

מבוא לתכנות מונחה עצמים מטלה 1

הנחיות כלליות:

1. את המטלה עושים בזוגות, יש להגיש את כל המטלות בזמן לפי הנחיות, על כל איחור לא מוצדק תהיה הורדת ניקוד.
2. המטלות תיבדקנה באופן אוטומטי באספקטים של "אין לבצע שום העתקה של קודים בין קבוצות שונות, מותר לעשות שימוש בקוד פתוח, אבל חובה לציין זאת בפירוש ולהביא את המקור המדויק. למען הסר ספק: שימוש בקוד פתוח (או כל קוד זמין ברשת) שלא יצוין מקור הקוד יחשב כהעתקה!
3. יש לתעד את כל המתודות של הפרויקט שלכם בעזרת javadoc.
4. צריך להגיש קובץ ZIP שם הקובץ - מספר זהות ראשון מקו תחתון מספר זהות שני. יש לקבץ קבצי JAVA בלבד.

חלק א שימוש בממשקים

בחלק זה נזקוק במחלקות Point, Triangle, Rectangle, PointContainer של קורס מבוא לחישוב.

נגדיר ממשק בשם Drawable:

```
public interface Drawable {
    public boolean equals(Drawable d);
    public boolean contains(Point p);
    public double perimeter();
    public double area();
    public void translate(Point p);
}
```

נגדיר מחדש מחלקות Point, Triangle, Rectangle כך שהן תממשנה את הממשק. נכתוב מחלקה ShapeContainer כך שהיא תכיל רק מערך אחד של צורות מטיפוס Drawable ובהתאם לזה נשנה את השיטות שלה. הדרכה: צריך להבחין בין שיטות שבהן צריך הקומפיילר לדעת את הטיפוס האמתי של צורה ושיטות שבהם מספיק להשתמש בטיפוס של Drawable. בדרך כלל ביצירת אובייקט חדש באמצעות בנאי מעתיק צריך להשתמש בטיפוס אמתי של האובייקט.

תיאור של מחלקת ShapeContainer

public ShapeContainer()

יצירה של אובייקט חדש שאינו מכיל צורות.

public ShapeContainer(ShapeContainer other)

"העתקה העמוקה" - אוסף הצורות של האובייקט הקיים ישתכפל באובייקט החדש כסדרן.

עליכם לאחסן את אוסף הצורות באמצעות מערך אחד. למחלקה שני קבועים ציבוריים ומוגדרת בה פעולת גדילה (שיטה פרטית):

```
public static final int INIT_SIZE=10;
public static final int RESIZE=10;
```

הקבוע INIT_SIZE הוא גודל המערכים הראשוני שיווצרו בעת יצירת אובייקט ריק. הקבוע RESIZE הוא מידת הגדילה של המערכים במידה ואין בהם יותר מקום לקלוט צורות נוספות, פעולת הגדילה תתבצע רק כאשר יש צורך להוסיף צורה ואין מקום.

```
public int size() // return the number of stored shapes in the container
```

```
public int T_size() // return the number of stored triangles in the container
public int R_size() // return the number of stored rectangles in the container
```

```
public void add (Drawable d); // add the given object to the container
```

שימו לב: פעולת ההוספה אינה יוצרת עותק חדש של האיבר המוסף!

```
public void remove(Point p) // remove triangles & rectangles containing p
public Triangle T_at(int i) // return a new copy of the triangle number i
public Rectangle R_at(int i) // return a new copy of the rectangle number i
public double sumArea() // return the sum of the
// areas of all the triangles &
// rectangles
```

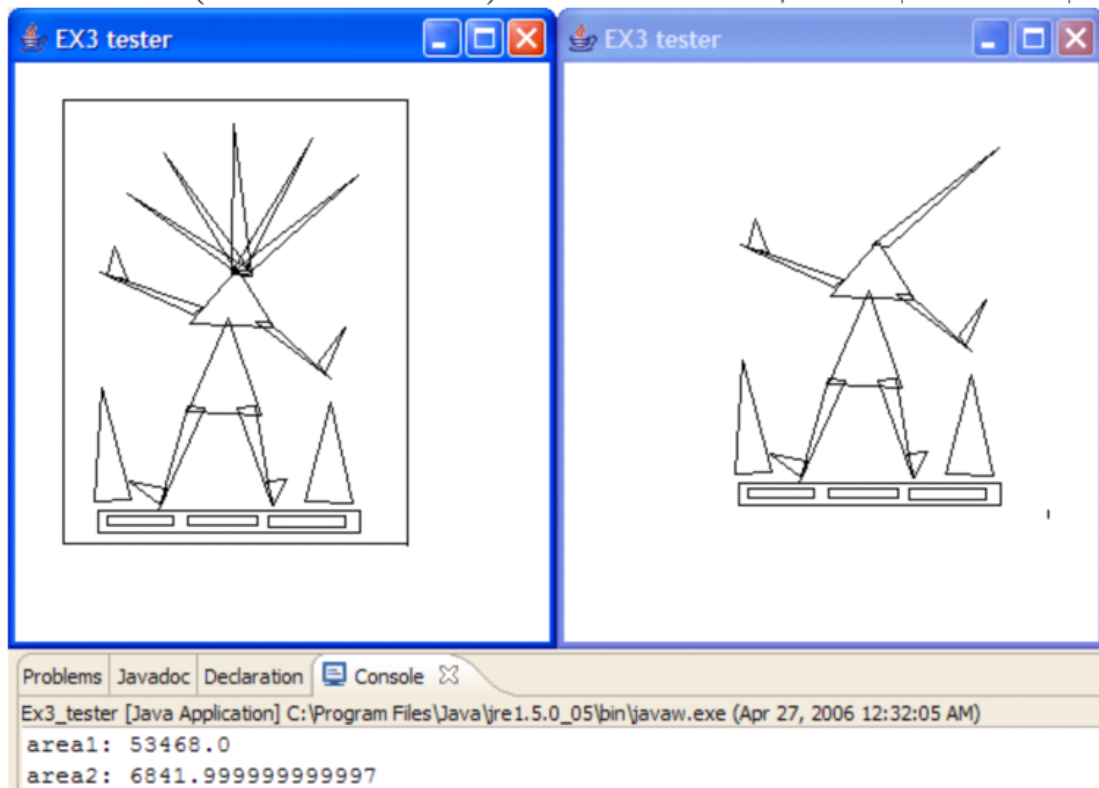
```
public void translate(Point p) //Translates (mutator) all the shapes by a Point
```

ראו הגדרת translate במחלקה נקודה.

```
public void minMaxPerimeter(int num)// calculates and prints min and max perimeter
// of the shapes (triangles & rectangles)
```

הערות:

שימו לב למקרי קצה, כלומר הוספה על גבול גודל המערך, ביצוע פעולות עם קלט לא תקין וכיו"ב.
לא נדרש לבצע פעולת "הקטנה" במידה ונמחקות צורות מהאוסף.
לא ניתן להניח כי הפרמטרים המתקבלים תקינים / מאותחלים.
חתימות השיטות חייבות להיות זהות לחלוטין לחתימות המתוארות במסמך זה.
יש לבדוק את התכנית בעזרת קובץ Ex1_Tester.java.
בדקו שתוכנית הבדיקה שלכם אכן מציירת את החלונות הבאים (ומחשבת את השטחים כאמור).



```
size1: 23
size2: 18
number of comparision1: 31
max perimeter1: 860.0
```

```

min perimeter1: 8.0
number of comparision2: 25
max perimeter2: 310.0
min perimeter2: 8.0

```

חלק ב. שימוש במבני נתונים המוגדרות ב-java

בחלק זה נכתוב תכנית בשם **TestBet.java** שקוראת קובץ טקסט וסופרת כמה פעמים מופיע כל מילה בקובץ. יש להעביר את קובץ הטקסט לאוסף מילים בודדים (השתמשו במחלקת **StringTokenizer**) ולשמור מונה לכל מילה כדי לדעת כמה פעמים היא מופיעה. בזמן הכנסת המילה קיימת עדכוננו את מונה המילה. את הסעיף יש לעשות בעזרת **מבנה נתונים מתאים**. בסוף הריצה על התוכנית יש להדפיס דוח שמכיל את הפרטים הבאים:

- מספר המילים הכללי בקובץ.
- מספר המילים השונות בקובץ.
- המילה שחוזרת הכי הרבה פעמים ואת מספר הפעמים שהיא חוזרת.
- את המילה הארוכה ביותר בקובץ.

לדוגמה, בטקסט: **To be, or not to be: that is the question**

מספר המילים הוא 10, מספר המילים השונות הוא 9, המילה שחוזרת הכי הרבה פעמים: **be** מופיעה פעמיים,

את מספר מילים חוזרות: מילה אחת.

הערה חשובה: כדי לקבל את התוצאה זהות תשתמשו בסימנים הבאים **"-/(. ; , "**

להפרדת מילים ב-**StringTokenizer**.

אחר שמימשתם את הטבלה כתבו פונקציית **main** לבדיקה עצמית. עשו זאת על הקבצים הבאים המצורפים למטלה:

tex1.txt, tex3.txt

נספח:

הפונקציה הבאה קוראת מקובץ טקסט. עליכם להשלים את הפונקציה כדי לקבל את הפרטים הנדרשים על הטקסט.

```

public static void readFile (String fileName){
    try { // try read from the file
        FileReader fr = new FileReader(fileName);
        BufferedReader br = new BufferedReader(fr);
        String str = br.readLine();
        System.out.println(0+" "+str);
        for(int i=1; str!=null; i=i+1) {
            str = br.readLine();
            if (str != null){
                System.out.println(i+" "+str);
            }
        }
        br.close();
    }
    catch(IOException ex) {
        System.out.print("Error reading file\n" + ex);
        System.exit(2);
    }
}

```

תוצאות של הרצה:

1) "text3.txt"

The Number of Differnt Words: 26764
The Number of Words: 493550
The Most Frequent Word: (the), frequenncy: 34310
The Longest Word:
(Charla_sobre_DRM_a_Microsoft_Research)

2) "text1.txt"

The Number of Different Words: 2180
The Number of Words: 26185
The Most Frequent Word: (the), frequenncy: 2140
The Longest Word: (ConcurrentModificationException)