאלות מסודרות מהקלות לקשות. כדאי ורצוי לפתור את כל השאלות, להעביר אותן קומפילציה ולהריץ אותן, כדי לתרגל את אופן כתיבת הקוד.

שיטות חשובות של המחלקה *String*

המחלקה String בג'אווה מכילה לא מעט שיטות חשובות שמאפשרות לבצע מניפולציות מכל סוג על מחרוזות. ניתן לקרוא את התיעוד הפורמלי של כל השיטות באתר ה-API של ג'אווה, בכתובת:
http://download.oracle.com/javase/7/docs/api/
בצד שמאל ניתן לבחור את המחלקה String.
להלן מספר שיטות שימושיות ביותר של המחלקה:
public boolean equals(String s) – השיטה מקבלת כפרמטר אובייקט מסוג מחרוזת, ומשווה אותו למחרוזת המיוצגת ע"י this. אם שתי המחרוזות שוות (כלומר, אותם תווים בדיוק, באותו הסדר) השיטה מחזירה true, אחרת היא מחזירה false.
public int length() – השיטה מחזירה את אורך המחרוזת (מספר התווים שבה).
public char charAt(int i) – השיטה מקבלת מספר אינדקס במחרוזת, ומחזירה את התו שנמצא באינדקס זה. שימו לב שהתווים במחרוזת נספרים מ-0 ועד length()-1. אם האינדקס אינו חוקי, השיטה עפה על שגיאת ריצה.
public int indexOf(char c) – השיטה מקבלת תו, ומחזירה את האינדקס של המופע הראשון של התו במחרוזת. אם התו אינו קיים במחרוזת השיטה תחזיר 1-.
public String substring(int i) – השיטה מקבלת מספר אינדקס במחרוזת, ומחזירה מחרוזת חדשה שמורכבת מכל התווים במחרוזת המקורית שנמצאים מאינדקס i ועד סוף המחרוזת.
כמו כן, ניתן להשתמש בשיטות של האובייקט Scanner כדי לבצע קלט מחרוזת מהמשתמש:
next() – השיטה קולטת מהמשתמש מחרוזת רצופה של תווים, עד רווח או שורה חדשה.
nextLine() – השיטה קולטת מהמשתמש שורה של תווים, עד סוף השורה.

שאלה 1

נתונה התוכנית הבאה. עבור כל שורת הדפסה בתוכנית, כתבו מה הפלט הצפוי והסבירו מה משמעותו של הפלט.

```
public class StringsTester
{
    public static void main(String[] args)
    {
        String s1 = "Hello";
        String s2 = new String("Hello");
        System.out.println(s1 + " " + s2);
        System.out.println(s1.length());
        System.out.println(s1.charAt(0));
        System.out.println(s1.equals(s2));
        System.out.println(s2.indexOf('o'));
        System.out.println(s2.indexOf('T'));
        s2 = s2.substring(2);
        System.out.println(s2);
        for(int i = 0; i < s1.length(); i += 2)
            System.out.println(s1.charAt(i));
    }
}
```

***

כתבו מחלקה בשם StringsMethods. את התשובות לכל השאלות הבאות כתבו כשיטות של מחלקה זו.

שאלה 2

א. כתבו שיטה שחתימתה:

public boolean isDigit(char c)

השיטה תקבל תו, ותחזיר true אם התו מייצג ספרה (0 – 9) ו-false אם לא.

ב. כתבו שיטה שחתימתה:

public boolean isWhiteSpace(char c)

השיטה תקבל תו, ותחזיר true אם התו מייצג רווח, טאב ("\t") או שורה חדשה ("\n"), ו-false אם לא.

שאלה 3

כתבו שיטה שחתימתה:

public int howMany(String s, char c)

השיטה תקבל כפרמטרים מחרוזת s ותו c. השיטה תחזיר כמה פעמים מופיע התו c במחרוזת.
למשל, אם s = "abcadca", אזי, אם c = 'a' השיטה תחזיר 3. אם c = 'b' השיטה תחזיר 1. אם c = 'f' השיטה תחזיר 0.

שאלה 4

כתבו שיטה שחתימתה:

public boolean notExist(String s, char c)

השיטה תקבל כפרמטרים מחרוזת s ותו c. השיטה תחזיר true אם התו c לא נמצא כלל במחרוזת, ו-false אחרת. למשל, אם s = "abcd", אזי אם c = 'a' השיטה תחזיר false ואם c = 'f' השיטה תחזיר true.

שאלה 5

מחרוזת פאלינדרומית היא מחרוזת שניתן לקרוא אותה מההתחלה לסוף ומהסוף להתחלה באותה צורה. למשל, המחרוזות הבאות הן פאלינדרומיות: "a", "madam", "abba".
כתבו שיטה שחתימתה:

public boolean isPalindrom(String s)

השיטה תקבל כפרמטר מחרוזת, ותחזיר true אם המחרוזת היא פאלינדרומית, ו-false אם לא.

שאלה 6

כתבו שיטה שחתימתה:

public String removeNonLetters(String s)

השיטה תקבל כפרמטר מחרוזת, תסיר ממנה את כל התווים שהם אינם אותיות (גדולות או קטנות) ותחזיר את המחרוזת החדשה.
למשל, אם s = "a$Bf# c" השיטה תחזיר את המחרוזת "aBfc".
הדרכה: שימו לב שאין אפשרות לשנות את המחרוזת הקיימת. יש לבנות מחרוזת חדשה ולהעתיק אליה את הנתונים הרלבנטים.

שאלה 7

כתבו שיטה שחתימתה:

public String reverse(String s)

השיטה תקבל כפרמטר מחרוזת, ותחזיר מחרוזת חדשה המכילה את אותם תווים בסדר הפוך.
למשל, אם s = "abcde" השיטה תחזיר את המחרוזת "edcba".

שאלה 8

שתי מחרוזות יקראו פרמוטציות אחת של השניה, אם הן מכילות את אותן תווים אבל בסדר שונה.
למשל, המחרוזות "abcde" ו-"acedb" הן פרמוטציות. לעומת זאת המחרוזות "abc" ו-"abb" אינן פרמוטציות.
כתבו שיטה שחתימתה:

```
public boolean arePermutation(String s, String t)
```

השיטה תקבל שתי מחרוזות, t ,s ותחזיר true אם הן פרמוטציות אחת של השניה ו-false אם לא.

שאלה 9

א. נגדיר, מחרוזת t היא תת מחרוזת של s אם כל התווים של t נמצאים ברצף ב-s.
למשל, אם s = "abcdef" אזי המחרוזות (בין היתר) "abc", "bcde", "f" הן תתי מחרוזות של s, אולם המחרוזת "acf" אינה תת מחרוזת.
כתבו שיטה שחתימתה:

```
public boolean isSubstring(String s, String t)
```

השיטה תקבל שתי מחרוזות, s, t ותחזיר true אם t היא תת מחרוזת של s, ו-false אם לא.

ב. כתבו שיטה שחתימתה:

```
public int howManySub(String s, String t)
```

השיטה תקבל שתי מחרוזות, s, t, ותחזיר כמה פעמים מופיעה המחרוזת t בתוך s.
למשל, אם s = "abcaabc" ו-t = "abc" השיטה תחזיר 2.

שאלה 10

כתבו שיטה שחתימתה:

```
public int howManySpecial(String s, char start, char end)
```

השיטה תקבל כפרמטרים מחרוזת s ושני תווים start ו-end. השיטה תחזיר כמה תתי מחרוזות שמתחילות ב-start ומסתיימות ב-end קיימות במחרוזת s.
למשל, אם s = "abcbagfb" ו- start = 'a', end = 'b' השיטה תחזיר 4.

שאלה 11

נגדיר שפה חדשה שבה מתקיימים החוקים הבאים:
1. האותיות היחידות הקיימות בשפה הן a, b, c.
2. כל מילה בשפה חייבת להתחיל באות a.
3. מספר ה-b בכל מילה הוא זוגי.
4. אחרי כל מופע של c חייבים לבוא שני a ברצף.

למשל, המילים הבאות חוקיות: "a", "abb", "abcaab", "acaaaa".
המילים הבאות אינן חוקיות: "bb", "ab", "aca", "abbd".
כתבו שיטה שחתימתה:

```
public boolean isLegal(String word)
```

השיטה תקבל כפרמטר מחרוזת, ותחזיר true אם המחרוזת מהווה מילה חוקית בשפה, לפי החוקים שלעיל, ו-false אם לא.

שאלה 12

כתבו שיטה שחתימתה:

public String[] split(String s, char delim)

השיטה תקבל כפרמטרים מחרוזת s ותו. השיטה תפריד את המחרוזת לתתי מחרוזות לפי התו delim ותחזיר מערך של תתי המחרוזות האלו.

למשל, אם s = "Hello world how do you do?" והתו delim = ' ' (רווח), השיטה תחזיר את המערך: {"Hello", "world", "how", "do", "you", "do?"}

שאלה 13

כתבו שיטה שחתימתה:

public String replace(String s, char origin, char target)

השיטה תקבל כפרמטרים מחרוזת s ושני תווים. השיטה תחזיר מחרוזת חדשה בה כל המופעים של התו origin יוחלפו במופעים של התו target.

למשל, אם s = "abcadga" ו- origin = 'a', target = '*' השיטה תחזיר את המחרוזת "*bc*dg*".

שאלה 14

כתבו שיטה שחתימתה:

public String compress(String s)

השיטה תקבל כפרמטר מחרוזת s. השיטה תחזיר מחרוזת חדשה שבה כל רצף תווים זהים במחרוזת s יוחלף בתו אחד בודד.

למשל, אם s = "aaaabccdddddd" השיטה תחזיר את המחרוזת "abcd".

ירושה ופולימורפיזם

קובץ תרגילים זה מכיל שאלות בנושא ירושה ופולימורפיזם בשפת ג'אווה. חשוב להבין את עקרון הירושה ולהבין את היחסים בין המחלקות השונות. כדאי מאוד לענות גם על השאלות התאורטיות בפירוט, כדי לוודא שאתם מבינים את המושגים.

שאלה 1

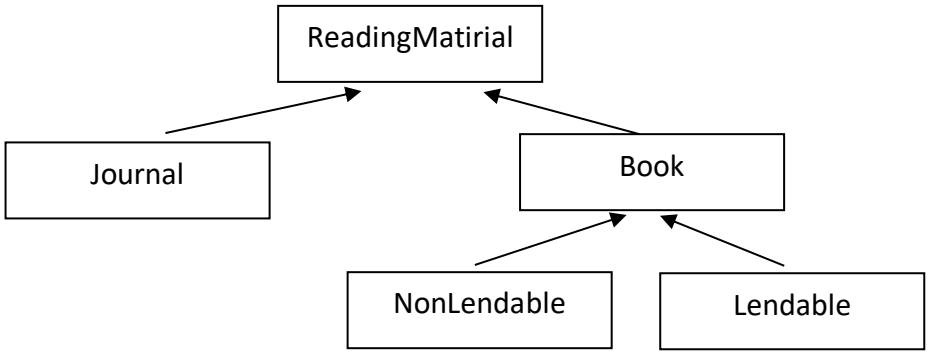
ענו על השאלות הבאות. השתדלו לנמק היטב ולהמנע מניסוחים מעורפלים.
א. מהי ירושה?
ב. מהו עץ ירושה?
ג. מהי המחלקה Object? מה נמצא במחלקה הזאת?
ד. איך ניתן להגדיר שמחלקה יורשת מ-Object?
ה. איך מגדירים שמחלקה A יורשת ממחלקה B?
ו. מהי ירושה מרובה?
ז. מהו מאפיין הגישה protected? מה ההבדל בינו לבין private ו-public?
ח. מהי המילה השמורה super? מה היא מייצגת?
ט. מהו סדר הפעלת הבנאים בירושה?
י. מה זה דריסה (overriding)? מה ההבדל בין דריסה לבין העמסה של שיטות (Overloading)?

שאלה 2

ברצונינו לתאר מחלקות המייצגות עיתונים שונים. נגדיר את המחלקה Newspaper המייצגת עיתון. בנוסף נגדיר שתי מחלקות נוספות היורשות ממנה – DailyNewspaper המייצגת עיתון המופץ בכל יום, ו-MonthlyNewspaper המייצגת ירחון (עיתון המופץ פעם בחודש).
א. ציירו את עץ הירושה עבור המחלקות.
ב. עבור התכונות הבאות, החליטו באיזו מחלקה צריכה להיות ממוקמת כל תכונה:
שם – שם העיתון.
כתובת ההוצאה – כתובת הוצאת העיתון.
שם עורך ראשי – שמו של העורך הראשי בעיתון.
תאריך חודשי – התאריך בחודש בו מופץ העיתון.

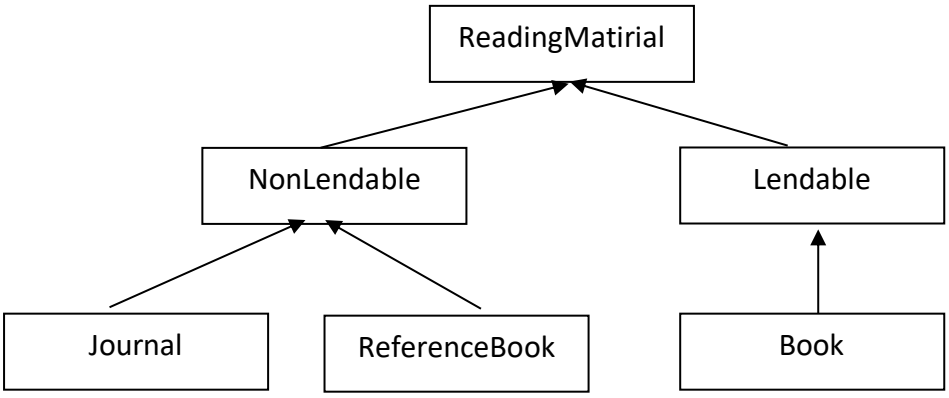
שאלה 3

בספריה מסויימת קיימים פריטים שונים. אנשי הספריה מעוניינים לייצג את הפריטים בעץ ירושה. הפריטים הקיימים בספריה – ספרים מסוגים שונים – ספרי עיון, אותם לא ניתן להשאיל אלא לעיין בהם בספריה, וספרים הניתנים להשאלה. כתבי עת אותם לא ניתן להשאיל.
שני מתכנתים שונים הציעו שני עצי ירושה שונים למערכת:



עץ ירושה א'

avor@gmail.com



עץ ירושה ב'

מהם היתרונות והחסרונות בכל אחד מעצי הירושה? התייחסו לתכונות שאמורות להיות בכל מחלקה.
אילו מעצי הירושה עדיף לדעתכם ליצוג המערכת?

שאלה 4

נתונה המחלקה הבאה המייצגת בן אדם (בעמוד הבא).
כתבו מחלקה בשם Lecturer המייצגת מרצה ויורשת מהמחלקה Person. לכל מרצה תהיה התכונה:
title - משתנה מסוג String המכיל את תואר המרצה – ד"ר או פרופ'.
הוסיפו למחלקה שיטות get/set עבור התכונה.
כמו כן דרסו במחלקה את השיטה getName שנורשת מ-Person. השיטה תחזיר מחרוזת המורכבת מתואר המרצה ומשמו. למשל, אם שם המרצה הוא דוד כהן, והוא פרופ', השיטה תחזיר את המחרוזת:

Prof' David Cohen

```
public class Person
{
    protected String name;

    public void setName(String n)
    {
        name = n;
    }
    public String getName()
    {
        return name;
    }
}
```

שאלה 5

בכל אחד מהסעיפים הבאים נתון קטע קוד שאינו עובר קומפילציה. הסבירו מדוע ומה השגיאה.

א.

```
public class A extends B
{
}
public class B extends A
{
}
```

ב.

```
public class A
{
}
public class B extends A, Object
{
}
```

ג.

```
public class A extends B
{
    private int x;
}
public class B extends A
{
    public void f()
    {
        x = 7;
    }
}
```

שאלה 6

נתונות המחלקות הבאות:

```
public class A
{
    public A(int x)
    {
        System.out.println("In A " + x);
    }
    public A()
    {
        System.out.println("In A ");
    }
}
public class B extends A
{
    public B(int x, int y)
    {
        System.out.println("In B " + x + ", " + y);
    }
    public B(int x)
    {
        super(x);
        System.out.println("In B " + x);
    }
}
public class C extends B
{
    public C()
    {
        super(1, 2);
        System.out.println("In C");
    }
    public C(int x)
    {
        super(x);
        System.out.println("In C " + x);
    }
}
```

ונתון ה-main הבא. עבור כל שורה ב-main רשמו מה יהיה הפלט:

```
public class Tester {
    public static void main(String[] args)
    {
        A a1 = new A();
        A a2 = new A(3);
        B b1 = new B(7, 8);
        B b2 = new B(5);
        C c1 = new C();
        C c2 = new C(6);
    }
}
```

שאלה 7

נתונות המחלקות הבאות:

```
public class A {
    public void f() {
        System.out.println("In A's f");
    }
    public void g() {
        System.out.println("In A's g");
    }
    public void m() {
        System.out.println("In A's m");
        f();
    }

}
public class B extends A {
    public void f() {
        System.out.println("In B's f");
    }
    public void g() {
        super.g();
        System.out.println("In B's g");
        f();
        super.f();
    }
}
```

ונתון ה-main הבא. עבור השורות המסומנות 1 עד 6 רשמו מה יהיה הפלט:

```
public class Tester
{
    public static void main(String[] args)
    {
        A a1 = new A();
        B b1 = new B();

        a1.f();      // 1
        a1.g();      // 2
        a1.m();      // 3
        b1.f();      // 4
        b1.g();      // 5
        b1.m();      // 6
    }
}
```

שאלה 8

א. כתבו מחלקה בשם BankAccount שתייצג חשבון בנק. המחלקה תכיל את התכונות הבאות:
num – מסוג int, מייצג את מספר החשבון.
balance – מסוג double, מייצג את היתרה בחשבון.
כמו כן הוסיפו למחלקה שני קבועים–
OVERDRAFT_INTEREST – מייצג את אחוז הריבית על משיכת יתר בחשבון, ומאותחל לערך 0.1 (כלומר, עשרה אחוז).
BALANCE_INTEREST – מייצג את אחוז הריבית על היתרה בחשבון, ומאותחל לערך 0.05 (כלומר, חמישה אחוז).
הוסיפו למחלקה את הבנאים:
public BankAccount(int num) – הבנאי יקבל את מספר החשבון, יציב אותו בתכונה ויציב 0 בתכונת היתרה.
public BankAccount(int num, double balance) – הבנאי יקבל את מספר החשבון, יציב אותו בתכונה ויקבל סכום כסף ראשוני ויציב אותו ביתרה. אם הפרמטר balance קטן מאפס, הבנאי יציב ביתרה 0.
הוסיפו שיטות get עבור שתי התכונות.
הוסיפו למחלקה את השיטות הבאות:
public void transaction(double amount) – השיטה תבצע משיכה או הפקדה בחשבון. אם הפרמטר amount חיובי, השיטה תוסיף אותו ליתרה. אם amount שלילי, השיטה תחסיר אותו מהיתרה.
public void applyInterest() – השיטה תבצע את גביית או הפקדת הריבית בחשבון. אם החשבון נמצא בזכות (יתרה חיובית) השיטה תוסיף ליתרה את אחוז הריבית (הקבוע BALANCE_INTEREST). אם החשבון נמצא במשיכת יתר (יתרה שלילית) השיטה תוריד מהיתרה את אחוז הריבית (הקבוע OVERDRAFT_INTEREST). למשל, אם היתרה היתה 100, לאחר פעולת השיטה היתרה תהיה 105. אם היתרה היתה 100-, לאחר פעולת השיטה היתרה תהיה 110-.

ב. כתבו מחלקה בשם SavingAccount שיורשת מהמחלקה BankAccount. המחלקה תייצג חשבון בנק המיועד לחסכונות בלבד. לכן לא ניתן להגיע למצב של משיכת יתר בחשבון (כלומר, היתרה לא יכולה להיות שלילית).
הוסיפו למחלקה את הבנאים:
public SavingAccount(int num) – הבנאי יקבל את מספר החשבון, יציב אותו בתכונה ויציב 0 בתכונת היתרה.
public SavingAccount(int num, double balance) – הבנאי יקבל את מספר החשבון, יציב אותו בתכונה ויקבל סכום כסף ראשוני ויציב אותו ביתרה. אם הפרמטר balance קטן מאפס, הבנאי יציב ביתרה 0.
דרסו במחלקה את השיטה transaction כך שאם הסכום הוא חיובי (הפקדה) היא תוסיף אותו ליתרה, ואם הסכום הוא שלילי, השיטה תוודא שהורדתו מהיתרה לא גורמת ליתרה שלילית. אם ניתן למשוך את הסכום בלי להגיע למשיכת יתר, השיטה תבצע את המשיכה. אם לא, השיטה לא תבצע את המשיכה ולא תשנה את היתרה.

שאלה 9

ענו על השאלות הבאות. השתדלו לנמק היטב ולהמנע מניסוחים מעורפלים.
א. מהי מחלקה מופשטת (abstract)? לשם מה נשתמש במחלקה מופשטת?
ב. האם ניתן לרשת ממחלקה מופשטת?
ג. מהי שיטה מופשטת? לשם מה נשתמש בשיטה מופשטת?
ד. האם מחלקה מופשטת יכולה לרשת ממחלקה מופשטת אחרת?
ה. האם מחלקה מופשטת יכולה לרשת ממחלקה לא מופשטת?

ו. האם מחלקה מופשטת מכילה בנאי?
ז. האם מחלקה מופשטת יכולה להכיל שיטות לא מופשטות?
ח. האם מחלקה מופשטת חייבת להכיל לפחות שיטה מופשטת אחת?
ט. האם מחלקה לא מופשטת יכולה להכיל שיטה מופשטת?

שאלה 10

נתונה המחלקה הבאה:

```
public abstract class A
{
    public boolean f(int x)
    {
        return x == 2;
    }
}
```

בכל אחד מהסעיפים נתונה מחלקה B היורשת מ-A. עבור כל סעיף כתבו האם הקוד של מחלקה B עובר קומפילציה. אם הקוד אינו עובר קומפילציה, הסבירו למה.

א.

```
public class B extends A {
    public void f() {
        System.out.println("In f");
    }
}
```

ב.

```
public abstract class B extends A {
    public void f() {
        System.out.println("In f");
    }
}
```

ג.

```
public abstract class B extends A {
    public boolean f(int x) {
        return x > 2;
    }
}
```

ד.

```
public abstract class B extends A {
    public B(int x)
    {
        super(x);
    }
}
```

ה.

```
public abstract class B extends A {
    public B(int x)
    {
        System.out.println("In B");
    }
}
```

שאלה 11

כתבו מחלקה מופשטת בשם Shape. המחלקה תייצג צורה גאומטרית.
הוסיפו למחלקה את השיטות המופשטות הבאות:
public abstract double area() – השיטה תחשב את שטח הצורה.
public abstract double perimeter() – השיטה תחשב את היקף הצורה.
כתבו מחלקה בשם Rectangle היורשת מהמחלקה Shape. המחלקה תייצג מלבן ותכיל שתי תכונות:
width – מסוג double תייצג את רוחב המלבן.
height – מסוג double תייצג את גובה המלבן.
הוסיפו למחלקה בנאי שחתימתו:
public Rectangle(double width, double height) – הבנאי יקבל כפרמטרים את אורך וגובה המלבן ויציב אותם בתכונות. אם אחד מהנתונים אינו חיובי, הבנאי יציב במקומו את הערך 1.
הוסיפו שיטות get/set עבור האורך והגובה. בשיטות ה-set אם מתקבל מספר שאינו חיובי, השיטה לא תשנה את הערך של התכונה.
ממשו במחלקה את השיטות המופשטות הנורשות מהמחלקה Shape.
כתבו מחלקה בשם Circle היורשת מ-Shape. המחלקה תייצג מעגל ותכיל את התכונה:
radius – מסוג double, מייצגת את רדיוס המעגל.
הוסיפו למחלקה בנאי שחתימתו:
public Circle(double r) – הבנאי יקבל את הרדיוס ויציב אותו בתכונה. אם הפרמטר שלילי, הבנאי יציב 1 בתכונה.
הוסיפו שיטות get/set עבור תכונת הרדיוס. בשיטת ה-set, אם מתקבל מספר שאינו חיובי, השיטה לא תשנה את ערך התכונה.
ממשו במחלקה את השיטותת המופשטות הנורשות מהמחלקה Shape.
כתבו מחלקה בשם Square היורשת מ-Rectangle. המחלקה תייצג ריבוע.
הוסיפו למחלקה בנאי שחתימתו:
public Square(int length) – הבנאי יקבל את אורך צלע הריבוע ויציב אותה בתכונות. אם אורך הצלע אינו חיובי, יוצב הערך 1 בתכונות.
דרסו את השיטות setWidth() ו-setHeight() של המחלקה Rectangle כך שכל שינוי של צלע אחת יביא לשינוי זהה גם בצלע השניה.

שאלה 12

ענו על השאלות הבאות. השתדלו לנמק היטב ולהמנע מניסוחים מעורפלים:
א. מהו פולימורפיזם (רב צורתיות)?
ב. לשם מה נשתמש במנגנון הפולימורפיזם?
ג. מהו down casting? לאיזה צורך נשתמש במנגנון זה?
ד. מהו up casting? לאיזה צורך נשתמש במנגנון זה?
ה. מהו האופרטור instanceof?

שאלה 13

נתון קטע הקוד הבא:

```
Class1 c1 = new Class2();
Class1 c2 = new Class3();
```

א. נתון ש- Class1, Class2, Class3 הן מחלקות, וקטע הקוד עובר קומפילציה ורץ בצורה תקינה. מהם יחסי הירושה האפשריים בין שלוש המחלקות? ציירו את כל עצי הירושה האפשריים בהתאם לשורות הקוד.
ב. בהמשך לקטע הקוד שלעיל, נתונה השורה הבאה:

```
Class2 c3 = (Class2) c2;
```

בהנתן שגם שורה זו עוברת קומפילציה ורצה בצורה תקינה, אילו מבין עצי הירושה שציירתם בסעיף א' הוא עץ הירושה המביע בצורה נכונה את יחסי הירושה בין שלוש המחלקות?

שאלה 14

נתונות שתי המחלקות הבאות:

```
public class A
{ }

public class B extends A
{ }
```

בהנתן ה-main הבא, עבור כל שורה רשמו האם היא תקינה, או שהיא לא עוברת קומפילציה (הסבירו מדוע) או שהיא עוברת קומפילציה אבל גורמת לשגיאת ריצה (הסבירו מדוע):

```
public class Tester {
    public static void main(String[] args) {
        A a1 = new A();
        A a2 = new B();
        A a3 = (A) (new B());
        B b1 = new A();
        B b2 = (B)(new A());
        B b3 = new B();
        B b4 = (A)(new B());
    }
}
```

שאלה 15

נתונות שלוש המחלקות הבאות:

```
public class A
{ }

public class B extends A
{ }

public class C extends A
{ }
```

ונתון ה-main הבא. עבור כל שורה מהשורות המסומנות 1-7 רשמו האם הפלט יהיה true או false:

```
public class Tester {
    public static void main(String[] args) {
        A a1 = new A();
        A a2 = new B();
        A a3 = new C();
        B b1 = new B();
        C c1 = new C();

        System.out.println(a1 instanceof A);        // 1
        System.out.println(a1 instanceof B);        // 2
        System.out.println(a2 instanceof A);        // 3
        System.out.println(a2 instanceof B);        // 4
        System.out.println(a3 instanceof C);        // 5
        System.out.println(b1 instanceof A);        // 6
        System.out.println(c1 instanceof Object);   // 7
    }
}
```

שאלה 16

נתונות שתי המחלקות הבאות:

```
public class A
{
    public void f() {
        System.out.println("In A's f");
    }
    public void g() {
        System.out.println("In g");
    }
}

public class B extends A
{
    public void f() {
        System.out.println("In B's f");
    }
    public void h() {
        System.out.println("In h");
    }
}
```

ונתון ה-main הבא. עבור כל אחת מהשורות ב-main רשמו האם השורה תקינה (ומה הפלט), או שהשורה אינה עוברת קומפילציה (רשמו מדוע) או שהשורה עוברת קומפילציה אבל גורמת לשגיאה בזמן ריצה (רשמו מדוע):

```
public class Tester {
    public static void main(String[] args) {
        A a1 = new A();
        A a2 = new B();
        B b1 = new B();

        a1.f();
        a2.f();
        b1.f();
        a1.g();
        a2.g();
        b1.g();
        a1.h();
        a2.h();
        b1.h();
        ((B)a1).h();
        ((B)a2).h();
        ((A)b1).f();
    }
}
```

שאלה 17

נתונות שלוש המחלקות הבאות:

```
public class A {
    protected int x;

    public A(int x) {
        this.x = x;
    }
    public boolean f(A other) {
        return x == other.x;
    }
}
```

```
public class B extends A {
    private int y;

    public B(int x, int y) {
        super(x);
        this.y = y;
    }

    public boolean f(B other) {
        return y == other.y && x == other.x;
    }
}

public class C extends A {
    private int y;

    public C(int x, int y) {
        super(x);
        this.y = y;
    }

    public boolean f(A other) {
        return other instanceof C &&
               other.x = x && ((C)other).y == y;
    }
}
```

ונתון ה-main הבא. עבור כל אחת מהשורות ב-main רשמו האם השורה תקינה (ומה הפלט), או שהשורה אינה עוברת קומפילציה (רשמו מדוע) או שהשורה עוברת קומפילציה אבל גורמת לשגיאה בזמן ריצה (רשמו מדוע):

```
public class Tester {
    public static void main(String[] args) {
        A a1 = new B(1, 2);
        B b1 = new B(1, 3);
        A a2 = new C(1, 2);
        C c1 = new C(1, 2);

        System.out.println(b1.y == a1.y);
        System.out.println(b1.f((B)a2));
        System.out.println(a1.f(b1));
        System.out.println(b1.f(a1));
        System.out.println(a1.f(c1));
        System.out.println(a2.f(c1));
        System.out.println(c1.f(b1));
    }
}
```

שאלה 18

בהנתן המחלקה הבאה:

```
public class A
{
    public void f() {
        System.out.println("In f");
    }
}
```

עבור כל אחד מקטעי הקוד הבאים, רשמו האם הוא תקין, או שהוא לא עובר קומפילציה (הסבירו מדוע) או שהוא עובר קומפילציה אבל גורם לשגיאה בזמן ריצה (הסבירו מדוע):

א.

```
Object obj1 = new A();
obj1.f();
```

ב.

```
A a1 = new A();
System.out.println(a1.equals(a1));
```

ג.

```
Object obj1 = new Object();
A a1 = (A) obj1;
```

שאלה 19

נתונות שתי המחלקות הבאות:

```
public class A {
    public void f() {
        System.out.println("In A's f");
    }
    public void g() {
        System.out.println("In A's g");
        f();
    }
}

public class B extends A {
    public void f() {
        System.out.println("In B's f");
    }
}
```

ונתון ה-main הבא. רשמו מה יהיה הפלט כתוצאה מהפעלתו:

```
public class Tester {
    public static void main(String[] args) {
        A[] arr = new A[3];
        arr[0] = new A();
        arr[1] = new B();
        arr[2] = new B();

        for(int i = 0; i < arr.length; i++)
            arr[i].f();
        for(int i = 0; i < arr.length; i++)
            arr[i].g();
        int k = 0;
        for(int i = 0; i < arr.length; i++)
            if(arr[i] instanceof B)
                k++;
        System.out.println(arr.length - k);
    }
}
```

שאלה 20

א. כתבו מחלקה מופשטת בשם Employee המייצג עובד במפעל. לעובד יהיו התכונות הבאות:
name – שם העובד, מסוג String.
הוסיפו למחלקה בנאי שחתימתו:
public Employee(String name) – הבנאי יקבל את שם העובד ויציב אותו בתכונה.
הוסיפו למחלקה את שיטת ה-get עבור תכונת השם.
הוסיפו למחלקה את השיטה
public String toString() – השיטה תחזיר את פרטי העובד בפורמט הבא. למשל, אם שם העובד הוא David השיטה תחזיר את המחרוזת:
Employee's name: David
הוסיפו למחלקה את השיטה המופשטת הבאה:
public abstract double salary() – השיטה תחזיר את משכורתו של העובד.

ב. כתבו מחלקה בשם Worker היורשת מ-Employee. המחלקה תייצג פועל בפס היצור במפעל. לפועל יהיו התכונות הבאות:
wagePerHour – השכר לשעה שמקבל הפועל, מסוג double.
hours – מספר השעות שעבד הפועל בחודש, מסוג int.
הוסיפו למחלקה בנאי שחתימתו:
public Worker(String name, double wage) – הבנאי יקבל כפרמטרים את שם העובד והשכר לשעה, ויציב אותם לתכונות.
הוסיפו למחלקה את השיטות get/set עבור התכונות של Manager.
הוסיפו למחלקה את השיטות הבאות:
public String toString() – השיטה תחזיר את פרטי הפועל בפורמט הבא. למשל, אם לעובד קוראים David, הוא מרוויח 50 שקל לשעה ועבד 10 שעות, השיטה תחזיר את המחרוזת:
Employee's name: David, wage = 50, hours = 10.
public double salary() – השיטה תחשב את שכרו של העובד. שכרו של העובד הוא מספר השעות שעבד כפול השכר לשעה.

ג. כתבו מחלקה בשם Manager היורשת מ-Employee. המחלקה תייצג מנהל במפעל. למנהל יהיו התכונות הבאות:
wage – שכרו של המנהל, מסוג double.
numOfWorkers – מספר העובדים שכפופים למנהל, מסוג int.
הוסיפו למחלקה בנאי שחתימתו:
public Manager(String name, double wage, int workers) – הבנאי יקבל את ערכי התכונות ויציב אותם בתכונות.
הוסיפו שיטות get/set עבור כל אחת מהתכונות של Manager.
הוסיפו למחלקה את השיטות הבאות:
public String toString() – השיטה תחזיר את פרטי המנהל בפורמט הבא. למשל, אם למנהל קוראים David, הוא מרוויח 5000 שקל ועובדים תחתיו 10 פועלים, השיטה תחזיר את המחרוזת:
Employee's name: David, wage = 5000, workers = 10.
public double salary() – השיטה תחשב את שכרו של המנהל. שכרו של המנהל הוא השכר הבסיסי שלו ובנוסף, בונוס לפי מספר העובדים הכפופים לו. אם למנהל כפופים עד עשרה עובדים, השכר שלו הוא שכרו הבסיסי (wage). אם כפופים לו בין עשרה (לא כולל) לעשרים עובדים (כולל) שכרו הוא השכר הבסיסי + 50 שקלים עבור כל עובד שהוא מנהל. אם כפופים לו מעל 20 עובדים, שכרו הוא השכר הבסיסי + 50 שקל עבור כל אחד מעשרת העובדים + 500 שקל.
למשל, אם המנהל מרוויח 5000 שקל וכפופים לו 10 עובדים, שכרו יהיה 5000 שקל.
אם המנהל מרוויח 5000 שקל וכפופים לו 15 עובדים, שכרו יהיה 5250 שקל.
אם המנהל מרוויח 5000 שקל וכפופים לו 25 עובדים, שכרו יהיה 6000 שקל.

ד. כתבו מחלקה בשם Factory שתייצג את המפעל. למחלקה יהיו התכונות הבאות:
emps – מסוג Employee[], מערך שישמור את עובדי המפעל.
numOfEmps – מסוג int, ישמור את מספר העובדים בפועל במפעל.
MAX_EMPS – קבוע שערכו 100 ויציין את מספר העובדים המקסימלי שיכולים להיות במפעל.
הוסיפו למחלקה בנאי ריק שיאתחל את המערך לגודל MAX_EMPS ואת התכונה numOfEmps לאפס.
הוסיפו למחלקה את השיטות הבאות:
public boolean addEmployee(Employee e) – השיטה תקבל אובייקט מסוג Employee. אם יש מקום במערך לעובד נוסף, השיטה תוסיף את e למערך ותחזיר true. אחרת השיטה תחזיר false.
public int numOfWorkers() – השיטה תחזיר את מספר **הפועלים** שיש במפעל.
public Manager mostManager() – השיטה תחזיר אובייקט מסוג Manager שמייצג את המנהל שאליו כפופים הכי הרבה עובדים.
public Worker minHours() – השיטה תחזיר אובייקט מסוג Worker שמייצג את הפועל שעובד את מספר השעות המינימלי.
public Employee maxSalary() – השיטה תחזיר אובייקט מסוג Employee שמרוויח את המשכורת המקסימלית.
public Employee getByName(String eName) – השיטה תחזיר אובייקט מסוג Employee ששמו הוא eName. אם עובד כזה אינו נמצא, השיטה תחזיר null.
public String toString() – השיטה תחזיר את פרטי העובדים במפעל בצורת מחרוזת, כאשר בתחילת המחרוזת יופיע המשפט :Workers in factory ואחריו פרטי העובדים, כל עובד בשורה נפרדת.
public double salaryByName(String eName) – השיטה תקבל כפרמטר של של עובד, ותחזיר את המשכורת החודשית שלו. אם עובד כזה אינו קיים, השיטה תחזיר 0.

שאלה 21

ברצונינו לכתוב מחלקות המייצגות ביטויים חשבוניים. ביטויים חשבוניים מתחלקים לשני סוגים:
ביטוי אטומי הוא מספר, למשל 3.0, 7.8, 4.1- הם ביטויים אטומיים.
ביטוי מורכב הוא ביטוי מהצורה a operator b כאשר a ו-b הם ביטויים מורכבים אחרים, ו-operator הוא אחד משתי פעולות החשבון - ,+. למשל, הביטויים הבאים הם ביטויים מורכבים:

4 + 5
3 – 2 + 5
1 + 4 + 7 – 5

א. כתבו מחלקה מופשטת בשם Expression שתייצג ביטוי חשבוני. המחלקה תכיל שיטה מופשטת שחתימתה:
public abstract double value() – השיטה תחזיר את ערך הביטוי החשבוני.
כתבו מחלקה בשם AtomicExpression היורשת מהמחלקה Expression ומייצגת ביטוי אטומי.
המחלקה תכיל תכונה אחת:
a – מסוג double, מייצגת את המספר שמייצג הביטוי.
הוסיפו למחלקה בנאי שחתימתו:
public AtomicExpression(double a) – הבנאי יקבל את ערך הביטוי האטומי ויציב אותו בתכונה.
ממשו את השיטה המופשטת. ערכו של ביטוי אטומי הוא ערך המספר שמייצג הביטוי.

ב. כתבו מחלקה מופשטת בשם CompundExpression היורשת מ-Expression ומייצגת ביטוי חשבוני מורכב. המחלקה תכיל שתי תכונות:
a – מסוג Expression, מייצג את הארגומנט השמאלי בביטוי.
b – מסוג Expression, מייצג את הארגומנט הימני בביטוי.
הוסיפו למחלקה בנאי שחתימתו:

public CompoundExpression(Expression a, Expression b) – הבנאי יקבל את ערכי שני הארגומנטים ויציב אותם לתכונות.
דרסו במחלקה את השיטה public boolean equals(Object other) שנורשת מ-Object. המחלקה תקבל אובייקט של ביטוי חשבוני, ותשווה אותו לאובייקט this. שני ביטויים חשבוניים יהיו שווים אם ערכם שווה (כלומר, תוצאת החישוב שלהם שווה). למשל, הביטוי 8.0 והביטוי 4.0+4.0 הם שווים.

ג. כתבו מחלקה בשם AdditionExpression היורשת מ-CompoundExpression . המחלקה תייצג ביטוי חשבוני של חיבור. הוסיפו למחלקה בנאי שחתימתו:
public AdditionExpression(Expression a, Expression b) – הבנאי יקבל את ערכי שני הארגומנטים ויציב אותם לתכונות.
ממשו במחלקה את השיטה המופשטת הנורשת מ-Expression. ערכו של ביטוי החיבור הוא ערכו של a פלוס ערכו של b.

ד. כתבו מחלקה בשם SubtractionExpression היורשת מ-CompoundExpression . המחלקה תייצג ביטוי חשבוני של חיסור. הוסיפו למחלקה בנאי שחתימתו:
public SubtractionExpression(Expression a, Expression b) – הבנאי יקבל את ערכי שני הארגומנטים ויציב אותם לתכונות.
ממשו במחלקה את השיטה המופשטת הנורשת מ-Expression. ערכו של ביטוי החיסור הוא ערכו של a מינוס ערכו של b.

יעילות

קובץ תרגילים זה מכיל שאלות בנושא יעילות (סיבוכיות) זמן ריצה בשפת ג'אווה. חלק מהשאלות הן תאורטיות ונועדו לעזור לקורא לבחון את הידע שלו בנושא. רוב השאלות הן שאלות כתיבת קוד, ומסודרות מהקל לקשה. כדאי מאוד להריץ את התשובות (גם את הקלות שבהן) ולוודא שאתם מבינים איך הקוד עובד.

שאלה 1

ענו על השאלות הבאות. נמקו היטב את התשובות והשתדלו לוודא שאתם מבינים את המושגים.

1. מהי יעילות זמן ריצה של אלגוריתם?
2. למה חשוב למדוד יעילות זמן ריצה?
3. למה מדידת זמן אבסולוטי של ריצת אלגוריתם אינה מספקת?
4. איך מודדים זמן ריצה?
5. מהו "סדר גודל"? למה הוא משמש?
6. מהי "הזנחה"? איך היא קשורה לחישוב יעילות זמן ריצה?
7. מהו זמן ריצה קבוע? לוגריתמי? ליניארי? ריבועי? רשמו לפניכם את סדרי הגודל השונים, מהטוב ביותר לגרוע ביותר. סולם זה ישמש אתכם בכל השאלות בקובץ זה.
8. מהו "המקרה הטוב ביותר"? "המקרה הגרוע ביותר"?

שאלה 2

בכל אחד מהסעיפים הבאים נתונה שיטה בשם f. חשבו את זמן הריצה של השיטה ונמקו את תשובותיכם.

א.

```
public void f(int n)
{
    for(int i = 0; i < n; i++)
        System.out.println("*");
}
```

ב.

```
public void f(int n)
{
    for(int i = 0; i < n; i+=2)
        System.out.println("*");
}
```

ג.

```
public void f(int n)
{
    for(int i = 10; i >= 1; i--)
        System.out.println("*");
}
```

ד.

```
public void f(int n)
{
    for(int i = 0; i < n; i++)
        for(int j = n; j > 0; j--)
            System.out.println("*");
}
```


shay.tavor@gmail.com

ה.

```
public void f(int n)
{
    for(int i = 1; i <= n; i*=2)
        System.out.println("*");
}
```

ו.

```
public void f(int n)
{
    for(int i = 0; i < n; i++)
    {
        int j = n;
        while(j > 0)
        {
            System.out.println("*");
            j /= 3;
        }
    }
}
```

ז.

```
public void f(int n)
{
    for(int i = n; i > 0; j /= 4)
        for(int j = 0; j < 5; j++)
            System.out.println("*");
}
```

ח.

```
public void f(int n)
{
    for(int i = 1; i <= n; i++)
        g(i);
}
public void g(int x)
{
    for(int i = 0; i < x; i++)
        System.out.println("*");
}
```

ט.

```
public void f(int n)
{
    for(int i = 1; i <= n; i*=2)
        System.out.println("*");
    for(int i = n; i > 0; i/=2)
        System.out.println("*");
    for(int i = 1; i <= n; i+=3)
        System.out.println("*");
}
```

י.

```
public void f(int n)
{
    for(int i = 1; i < n; i*=2)
    {
        for(int j = 0; j < n; j += 5)
            System.out.println("*");
        for(int k = n; k < n / 2; k--)
            System.out.println("*");
    }
}
```


shay.tavor@gmail.com

שאלה 3

עבור כל אחת מהמשימות הבאות, רשמו מה יעילות האלגוריתם במקרה הטוב ביותר והגרוע ביותר.
נמקו את תשובותיכם:
א. מציאת מספר מקסימלי במערך שמכיל n תאים.
ב. מציאת מספר מקסימלי במערך **ממויין בסדר עולה** שמכיל n תאים.
ג. הדפסת כל איבריו של מערך שמכיל n תאים.
ד. מציאת מספר במערך שמכיל n תאים.
ה. מציאת מספר במערך ממויין בסדר עולה שמכיל n תאים.
ו. מציאת ההפרש המינימלי בין שני איברים כלשהם במערך.
ז. מיון מערך של מספרים שלמים בסדר עולה.

בחלק מהשאלות הבאות תתבקשו להסביר מה שיטה נתונה עושה ולא כיצד היא מבצעת זאת. הכוונה היא לנסח מהי הפונקציונליות שמבצעת השיטה. אם השיטה מדפיסה פלט כלשהו – מה משמעות הפלט שהיא מדפיסה? אם השיטה מחזירה ערך בוליאני – מתי היא מחזירה true ומתי היא מחזירה false? אם השיטה מחזירה מספר – מהו המספר הזה? המטרה היא לא לתרגם לעברית את שורות הקוד, אלא לנסות להבין מה המשימה הכללית שאותה מבצעת השיטה. חשבו כאילו אתם מוכרים את השיטה למישהו שלא יודע ג'אווה. כשהקונה ישאל אתכם "מה עושה השיטה" תצטרכו להסביר לו בצורה כללית ככל האפשר, וללא שימוש במונחים טכניים מתוך גוף השיטה...

שאלה 4

נתונה השיטה הבאה:

```
public int f1(int[] a)
{
    int m = 0;
    for(int i = 0; i < a.length; i++)
        for(int j = 0; j < a.length; j++)
            if(Math.abs(a[i] - a[j]) > m)
                m = Math.abs(a[i] - a[j]);
    return m;
}
```

השיטה Math.abs מחזירה את הערך המוחלט של הביטוי שבסוגריים.
א. מה מבצעת השיטה f1? הסבירו מה היא מבצעת ולא כיצד היא מבצעת זאת.
ב. מה סיבוכיות זמן הריצה של השיטה? הסבירו.
ג. כתבו את השיטה f1 מחדש כך שתבצע את אותה פעולה שביצעה השיטה המקורית, בסיבוכיות זמן ריצה קטנה יותר (קטנה בסדר גודל ולא בקבוע).
ד. מה סיבוכיות זמן הריצה של השיטה שכתבתם?

שאלה 5

נתונה השיטה הבאה:

```
public boolean f2(int[] a, int n)
{
    for(int i = 0; i < a.length; i++)
        for(int j = i+1; j < a.length; j++)
            if(a[i] + a[j] == n)
                return true;
    return false;
}
```

א. בהנחה שהשיטה מקבלת מערך a **ממויין בסדר עולה ממש**, מה מבצעת השיטה f2? הסבירו מה היא מבצעת ולא כיצד היא מבצעת זאת.
ב. מה סיבוכיות זמן הריצה של השיטה? הסבירו.
ג. כתבו את השיטה f2 מחדש כך שתבצע את אותה פעולה שביצעה השיטה המקורית, בסיבוכיות זמן ריצה קטנה יותר (קטנה בסדר גודל ולא בקבוע).
ד. מה סיבוכיות זמן הריצה של השיטה שכתבתם?

שאלה 6

נתונה השיטה הבאה:

```
public int f3(int[] a)
{
    int m = 0;
    for(int i = 0; i < a.length; i++)
    {
        int c = 0;
        for(int j = 0; j < a.length; j++)
            if(a[j] == a[i])
                c++;
        if(c > m)
            m = c;
    }
    return m;
}
```

א. מה מבצעת השיטה f3? הסבירו מה היא מבצעת ולא כיצד היא מבצעת זאת.
ב. מה סיבוכיות זמן הריצה של השיטה? הסבירו.
ג. כתבו את השיטה f3 מחדש כך שתבצע את אותה פעולה שביצעה השיטה המקורית, בסיבוכיות זמן ריצה קטנה יותר (קטנה בסדר גודל ולא בקבוע).
ד. מה סיבוכיות זמן הריצה של השיטה שכתבתם?

שאלה 7

נתונה השיטה הבאה:

```
public int f4(int[] a, int num)
{
    int size = 0;
    boolean found = true;

    for(int i = 1; i < a.length && found; i++)
    {
        found = false;
        for(int j = 0; j < a.length && !found; j++)
            if(a[j] == num * i)
            {
                size++;
                found = true;
            }
    }
    return size;
}
```

א. מה מבצעת השיטה f4? הסבירו מה היא מבצעת ולא כיצד היא מבצעת זאת.
ב. מה סיבוכיות זמן הריצה של השיטה? הסבירו.

ג. כתבו את השיטה f4 מחדש כך שתבצע את אותה פעולה שביצעה השיטה המקורית, בסיבוכיות זמן ריצה קטנה יותר (קטנה בסדר גודל ולא בקבוע).
ד. מה סיבוכיות זמן הריצה של השיטה שכתבתם?

שאלה 8

נתונה השיטה הבאה:

```
public int f5(int[] a)
{
    int count = 1;
    for(int i = 1; i < a.length; i++)
    {
        boolean b = true;
        for(int j = 0; j < i && b; j++)
            if(a[j] == a[i])
                b = false;
        if(b)
            count++;
    }
    return count;
}
```

א. מה מבצעת השיטה f5? הסבירו מה היא מבצעת ולא כיצד היא מבצעת זאת.
ב. מה סיבוכיות זמן הריצה של השיטה? הסבירו.
ג. כתבו את השיטה f5 מחדש כך שתבצע את אותה פעולה שביצעה השיטה המקורית, בסיבוכיות זמן ריצה קטנה יותר (קטנה בסדר גודל ולא בקבוע).
ד. מה סיבוכיות זמן הריצה של השיטה שכתבתם?

שאלה 9

נתונה השיטה הבאה:

```
public boolean f6(int[] a, int[] b)
{
   for(int i = 0; i < a.length; i++)
       for(int j = 0; j < b.length; j++)
            if(b[j] >= a[i])
                return false;
   return true;
}
```

א. מה מבצעת השיטה f6? הסבירו מה היא מבצעת ולא כיצד היא מבצעת זאת.
ב. מה סיבוכיות זמן הריצה של השיטה? הסבירו.
ג. כתבו את השיטה f6 מחדש כך שתבצע את אותה פעולה שביצעה השיטה המקורית, בסיבוכיות זמן ריצה קטנה יותר (קטנה בסדר גודל ולא בקבוע).
ד. מה סיבוכיות זמן הריצה של השיטה שכתבתם?

שאלה 10

נתונה השיטה הבאה:

```
public boolean f7(int[] a, int[] b)
{
    for(int i = 0; i < b.length; i++)
        for(int j = 0; j < a.length-1; j++)
            if(a[j] + a[j+1] == b[i])
                return true;
    return false;
}
```

א. בהנתן שני מערכים, a, b, כאשר המערך a **ממויין בסדר עולה,** מה מבצעת השיטה f7? הסבירו מה היא מבצעת ולא כיצד היא מבצעת זאת.
ב. מה סיבוכיות זמן הריצה של השיטה? הסבירו.
ג. כתבו את השיטה f7 מחדש כך שתבצע את אותה פעולה שביצעה השיטה המקורית, בסיבוכיות זמן ריצה קטנה יותר (קטנה בסדר גודל ולא בקבוע).
ד. מה סיבוכיות זמן הריצה של השיטה שכתבתם?

שאלה 11

נתונה השיטה הבאה:

```
public int f8(int[] a)
{
    int s1, s2;
    for(int i = 0; i < a.length-1; i++)
    {
        s1 = s2 = 0;
        for(int j = 0; j <= i; j++)
            s1 += a[j];
        for(int k = i+1; k < a.length; k++)
            s2 += a[k];
        if(s1 == s2)
            return i;
    }
    return -1;
}
```

א. בהנתן מערך a **מלא במספרים חיוביים בלבד**, מה מבצעת השיטה f8? הסבירו מה היא מבצעת ולא כיצד היא מבצעת זאת.
ב. מה סיבוכיות זמן הריצה של השיטה? הסבירו.

ג. כתבו את השיטה f8 מחדש כך שתבצע את אותה פעולה שביצעה השיטה המקורית, בסיבוכיות

זמן ריצה קטנה יותר (קטנה בסדר גודל ולא בקבוע).

ד. מה סיבוכיות זמן הריצה של השיטה שכתבתם?

שאלה 12

א. כתבו שיטה שחתימתה:

public boolean bigger(int[] a, int num)

השיטה תקבל כפרמטרים מערך a ומספר num. השיטה תחזיר true אם קיימים במערך שני מספרים שהפרשם בערך מוחלט **גדול** מ-num, ו-false אם לא קיימים מספרים כאלה. השיטה צריכה להיות יעילה ככל הניתן.

ב. כתבו שיטה שחתימתה:

public boolean smaller(int[] a, int num)

השיטה תקבל כפרמטרים מערך a ומספר num. השיטה תחזיר true אם קיימים במערך שני מספרים שהפרשם בערך מוחלט **קטן** מ-num, ו-false אם לא קיימים מספרים כאלה. השיטה צריכה להיות יעילה ככל הניתן.

שאלה 13

כתבו שיטה שחתימתה:

public void sortEvenOdd(int[] a)

השיטה תקבל כפרמטר מערך של מספרים שלמים חיוביים. השיטה תמיין את המערך כך שכל המספרים הזוגיים יופיעו בתחילת המערך, וכל המספרים האי זוגיים יופיעו בסוף המערך. כלומר השיטה תחלק את המערך לשני חלקים – חלק של מספרים זוגיים וחלק של מספרים אי זוגיים. אין חשיבות לסדר הפנימי בתוך כל אחד מהחלקים. השיטה צריכה לעבוד ביעילות זמן ריצה של O(n).

שאלה 14

כתבו שיטה שחתימתה:

public int[] unify(int[] a, int[] b)

השיטה תקבל כפרמטרים שני מערכים. בכל מערך המספרים שונים זה מזה. השיטה תאחד את שני המערכים למערך אחד, כך שהמערך יכיל את כל המספרים מ-a ומ-b, בלי חזרות (כלומר, כל מספר יופיע רק פעם אחת). השיטה תחזיר את מערך התוצאה.

למשל, אם a = {1, 6, 3, 9} ו-b = {6, 2, 7, 1, 5}, השיטה תחזיר את המערך {1, 6, 3, 9, 2, 7, 5}. אין חשיבות לסדר האיברים במערך התוצאה.

השיטה צריכה לעבוד ביעילות זמן ריצה של O(nlogn) .

שאלה 15

נתונה השיטה הבאה:

```
public int what(String s, char start, char end)
{
   int count = 0;
   for(int i = 0; i < s.length(); i++)
   {
    if(s.charAt(i) == start)
        for(int j = i; j < s.length(); j++)
        {
            if(s.charAt(j) == end)
                count++;
        }
   }
   return count;
}
```

א. מה מבצעת השיטה what? הסבירו מה היא מבצעת ולא כיצד היא מבצעת זאת.

ב. מה סיבוכיות זמן הריצה של השיטה? הסבירו.

ג. כתבו את השיטה what מחדש כך שתבצע את אותה פעולה שביצעה השיטה המקורית, בסיבוכיות זמן ריצה קטנה יותר (קטנה בסדר גודל ולא בקבוע).

ד. מה סיבוכיות זמן הריצה של השיטה שכתבתם?

שאלה 16

נתון מערך דו ממדי בו כל האיברים בשורה i קטנים ממש מכל האיברים בשורה i+1. לדוגמא, המערך הבא מקיים את התנאי:

	0	1	2	3	4
0	0	5	1	2	3
1	6	8	7	10	7
2	13	14	12	15	11
3	17	19	16	19	24

כתבו שיטה שחתימתה:

```
public boolean find1(int[][] a, int num)
```

השיטה תקבל כפרמטר מערך דו ממדי המסודר כמוסבר בתחילת השאלה, ומספר num. השיטה תחזיר true אם המספר נמצא במערך, ו-false אחרת. השיטה צריכה להיות יעילה ככל הניתן.

שאלה 17

נגדיר – מערך דו ממדי יקרא **ממויין שורות** אם כל שורה במערך ממויינת בסדר עולה ממש. למשל, המערך הבא הוא מערך ממויין שורות:

	0	1	2	3	4
0	0	3	4	6	7
1	1	3	5	7	8
2	3	5	9	10	12
3	5	9	10	12	19

כתבו שיטה שחתימתה:

```
public boolean find2(int[][] a, int num)
```

השיטה תקבל כפרמטר מערך דו ממדי ממויין שורות כמוסבר לעיל ומספר num. השיטה תחזיר true אם המספר נמצא במערך, ו-false אם לא. השיטה צריכה להיות יעילה ככל הניתן.

שאלה 18

בהנתן מערך דו ממדי שכל איבריו הם 0 או 1, נגדיר **בור** במערך כתא בשורה i ועמודה j כך שכל האיברים בשורה ה-i שווים ל-1 (חוץ מהאיבר ה (i,j) וכל האיברים בעמודה ה-j שווים לאחד (חוץ מהאיבר ה-(i, j).

לדוגמא, נתונים המערכים הבאים. במערך הימני קיים בור בתא [1,2] (מסומן בתכלת) ובמערך השמאלי לא קיים.

	0	1	2	3	4
0	0	1	1	0	0
1	0	1	0	0	1
2	0	0	0	0	0
3	0	1	0	1	1

	0	1	2	3	4
0	0	0	1	0	0
1	1	1	0	1	1
2	0	0	1	0	0
3	0	1	1	0	1

כתבו שיטה שחתימתה:

```
public boolean isSinkExist(int[][] a)
```

השיטה תקבל כפרמטר מערך דו ממדי המכיל אפסים ואחדים, ותחזיר true אם קיים בור במערך ו-false אם לא קיים. השיטה צריכה להיות יעילה ככל הניתן.

שאלה 19

כתבו שיטה שחתימתה:

public Boolean howMany(int[] a, int x)

השיטה מקבלת כפרמטרים מערך a **ממויין בסדר עולה** ומספר x. השיטה תחזיר את מספר הפעמים ש-x מופיע במערך. השיטה צריכה להיות ביעילות זמן הריצה הטובה ביותר.

שאלה 20

נגדיר – מערך ממויין לסירוגין הוא מערך שבו כל המספרים באינדקסים הזוגיים (כולל אפס) ממויינים בסדר עולה, וכל המספרים באינדקסים האי זוגיים ממויינים בסדר יורד. למשל, המערך הבא ממויין לסירוגין –

2	1	3	4	7	5	10	7	12	15
0	1	2	3	4	5	6	7	8	9

כתבו שיטה שחתימתה:

public boolean find(int[] a, int x)

השיטה תקבל כפרמטרים מערך ממויין לסירוגין כמתואר, ומספר x. השיטה תחזיר true אם x נמצא במערך ו-false אם לא. השיטה צריכה להיות ביעילות זמן הריצה הטובה ביותר.

shay.tavor@gmail.com

רקורסיה ועקיבה לאחור

קובץ תרגילים זה מכיל שאלות בנושא רקורסיה ו-backtracking (עקיבה לאחור). בשפת ג'אווה. חלק מהשאלות הן תאורטיות ונועדו לעזור לקורא לבחון את הידע שלו בנושא. רוב השאלות הן שאלות כתיבת קוד, ומסודרות מהקל לקשה. כדאי מאוד להריץ את התשובות (גם את הקלות שבהן) ולוודא שאתם מבינים איך הקוד עובד. כיוון שהתכנות הרקורסיבי מהווה אתגר חשיבתי, מומלץ מאוד לתרגל את דרך החשיבה הרקורסיבית עבור כל שאלה – לנסות לנסח את האלגוריתם בצורה רקורסיבית בעברית לפני שכותבים את הקוד.

שאלה 1

ענו על השאלות הבאות. השתדלו לנמק בצורה ברורה ולא מעורפלת:

10. מה זאת רקורסיה?
11. מהו צעד הרקורסיה?
12. מהי הקריאה הרקורסיבית והקטנת הקלט?
13. מהו תנאי העצירה?
14. מהי רקורסיה אינסופית?

שאלה 2

בכל אחד מהסעיפים הבאים נתונה שיטה. עבור כל שיטה כתבו מה יהיה פלט השיטה. בכל סעיף נתונה הקריאה הראשונה לשיטה.

א. מה יהיה הפלט עבור הקריאה: f1(5)?

```
public void f1(int i)
{
    if(i > 0)
    {
        System.out.print("*");
        f1(i-1);
    }
}
```

ב. מה יהיה הפלט עבור הקריאה f2(10)?

```
public int f2(int x)
{
    if(x < 3)
        return 0;
    return 1 + f2(x-3);
}
```

ג. מה יהיה הפלט עבו הקריאה f3(5, 4)?

```
public int f3(int n, int m)
{
    if(m == 1)
        return n;
    return n + f3(n, m-1);
}
```


shay.tavor@gmail.com

ד. מה יהיה הפלט עבור הקריאה f4(5, 1)?

```
public int f4(int x, int y)
{
    if(x == 0)
        return 0;
    return x * y + f4(x-1, y*-1);
}
```

שאלה 3

בכל אחד מהסעיפים הבאים ישנה שיטה רקורסיבית שהרצתה תגרום לרקורסיה אינסופית. עבור כל שיטה תנו דוגמא לקלט שעבורו תהיה רקורסיה אינסופית, והציעו דרך לשנות את הקוד כך שתמנע הרקורסיה האינסופית.

א.

```
public int g1(int x, int y)
{
    int z = g1(x-1, y+1);

    if(x == 0)
        return y;
}
```

ב.

```
public void g2(int x)
{
    System.out.print(x % 10);
    g2(x / 10);
}
```

****

כתבו מחלקה בשם RecursiveMethods. את התשובות לשאלות הבאות כתבו כשיטות בתוך מחלקה זו. בכל שאלה ניתן להגדיר גם **שיטות עזר פרטיות** בעת הצורך. בכל השיטות **אין להשתמש בלולאות כלל!**

שאלה 4

כתבו שיטה רקורסיבית שחתימתה:

public void printStars(int n)

השיטה תקבל כפרמטר מספר שלם n ותדפיס n כוכביות על המסך.

שאלה 5

כתבו שיטה רקורסיבית שחתימתה:

public int factorial(int n)

השיטה תקבל כפרמטר מספר שלם n ותחזיר את העצרת של n. עצרת מוגדרת כמכפלת כל המספרים מ-1 עד n. למשל,

4! = 1*2*3*4 = 24
5! = 1*2*3*4*5 = 120

ובאופן כללי:

n! = 1*2*3*…*n

אם הפרמטר n אינו חיובי, השיטה תחזיר 1.

שאלה 6

כתבו שיטה רקורסיבית שחתימתה:

public int power(int n, int m)

השיטה תקבל כפרמטרים שני מספרים שלמים n, m ותחזיר את n בחזקת m. למשל, אם n = 2 ו- m=3 השיטה תחזיר 8.

שאלה 7

כתבו שיטה רקורסיבית שחתימתה:

public int howManyDigits(int n)

השיטה תקבל כפרמטר מספר שלם, ותחזיר את מספר הספרות במספר. למשל, אם n = 3426 השיטה תחזיר 4.

שאלה 8

כתבו שיטה רקורסיבית שחתימתה:

public int howManyEven(int num)

השיטה תקבל כפרמטר מספר שלם, ותחזיר כמה ספרות במספר הן זוגיות. למשל, עבור המספר 25643 השיטה תחזיר 3 כיוון שיש 3 ספרות זוגיות במספר (2, 6, 4).

שאלה 9

כתבו שיטה רקורסיבית שחתימתה:

public int sumDigits(int n)

השיטה תקבל מספר שלם n ותחזיר את סכום הספרות במספר. למשל, אם n = 2198 השיטה תחזיר 20.

שאלה 10

כתבו שיטה רקורסיבית שחתימתה:

public boolean isAscending(int num)

השיטה תקבל כפרמטר מספר שלם, ותחזיר true אם הספרות במספר מסודרות בסדר עולה ממש, משמאל לימין. למשל, המספרים הבאים הם עולים: 1234, 425689, 267, 3. המספרים הבאים אינם מספרים עולים: 264, 2556, 7654.

שאלה 11

כתבו שיטה רקורסיבית שחתימתה:

public int maxDigit(int num)

השיטה תקבל כפרמטר מספר שלם, ותחזיר את הספרה המקסימלית במספר. למשל, עבור המספר 352875 השיטה תחזיר 8.

שאלה 12

כתבו שיטה רקורסיבית שחתימתה:

public int howManyOccurences(int num, int d)

השיטה תקבל כפרמטרים מספר שלם num וספרה d. השיטה תחזיר כמה פעמים מופיעה הספרה d במספר num.

למשל, אם המספר הוא 1434624 ו-d = 4 השיטה תחזיר 3. אם עבור אותו מספר d = 3 השיטה תחזיר 1, ואם d = 9 השיטה תחזיר 0.

שאלה 13

כתבו שיטה רקורסיבית שחתימתה:

public int filter(int num, int d)

השיטה תקבל כפרמטרים מספר שלם num וספרה d. השיטה תחזיר מספר חדש שמכיל את כל הספרות מ-num שגדולות או שוות ל-d.

למשל, אם num = 316462 ו- d = 3 השיטה תחזיר את המספר 3646.

שאלה 14

כתבו שיטה רקורסיבית שחתימתה:

public int reverse(int num)

השיטה תקבל כפרמטר מספר שלם ותחזיר את המספר ההפוך (כלומר, מספר שסדר ספרותיו הפוך מהמספר המקורי). למשל, אם num = 3628 השיטה תחזיר 8263. אם num = 4 השיטה תחזיר 4.

שאלה 15

סדרת פיבונאצ'י היא סדרת מספרים שמוגדרת באופן הבא:

A0 = 0
A1 = 1
A2 = A1 + A0 = 1
A3 = A2 + A1 = 2
A4 = A3 + A2 = 3
A5 = A4 + A3 = 5

כלומר, האיבר הראשון בסדרה (מסומן כ A0) שווה לאפס, והאיבר השני (מסומן כ-A1) שווה לאחד.
כל איבר אחר בסדרה שווה לסכום שני האיברים הקודמים לו. באופן כללי, האיבר ה-n בסדרה הוא:

An = An-1 + An-2

כתבו שיטה רקורסיבית שחתימתה:

public int fibo(int n)

השיטה תקבל כפרמטר מספר שלם n ותחזיר את האיבר ה-n בסדרת פיבונאצ'י.

שאלה 16

מספר ראשוני הוא מספר שלם שמתחלק ללא שארית רק בעצמו ובאחד. למשל, המספרים 3, 5, 7, 13 הם מספרים ראשוניים. המספרים 6, 9, 21 הם לא ראשוניים.

א. כתבו שיטה רקורסיבית שחתימתה:

public boolean isPrime(int n)

השיטה תקבל כפרמטר מספר שלם, ותחזיר true אם המספר הוא ראשוני, ו-false אם לא.

ב. כתבו שיטה רקורסיבית שחתימתה:

public void printAllPrimes(int a, int b)

השיטה תקבל כפרמטרים שני מספרים a, b ותדפיס על המסך את כל המספרים הראשוניים בין a ל-b כולל. למשל, אם a = 3 ו- b = 10 השיטה תדפיס את המספרים 3, 5, 7.

ג. כתבו שיטה רקורסיבית שחתימתה:

public int differencePrimes(int a, int b)

השיטה תקבל כפרמטרים שני מספרים a, b ותחזיר את ההפרש בין המספר הראשוני הגדול ביותר בניהם והמספר הראשוני הקטן ביותר בניהם. למשל, אם a = 4 ו- b = 20 השיטה תחזיר את המספר 14, כיוון שהמספר הראשוני הגדול ביותר בתחום הוא 19, והקטן ביותר הוא 5.

שאלה 17

כתבו שיטה רקורסיבית שחתימתה:

public int formula1(int n)

השיטה תקבל כפרמטר מספר שלם n ותחזיר את תוצאת הנוסחה הבאה:

$$n + n^2 + n^3 + \ldots + n^n$$

למשל, אם n=4 השיטה תחזיר את המספר 340, שכן: $4 + 4^2 + 4^3 + 4^4 = 340$

שאלה 18

כתבו שיטה רקורסיבית שחתימתה:

public int formula2(int n)

השיטה תקבל כפרמטר מספר שלם n ותחזיר את סכום הסדרה הבאה:

1 – 2 + 3 – 4 + 5 - … + n

למשל, אם n = 5 השיטה תחזיר 3 שכן:

1 – 2 + 3 – 4 + 5 = 3

שאלה 19

עבור כל אחת מהשיטות בסעיפים הבאים רשמו מה השיטה עושה.

א.

```
public void f1(int[] a)
{
    f1(a, 0);
}

private void f1(int[] a, int i)
{
    if(i < a.length)
    {
        f1(a, i+1);
        System.out.print(a[i] + "\t");
    }
}
```

ב. השיטה מקבלת מערך דו ממדי **ריבועי**.

```
public boolean f2(int[][] a)
{
    return f2(a, 0);
}

private boolean f2(int[][] a, int i)
{
    if(i == a.length)
        return true;
    if(a[i][i] != 0)
        return false;
    return f2(a, i+1);
}
```

ג.

```
public void f3(String s)
{
    f3(s, 0, true);
}

private void f3(String s, int i, boolean b)
{
    if(i < s.length() / 2)
    {
        if(b)
            System.out.print(s.charAt(i));
        else
            System.out.print(s.charAt(s.length() - 1 - i));
        b = !b;
        f3(s, i+1, b);
    }
}
```

***

כתבו מחלקה בשם RecursiveArraysMethods. את התשובות לשאלות הבאות כתבו כשיטות בתוך מחלקה זו. אין להשתמש בלולאות כלל בשיטות. ניתן ורצוי להשתמש בשיטות עזר פרטיות.

שאלה 20

כתבו שיטה רקורסיבית שחתימתה:

public void printArray(int[] a)

השיטה תקבל כפרמטר מערך של מספרים שלמים a ותדפיס על המסך את איברי המערך מהראשון לאחרון בשורה אחת, כאשר האיברים יופרדו בטאבים (השתמשו בסימן "\t" כדי להדפיס טאב) אחד מהשני.

שאלה 21

כתבו שיטה רקורסיבית שחתימתה:

public int sumArray(int[] a)

השיטה תקבל כפרמטר מערך של מספרים שלמים a ותחזיר את סכום המספרים במערך.

שאלה 22

כתבו שיטה רקורסיבית שחתימתה:

public int minInArray(int[] arr)

השיטה תקבל כפרמטר מערך של מספרים שלמים, ותחזיר את האיבר המינימלי במערך.

שאלה 23

כתבו שיטה רקורסיבית שחתימתה:

public boolean isSorted(int[] a)

השיטה תקבל כפרמטר מערך של מספרים שלמים, ותחזיר true אם המערך ממויין בסדר עולה, ו- false אם לא.

למשל, עבור המערך {1, 3, 3, 5, 7} (משמאל לימין) השיטה תחזיר true ועבור המערך {4, 6, 7, 3, 8} השיטה תחזיר false.

שאלה 24

כתבו שיטה רקורסיבית שחתימתה:

public int countNums(double[] dArr, double num)

השיטה תקבל כפרמטרים מערך של מספרים ממשיים ומספר num. השיטה תחזיר כמה איברים במערך גדולים או שווים ל-num. למשל, אם dArr = {2, 5, 1, 8, 4, 5} ו- num = 5 השיטה תחזיר 3.

שאלה 25

כתבו שיטה רקורסיבית שחתימתה:

public int countOdd(int[] nums)

השיטה תקבל כפרמטר מערך של מספרים שלמים, ותחזיר כמה מספרים אי זוגיים יש במערך.
למשל, אם nums = {3, 5, 2, 7, 8} השיטה תחזיר 3.

שאלה 26

כתבו שיטה רקורסיבית שחתימתה:

public boolean isPalindrom(int[] a)

השיטה תקבל כפרמטר מערך של מספרים שלמים, ותחזיר true אם המערך הוא פאלינדרום, ו-false אם לא. מערך פאלינדרום הוא מערך שאפשר לקרוא אותו מההתחלה לסוף ומהסוף להתחלה באותה דרך. למשל, המערך הבא הוא פאלינדרום – {1, 5, 3, 7, 3, 5, 1}.

שאלה 27

כתבו שיטה רקורסיבית שחתימתה:

public int[] merge(int[] a, int[] b)

השיטה תקבל כפרמטרים שני מערכים **ממויינים בסדר עולה** (כלומר מהקטן לגדול). השיטה תחזיר מערך ממויין בסדר עולה שמכיל את כל איברי a ו-b. שימו לב, המערכים a ו-b לא בהכרח באותו הגודל.
למשל, אם a = {1, 5, 7, 9} ו- b = {2, 3, 7, 10, 11} השיטה תחזיר את המערך:
{1, 2, 3, 5, 7, 7, 9, 10, 11}

שאלה 28

כתבו שיטה שחתימתה:

public boolean equals(int[] a, int[] b)

השיטה תקבל שני מערכים של מספרים שלמים a, b ותחזיר true אם הם **שווים לחלוטין**. כלומר, מכילים את אותם איברים ובאותו סדר בדיוק. שימו לב שהמערכים לא חייבים להיות באותו גודל.
למשל, אם a = {1, 2, 3, 4}, b = {1, 2, 3, 4} השיטה תחזיר true.

אם a = {1, 2, 3, 4}, b = {3, 1, 2, 4} השיטה תחזיר false.

אם a = {1, 2, 3, 4}, b = {1, 2, 3} השיטה תחזיר false.

שאלה 29

כתבו שיטה רקורסיבית שחתימתה:

public int sumRow(int[][] a, int row)

השיטה תקבל כפרמטר מערך דו ממדי ואינדקס של שורה. השיטה תחזיר את סכום האיברים הנמצאים בשורה row.

שאלה 30

כתבו שיטה רקורסיבית שחתימתה:

public boolean allPositive(double[][] arr)

השיטה תקבל כפרמטר מערך דו ממדי, ותחזיר true אם המערך מכיל אך ורק מספרים חיוביים, ו-false אחרת.

שאלה 31

א. כתבו שיטה רקורסיבית שחתימתה:

public boolean isRowSorted(int[][] a. int row)

השיטה תקבל מערך דו ממדי ואינדקס של שורה. השיטה תחזיר true אם השורה מספר row ממויינת בסדר עולה ממש.

ב. כתבו שיטה רקורסיבית שחתימתה:

public boolean allRowsSorted(int[][] a)

השיטה תקבל מערך דו ממדי ותחזיר true אם כל אחת מהשורות במערך ממויינת בסדר עולה ממש.

שאלה 32

כתבו שיטה רקורסיבית שחתימתה:

public boolean special(int[][] a)

השיטה תקבל כפרמטר מערך דו ממדי, ותחזיר true אם כל האיברים בשורה ה-i קטנים ממש מכל האיברים בשורה ה-i+1. כלומר, כל האיברים בשורה ה-0 קטנים ממש מכל האיברים בשורה ה-1. כל האיברים בשורה ה-1 קטנים ממש מכל האיברים בשורה ה-2 וכן הלאה.

שאלה 33

נגדיר – מערך דו ממדי ריבועי יקרא **מערך משופע** אם הוא מקיים את התנאים הבאים:

4. כל איברי האלכסון הראשי שווים ל-0.
5. כל האיברים שמעל לאלכסון הראשי גדולים מ-0.
6. כל האיברים שמתחת לאלכסון הראשי קטנים מ-0.

להלן שני מערכים לדוגמא:

	0	1	2	3	4
0	0	7	5	9	3
1	-1	0	4	7	6
2	-3	-2	0	2	3
3	-7	-9	-2	0	4
4	-1	-8	-5	-8	0

	0	1	2	3	4
0	0	7	5	9	3
1	-1	0	4	7	6
2	-3	-2	0	2	3
3	-7	-9	-2	0	4
4	-1	-8	-5	-8	0

המערך מצד ימין הוא מערך שיפועי, והמערך השמאלי אינו שיפועי.

כתבו שיטה רקורסיבית שחתימתה:

public boolean isSloping(int[][] a)

השיטה תקבל כפרמטר מערך דו מימדי ריבועי ותחזיר true אם המערך הוא שיפועי לפי ההגדרה שלעיל, ו-false אם לא.

שאלה 34

כתבו שיטה רקורסיבית שחתימתה:

public int howMany(String s, char c)

השיטה תקבל כפרמטרים מחרוזת s ותו c. השיטה תחזיר כמה פעמים מופיע התו c במחרוזת.

למשל, אם s = "abcadca", אזי, אם c = 'a' השיטה תחזיר 3. אם c = 'b' השיטה תחזיר 1. אם c = 'f' השיטה תחזיר 0.

שאלה 35

כתבו שיטה רקורסיבית שחתימתה:

public boolean notExist(String s, char c)

השיטה תקבל כפרמטרים מחרוזת s ותו c. השיטה תחזיר true אם התו c לא נמצא כלל במחרוזת, ו-false אחרת. למשל, אם s = "abcd", אזי אם c = 'a' השיטה תחזיר false ואם c = 'f' השיטה תחזיר true.

שאלה 36

כתבו שיטה רקורסיבית שחתימתה:

public String reverse(String s)

השיטה תקבל כפרמטר מחרוזת, ותחזיר מחרוזת חדשה המכילה את אותם תווים בסדר הפוך.
למשל, אם s = "abcde" השיטה תחזיר את המחרוזת "edcba".

שאלה 37

נגדיר, מחרוזת t היא תת מחרוזת של s אם כל התווים של t נמצאים ברצף ב-s.
למשל, אם s = "abcdef" אזי המחרוזות (בין היתר) "abc", "bcde", "f" הן תתי מחרוזות של s, אולם המחרוזת "acf" אינה תת מחרוזת.
כתבו שיטה רקורסיבית שחתימתה:

public boolean isSubstring(String s, String t)

השיטה תקבל שתי מחרוזות, s, t ותחזיר true אם t היא תת מחרוזת של s, ו-false אם לא.

השאלות הבאות עוסקות ב-Backtracking (עקיבה לאחור)

שאלה 38

ענו על השאלות הבאות. השתדלו לנמק היטב בצורה ברורה ולא מעורפלת.
א. מהי שיטת backtracking?
ב. מהו "מרחב מצבים"?
ג. מדוע רקורסיה משמשת באלגוריתם backtracking? מה יתרונה של הרקורסיה?

שאלה 39

בכל אחד מהסעיפים הבאים נתונה בעיה שניתן לפתור אותה באמצעות backtracking. נסחו אלגוריתם עבור כל בעיה. אין צורך לכתוב קוד אלא לנסח את האלגוריתם בעברית.
א. **בעיית מסעות הפרש** - במשחק השחמט ישנו לוח של 8X8 משבצות. כלי המשחק יכולים לנוע בכיוונים שונים, בהתאם לסוג הכלי. הפרש (knight) במשחק נע בצורה הבאה – שתי משבצות בכיוון אחד, ומשבצת אחת בכיוון המאונך. למשל, שתי משבצות קדימה ומשבצת אחת שמאלה, או שתי משבצות שמאלה ומשבצת אחת למטה.
למשל, האיור הבא מראה את לוח השחמט ובו פרש בנקודה (4, 5) (שורה 4 עמודה 5) ואת המשבצות אליהן יכול הפרש להגיע מנקודה זו (מסומנות במספרים 1 עד 8).
בהנתן לוח שחמט ריק, ופרש המוצב בשורה x ובעמודה y כלשהי, נסחו אלגוריתם המחליט האם הפרש יכול לבקר בכל אחת ממשבצות הלוח, כאשר הוא לא עובר באותה משבצת פעמיים?

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

ב. **בעיית שמונה המלכות** – בלוח השחמט, המלכה היא כלי משחק שיכול לנוע בכל כיוון (ימין, שמאל, למעלה, למטה ובאלכסונים) ולכל מרחק אפשרי. הבעיה – נסחו אלגוריתם המציב על לוח שחמט ריק שמונה מלכות, כך שאף אחת מהן לא מאויימת.

ג. **סודוקו** – משחק הסודוקו בנוי מלוח של 9X9 משבצות, ומחולק לקוביות של 3X3. צריך לשבץ את המספרים 1-9 במשבצות הלוח, כאשר כל ספרה תופיע פעם אחת בלבד בכל שורה, עמודה וקוביה. הבעיה – בהנתן לוח סודוקו מלא חלקית (כלומר, רק בחלק מהמשבצות קיימים מספרים, וכל שאר המשבצות ריקות) נסחו אלגוריתם הפותר את הלוח, כלומר מציב במשבצות הריקות את המספרים באופן חוקי.

ד. **יציאה ממבוך** – נתון מבוך שמיוצג באמצעות לוח דו מימדי. במבוך ישנם קירות, כלומר משבצות שאי אפשר לעבור אותן. הבעיה – בהנתן נקודת כניסה למבוך, ונקודת יציאה, נסחו אלגוריתם שמוצא דרך יציאה מהמבוך.

***

כתבו מחלקה בשם Backtracking. את התשובות לכל השאלות הבאות כתבו במחלקה זו. בכל השאלות אין להשתמש בלולאות כלל. בכל השאלות מותר ורצוי להשתמש בשיטות עזר פרטיות (רקורסיביות גם הן).

שאלה 40

נתון מערך חד ממדי המלא במספרים שלמים חיוביים. מכל תא במערך ניתן לנוע ימינה לפי ספרת האחדות שבתא, או לפי ספרת העשרות. למשל, בהנתן המערך הבא:

34	59	74	12	15	17
0	1	2	3	4	5

בתא מספר 0, למשל, נמצא המספר 34. לכן מתא זה אפשר להתקדם 4 צעדים ימינה ולהגיע לתא מספר 4 (לפי ספרת האחדות) **או** שלושה צעדים ימינה ולהגיע לתא מספר 3 (לפי ספרת העשרות).

א. כתבו שיטה רקורסיבית שחתימתה:

```
public boolean pathExist(int[] a)
```

השיטה תקבל כפרמטר מערך חד ממדי כמתואר בשאלה. השיטה תחזיר true אם קיים מסלול במערך שמתחיל בתא מספר 0 ומסתיים בתא האחרון, ומורכב מתזוזות ימינה כמתואר. למשל, במערך הנתון יש מסלול – מתא 0 ניתן להתקדם שלושה צעדים ימינה לפי ספרת העשרות, ולהגיע לתא מספר 3. משם אפשר להתקדם 2 צעדים ימינה לפי ספרת האחדות, ולהגיע לתא האחרון. השיטה תחזיר false אם לא קיים מסלול כזה במערך.

ב. כתבו שיטה רקורסיבית שחתימתה:

```
public int howManyPaths(int[] a)
```

השיטה תקבל מערך חד ממדי כמתואר בשאלה. השיטה תחזיר את מספר המסלולים הקיימים מהתא הראשון ועד התא האחרון. למשל, במערך הנתון ישנם שלושה מסלולים כאלה – הראשון כמתואר בסעיף א'. השני – מתא 0 זזים לתא 3 לפי ספרת העשרות. משם משם זזים לתא 4 לפי ספרת האחדות ומשם לתא 5 לפי ספרת העשרות. תוכלו למצוא בעצמכם גם את המסלול השלישי.

שאלה 41

כתבו שיטה רקורסיבית שחתימתה:

public int minOperations(int x, int y)

השיטה תקבל שני פרמטרים, x ו-y. השיטה תחזיר את מספר פעולות החשבון המינימלי הדרוש להגיע מ-x ל-y, כאשר פעולות החשבון המותרות הן הוספת 1 ל-x או הכפלת x פי 2.

למשל, אם x=10 ו-y=20 השיטה תחזיר 1, כיוון שאפשר להגיע מ-10 ל-20 ע"י הכפלה פי 2.

אם x = 10 ו-y=22 השיטה תחזיר 2, כיוון שאפשר להגיע מ-10 ל-22 ע"י הוספת 1 ל-10 וקבלת 11, ואז הכפלת 11 פי 2 לקבלת 22.

שאלה 42

כתבו שיטה רקורסיבית שחתימתה:

public void printOptions(int num)

השיטה תדפיס על המסך את כל האופציות לקבלת המספר num ע"י פעולות חיבור של 1 ושל 2.

למשל, אם num = 4 השיטה תדפיס את הפלט הבא:

```
1 1 1 1
1 1 2
1 2 1
2 1 1
2 2
```

שאלה 43

כתבו שיטה שחתימתה:

public boolean sameMult(int[] a)

השיטה תקבל כפרמטר מערך a מלא במספרים חיוביים. השיטה תחזיר true אם ניתן לחלק את איברי המערך לשתי קבוצות, כך שמכפלת האיברים בקבוצה הראשונה שווה למכפלת האיברים בקבוצה השניה. השיטה תחזיר false אם אי אפשר לחלק את המערך כאמור.

למשל, עבור המערך הבא:

6	2	5	5	3
0	1	2	3	4

השיטה תחזיר true שכן ניתן לחלק את איברי המערך לשתי קבוצות שמכפלתם זהה:

6 * 5 = 2 * 5 * 3

שאלה 44

כתבו שיטה שחתימתה:

public boolean cover(int[] a, int m)

השיטה תקבל כפרמטרים מערך של מספרים שלמים a ומספר m. השיטה תחזיר true אם קיים במערך אוסף של איברים שסכומם הוא בדיוק m. השיטה תחזיר false אם לא קיים סכום כזה.

למשל, עבור המערך הבא:

7	2	15	4	10	17
0	1	2	3	4	5

אם m=7 השיטה תחזיר true

אם m=6 השיטה תחזיר true.

אם m=100 השיטה תחזיר false.

שאלה 45

נגדיר – **ג'וקר** במחרוזת (הסימן *) מייצג תת מחרוזת של אפס או יותר תווים רצופים. בהנתן שתי מחרוזות, s1 ו-s2, נאמר ש-s1 זהה לתבנית s2 אם ניתן להתאים כל תו ב-s1 לתו באותו מקום ב-s2, או שיש ב-s2 ג'וקר ש"משלים" את התווים החסרים. למשל, המחרוזות האלו זהות תבנית:

s1 = "abcde" s2 = "ab*e"
s1 = "abcde" s2 = "*"
s1 = "abcde" s2 = "*b*"

המחרוזות הבאות אינן זהות תבנית:

s1 = "abcde" s2 = "a*k"
s1 = "abcde" s2 = "ab*bcde"

כתבו שיטה רקורסיבית שחתימתה:

```
public boolean isSamePattern(String s1, String s2)
```

השיטה תקבל כפרמטרים שתי מחרוזות כמתואר לעיל ותחזיר true אם המחרוזות זהות תבנית ו-false אם לא.
שימו לב – ניתן להניח שהג'וקר מופיע רק במחרוזת s2, ושלא מופיעים שני סימני ג'וקר רצופים.

שאלה 46

בהנתן מערך דו ממדי שמורכב מהמספרים 0 ו-1, נגדיר **כתם** במערך כרצף של תאים המכילים את הספרה 1 ונמצאים בשכנות. למשל, בהנתן המערך הבא:

	0	1	2	3	4
0	1	0	0	0	1
1	1	0	1	1	0
2	0	0	0	1	1
3	1	0	0	1	0

במערך ישנם שלושה כתמים, מסומנים בצבעים שונים. שימו לב שלכל משבצת במערך יכולים להיות עד שמונה שכנים.
כתבו שיטה רקורסיבית שחתימתה:

```
public int stainSize(int[][] a, int row, int col)
```

השיטה תקבל כפרמטרים מערך דו מימדי המורכב מהמספרים 0 ו-1, ואינדקסים של שורה ועמודה.
השיטה תחזיר את גודל הכתם (מספר התאים שהוא מכיל) שמכיל את התא row, col. למשל, עבור המערך הנתון, אם row = 0, col = 0 השיטה תחזיר 2
אם row = 3, col = 0 השיטה תחזיר 1
אם row = 2, col = 3 השיטה תחזיר 6

שאלה 47

נתון מערך בגודל n ומספר שלם חיובי max. אנחנו רוצים לשבץ במערך n מספרים מתוך התחום 1 עד max כך שהמספרים יהיו מסודרים בסדר לא יורד.
למשל, אם n = 3 ו-max = 2 (כלומר ניתן לשבץ את המספרים 1 או 2). לכן יש ארבע אפשרויות לסידור המספרים –

{1, 1, 1}, {1, 1, 2}, {1, 2, 2, }, {2, 2, 2}

כתבו שיטה רקורסיבית שחתימתה:

```
public int howManySorted(int n, int max)
```

השיטה תקבל כפרמטרים גודל מערך n ומספר מקסימלי max. השיטה תחזיר את מספר האפשרויות שניתן לשבץ n מספרים מתוך התחום [1, max] במערך, כמתואר.

רשימות מקושרות

קובץ שאלות זה מכיל תרגילים בנושא רשימות מקושרות בג'אווה. נושא הרשימות הוא נושא טכני בעיקרו, ולכן חשוב להריץ "על נייר" את שאלות הסימולציה שבתחילת הקובץ, ולוודא שאתם מבינים איך עובד מנגנון ההצבעה ברשימות. כמו כן, חשוב מאוד לפתור את שאלות כתיבת הקוד כדי לתרגל את טכניקת התכנות ברשימות.

שאלה 1

ענו על השאלות הבאות. השתלו לנמק היטב ובצורה לא מעורפלת, וודאו שאתם מבינים את המושגים:

א. מהי רשימה מקושרת חד סטרית?

ב. מה ההבדל בין רשימה מקושרת למערך?

ג. מהו איבר ברשימה מקושרת?

ד. מהו ראש הרשימה?

ה. מהו מושג ההתייחסות העצמית (self reference) ואיך הוא מיושם ברשימה מקושרת?

***

נתונה הגדרת המחלקה IntNode המייצגת איבר ברשימה מקושרת השומרת מספרים שלמים מטיפוס int. נשתמש במחלקה זו עבור כל השאלות הבאות.

```
public class IntNode {
    private int vlaue;
    private IntNode next;

    public IntNode(int v, IntNode n) {
        value = v;
        next = n;
    }

    public int getValue() {
        return value;
    }
    public void setValue(int v) {
        value = v;
    }
    public IntNode getNext() {
        return next;
    }
    public void setNext(IntNode n) {
        next = n;
    }
}
```


shay.tavor@gmail.com

שאלה 2

נתון ה-main הבא, העושה שימוש במחלקה IntNode שלעיל. ציירו את הרשימה המתקבלת אחרי סיום פעולת ה-main:

```
public class Tester
{
    public static void main(String[] args)
    {
        IntNode list = null;
        list = new IntNode(5, null);
        list = new IntNode(8, list);
        list = new IntNode(3, list);

        IntNode p = new IntNode(9, list.getNext());
        list.setNext(p);

        for(p = list; p.getNext() != null; p = p.getNext());
        p.setNext(new IntNode(7, null));
    }
}
```

שאלה 3

נתון משתנה בשם t מסוג IntNode שמצביע על הרשימה הבאה:

t --> 4 --> 7 --> 1 --> null

כלומר, הרשימה מכילה שלושה איברים: האיבר הראשון הוא 4, השני הוא 7 והשלישי הוא 1.

נתון ה-main הבא המתייחס לרשימה שלעיל. עבור כל שורה ב-main רשמו האם השורה תקינה, מה הפלט (אם יש) או האם השורה גורמת לשגיאת ריצה (רשמו מדוע):

```
public class Tester
{
    public static void main(String[] args)
    {
      System.out.println(t.getValue());
      System.out.println(t.getNext().getValue());
      System.out.println(t.getNext().getNext().getValue());
      System.out.println(t.getNext().getNext().getNext().getValue());
      t.getNext().setNext(null);
      IntNode p = t.getNext().getNext();
      System.out.println(p.getValue());
    }
}
```

שאלה 4

נתון משתנה בשם t מסוג IntNode שמצביע על הרשימה הבאה:

t --> 4 --> 7 --> 1 --> 6 --> null

בכל אחד מהסעיפים הבאים נתונה שיטה המקבלת כפרמטר מצביע לתחילת רשימה מקושרת. עבור כל סעיף, ציירו את הרשימה ש-t מצביע עליה, אחרי הקריאה לשיטה עם t כפרמטר.

א.

```
public void f(IntNode t) {
    IntNode p = t;

    while(p.getNext() != null)
        p = p.getNext();

    int temp = t.getValue();
    t.setValue(p.getValue());
    p.setValue(temp);
}
```

ב.

```
public void g(IntNode t) {
    IntNode p = t;
    t = t.getNext();
    p.setNext(null);
    IntNode q = t;
    while(q.getNext() != null)
        q = q.getNext();
    g.setNext(p);
}
```

ג.

```
public void h(IntNode t) {
    for(IntNode p = t; p != null; p = p.getNext().getNext())
        p.setNext(new IntNode(p.getValue(), p.getNext());
}
```

***

נתונה המחלקה IntList המייצגת רשימה מקושרת של מספרים שלמים מטיפוס int:

```
public class IntList {
    private IntNode head;

    public IntList() {
        head = null;
    }
}
```

המחלקה מכילה תכונה אחת מסוג IntNode שמייצגת את המצביע לראש הרשימה.
את התשובות לכל השאלות הבאות רשמו כשיטות בתוך המחלקה IntList.

תזכורת חשובה – בכל שאלה, אלא אם כן מצויין במפורש אחרת, התייחסו לשלושה מקרי הקצה הנפוצים:
1. מה יקרה כאשר הרשימה ריקה? (כלומר, כאשר head שווה ל-null).
2. מה יקרה כאשר מטפלים באיבר הראשון ברשימה?
3. מה יקרה כאשר מטפלים באיבר האחרון ברשימה?
בהחלט יכול להיות שישנם מקרי קצה אחרים שיש לטפל בהם, אולם שלושת המקרים שלעיל נפוצים מאוד ולכן כדאי לזכור להתייחס אליהם בכל שיטה.

שאלה 5

כתבו שיטה שחתימתה:

public String toString()

השיטה תחזיר את הרשימה כמחרוזת בפורמט הבא – איברי הרשימה יודפסו בשורה אחת, מהראשון לאחרון, כאשר בין כל שני איברים יופיע הסימן <--. אחרי האיבר האחרון תופיע המילה null. למשל, אם הרשימה מכילה שלושה איברים – האיבר הראשון הוא 7, השני הוא 4 והשלישי הוא 9, השיטה תחזיר את המחרוזת:

7-->4-->9-->null

שאלה 6

כתבו שיטה שחתימתה:

public void addToStart(int x)

השיטה תוסיף לתחילת הרשימה איבר חדש שערכו הוא x. למשל, אם הרשימה היתה:

5-->3-->8 -->null

וערכו של x היה 4, אזי אחרי פעולת השיטה הרשימה תיראה כך:

4-->5-->3-->8 -->null

שאלה 7

כתבו שיטה שחתימתה:

public void addToEnd(int x)

השיטה תוסיף לסוף הרשימה (כלומר, אחרי האיבר האחרון) איבר חדש שערכו הוא x. למשל, אם הרשימה היתה:

5-->3-->8 -->null

וערכו של x היה 4, אזי אחרי פעולת השיטה הרשימה תיראה כך:

5-->3-->8 -->4-->null

שאלה 8

כתבו שיטה שחתימתה:

public void addIfNotExist(int x)

השיטה תוסיף לתחילת הרשימה איבר חדש שערכו הוא x בתנאי שלא קיים ברשימה איבר נוסף בעל אותו הערך.

שאלה 9

כתבו שיטה שחתימתה:

public int length()

השיטה תחזיר את אורך הרשימה, כלומר את מספר האיברים ברשימה. למשל, אם הרשימה היתה:

5-->3-->8 -->null

השיטה תחזיר 3. שימו לב שאורכה של רשימה ריקה הוא 0.

שאלה 10

כתבו שיטה שחתימתה:

public int howMany(int x)

השיטה תחזיר כמה פעמים מופיע איבר שערכו x ברשימה.

שאלה 11

כתבו שיטה שחתימתה:

public int maxInList()

השיטה תחזיר את ערכו של האיבר המקסימלי ברשימה. אם הרשימה ריקה השיטה תחזיר 0.

שאלה 12

כתבו שיטה שחתימתה:

public int howManyBigger(int x)

השיטה תקבל כפרמטר מספר שלם x ותחזיר כמה איברים ברשימה מכילים ערך שגדול מ-x.

שאלה 13

א. כתבו שיטה שחתימתה:

public void remove(int x)

השיטה תקבל כפרמטר מספר שלם x ותסיר מהרשימה את האיבר שמכיל את הערך x. אם יש כמה איברים שמכילים את הערך x השיטה תסיר את הראשון. אם לא קיים ברשימה איבר שערכו x השיטה תשאיר את הרשימה כמו שהיא היתה.

למשל, אם הרשימה היתה:

5-->3-->8 -->4-->8-->null

ו-x=8, בסיום השיטה הרשימה תיראה כך:

5-->3-->4-->8-->null

ב. כתבו שיטה שחתימתה:

public void removeAll(int x)

השיטה תקבל כפרמטר מספר שלם x ותסיר מהרשימה **את כל** האיברים שמכילים את הערך x.
למשל, אם הרשימה היתה כמתואר בסעיף א', ו- x=8, לאחר פעולת השיטה הרשימה תיראה כך:

5-->3-->4-->null

שאלה 14

כתבו שיטה שחתימתה:

public boolean allDifferent()

השיטה תחזיר true אם כל ערך ברשימה מופיע פעם אחת בלבד (אין חזרות) ו-false אם לא.

שאלה 15

כתבו שיטה שחתימתה:

public void removeDuplications()

השיטה תסיר מהרשימה מופעים כפולים של איברים, ותשאיר את הרשימה במצב בו כל איבר מופיע פעם אחת בלבד. למשל, אם הרשימה היתה:

1-->7-->1-->9-->9-->1-->3-->null

בסיום השיטה הרשימה תיראה כך:

1-->7-->9-->3-->null

שאלה 16

כתבו שיטה שחתימתה:

public void reverse()

השיטה תהפוך את סדר איברי הרשימה. כלומר, האיבר הראשון יהפוך להיות האיבר האחרון, האיבר השני יהפוך להיות האיבר הלפני אחרון, וכן הלאה. השיטה צריכה לעבוד ביעילות זמן ריצה של O(n) כאשר n הוא אורך הרשימה, ובסיבוכיות מקום של O(1) (כלומר, אין ליצור רשימה חדשה).
למשל, אם הרשימה היתה:

1-->7-->2-->9-->3-->null

בסיום השיטה הרשימה תיראה כך:

3-->9-->2-->7-->1-->null

שאלה 17

א. נגדיר – רצף ברשימה הוא מספר איברים צמודים ברשימה שמכילים את אותו ערך. למשל ברשימה הבאה:

5-->5-->8-->8-->8-->2-->null

ישנם שלושה רצפים – שני האיברים הראשונים מכילים את הערך 5, שלושת האיברים הבאים מכילים את הערך 8 והאיבר האחרון מכיל את הערך 2.
כתבו שיטה שחתימתה:

public void longestSequence()

השיטה תחזיר את אורכו של הרצף הארוך ביותר ברשימה. למשל, ברשימה שלעיל, השיטה תחזיר 3, כיוון שזהו הרצף הארוך ביותר.
ב. כתבו שיטה שחתימתה:

public void removeLongest()

השיטה תסיר מהרשימה את הרצף הארוך ביותר. למשל, עבור הרשימה מסעיף א', אחרי פעולת השיטה הרשימה תיראה כך:

5-->5-->2-->null

שאלה 18

כתבו שיטה שחתימתה:

public void concat(IntList other)

השיטה תקבל כפרמטר אובייקט מסוג IntList שמייצג רשימה. השיטה תחבר את הרשימה שמיוצגת על ידי other לסוף הרשימה המקורית. למשל, אם הרשימה המקורית היתה:

3-->7-->2-->null

והרשימה other היתה:

8-->1-->7-->3-->null

בסיום השיטה הרשימה המקורית תיראה כך:

3-->7-->2-->8-->1-->7-->3-->null

שאלה 19

כתבו שיטה שחתימתה:

public boolean isSubList(IntList other)

השיטה תקבל כפרמטר אובייקט מסוג IntList שמייצג רשימה. השיטה תחזיר true אם כל איברי הרשימה other נמצאים ברשימה המקורית, ו-false אם לא.

למשל, אם הרשימה המקורית היתה:

4-->7-->2-->9-->3-->null

והרשימה other היתה:

3-->7-->4-->null

השיטה תחזיר true.

שאלה 20

כתבו שיטה שחתימתה:

public IntList intersect(IntList other)

השיטה תקבל כפרמטר אובייקט מסוג IntList שמייצג רשימה. השיטה תחזיר אובייקט חדש מסוג IntList שמכיל את האיברים שנמצאים בשתי הרשימות. ניתן להניח שבכל רשימה האיברים לא חוזרים אחד על השני.

למשל, אם הרשימה המקורית היתה:

4-->7-->2-->9-->3-->null

והרשימה other היתה:

3-->7-->1-->9-->null

השיטה תחזיר את הרשימה:

7-->9-->3-->null

***

השאלות הבאות מתרגלות רקורסיה ברשימות. בשאלות אלו אין להשתמש בלולאות כלל. מותר להגדיר שיטות עזר פרטיות.

שאלה 21

כתבו שיטה רקורסיבית שחתימתה:

public void printReverse()

השיטה תדפיס את איברי הרשימה מהסוף להתחלה.

שאלה 22

כתבו שיטה רקורסיבית שחתימתה:

public int listSum()

השיטה תחזיר את סכום הערכים שבאיברי הרשימה.

שאלה 23

כתבו שיטה רקורסיבית שחתימתה:

public int howManySmaller(int x)

השיטה תקבל כפרמטר מספר שלם x ותחזיר כמה איברים ברשימה מכילים ערך שקטן מ-x.

שאלה 24

כתבו שיטה רקורסיבית שחתימתה:

public boolean isSorted()

השיטה תחזיר true אם הרשימה ממויינת בסדר עולה ממש, כלומר כל איבר ברשימה קטן ממש מהאיבר שבא אחריו. השיטה תחזיר false אם הרשימה אינה ממויינת כמתואר.

שאלה 25

כתבו שיטה רקורסיבית שחתימתה:

public IntList copyBigger(int x)

השיטה תקבל כפרמטר מספר שלם x ותחזיר רשימה חדשה שמכילה את כל האיברים מהרשימה המקורית שגדולים או שווים ל-x. למשל, אם הרשימה המקורית היתה:

5-->2-->9-->1-->null

ו- x = 3, השיטה תחזיר את הרשימה הבאה:

5-->9-->null

שאלה 26

כתבו שיטה רקורסיבית שחתימתה:

public void removeEven()

השיטה תסיר מהרשימה את כל האיברים שערכם זוגי. למשל, אם הרשימה המקורית היתה:

5-->2-->9-->4-->null

אחרי פעולת השיטה, הרשימה תיראה כך:

2-->4-->null