מעבדה 1. נושא: עבודה עם מערכים

תאריך הגשה: 01/11/2022 בשעה 23:00 (בזוגות)

יש לקרוא היטב לפני תחילת העבודה !

מבוא:

מערך הוא מבנה נתונים בסיסי השומר את הנתונים ומאפשר גישה מהירה אל האיברים שבו.

במעבדה זו נציע שימוש במערך לצורך החזקת נקודות במישור (בעלות שתי קואורדינטות).

מטרות:

1) מימוש רשימות של נקודות בעזרת מערך, כולל פעולות המאפשרות לנוע במערך זה (חלק א').

2) שימוש ב Java tokenizer, לצורך קריאת רצפים של תווים מהקלט (חלק ב').

חלק א'

תיאור:

1. נגדיר רשימה סדורה של נקודות: נקודה במישור מאופיינת ע"י שתי קואורדינטות:

P = (x,y)

2. שימוש בקואורדינטות הינו רב-תחומי: בגרפיקה ממוחשבת, מודלים חישוביים להצגת קווים, עקומות וכו'.
3. הממשק מאפשר הוספת נקודות, שאילתות שונות לגבי מצב המבנה, ותזוזה בתוך המבנה.
4. לצורך התזוזה בתוך המבנה נגדיר **סמן (cursor)**
5. להלן הגדרת הממשק PointList:

5.1 איברי הרשימה:

כל איבר יהיה מסוג נקודה **Point** (מחלקה המוגדרת בספריה java.awt שמגיעה עם Java) המתארת נקודה במישור עם קואורדינטות (x,y).

5.2 מבנה הרשימה:

- ברשימה לא ריקה, בכל רגע נתון נקודה אחת תהיה מסומנת ע"י הסמן.
- עליכם "לטייל" ברשימה ולשנות את מיקום הסמן.

5.3 שיטות:

השיטות מוגדרות ומתוארות בממשק PointList, כפי שניתן לראות גם בתמונה הבאה.

```
import java.awt.Point;

public interface PointList {

    static final int MAX_SIZE = 100;

    void append (Point newPoint); // Adds newPoint to the end of the list,
                                  // Moves cursor to newPoint
    void clear();                 // Removes all points in the list

    boolean isEmpty();
    boolean isFull();

    boolean goToBeginning();      // If list is not empty moves the cursor to the first element of the list
                                  // and returns true. Otherwise, returns false.
    boolean goToEnd();            // If list is not empty moves the cursor to the last element of the list
                                  // and returns true. Otherwise, returns false.

    boolean goToNext();           // If the cursor is not in the last element of the list,
                                  // moves it to the next element and returns true. Otherwise, returns false.
    boolean goToPrior();          // If the cursor is not in the first element of the list,
                                  // moves it to the previous element and returns true. Otherwise, returns false.

    Point getCursor();            // Returns a copy of the point marked by the cursor, and null if the list is empty

    String toString();            // Outputs the Points in the list. If the list is empty, outputs "Empty list".
                                  // Intended for debugging purposes only.
}
```

5.4 בנאים:

- בנאי ברירת מחדל היוצר מערך בגודל MAX_SIZE
- בנאי עם פרמטר יחיד היוצר מערך שגודלו כערך הפרמטר.

דרישות המעבדה – חלק א':

1. ממשו את הממשק PointList במחלקה בשם ArrayPointList תוך **שימוש במערך** לצורך שמירת רשימת הנקודות
2. בדקו את המחלקה שכתבתם בעזרת מחלקת הבדיקה PointListTest שנכתבה ב JUnit.

חלק ב'

תיאור:

בחלק זה נכיר tokenizer של Java המשמש לניתוח קלטים.

להלן דוגמה של קטע קוד העושה שימוש ב-tokenizer.

קטע הקוד בדוגמה קורא את הנתונים מהקלט התקני ומשתמש בממשק PointList.

המילים מופרדות ע"י הרווחים.

שימו לב, השיטה tokens.nextToken קוראת את רצף הסימנים הבא. רצף זה יכול להיות מילה (TT_WORD) או מספר (TT_NUMBER). אם הרצף הוא מילה, הוא יישמר ב sval ואם הוא מספר הוא יישמר ב nval. התוכנית ממשיכה לקרוא את המספרים עד שתופיעה מילה (למשל abc).

שימו לב, כי קטע קוד זה הוא רק דוגמה לשימוש ב-tokenizer.

ההוראות למה שאתם נדרשים לבצע בחלק זה של התרגיל מופיעות אחריו.

```
import java.io.*;
import java.awt.Point;

public class ArrayPointListTest {

    public static void main(String args[]) throws IOException {
        PointList polygon = new ArrayPointList();       // Set of vertices for a polygon
        Point vertex;
        // Initialize reader and tokenizer for the input stream
        // for reading 'tokens' (namely point values) input from the keyboard.
        InputStreamReader reader = new InputStreamReader(System.in);
        StreamTokenizer tokens = new StreamTokenizer(reader);
        // Use the tokenizer's nextToken( ) method to step through a stream of tokens.
        // Use the tokenizer's instance variable nval to obtain the number read.
        // Since nval is of type double, cast it to an int when reading points x and y (int)tokens.nval
        // Read in the polygon's vertices.
        // Keep reading as long as word (not number!) has not been entered
        System.out.print("Enter the polygon's vertices (end with abc) : ");
        while ((tokens.nextToken()) != StreamTokenizer.TT_WORD){
            vertex = new Point();
            vertex.x = (int)tokens.nval;
            tokens.nextToken();
            vertex.y = (int)tokens.nval;
            polygon.append(vertex);
        }
        // Output the vertices one per line.
        if (polygon.goToBeginning()) {
            do {
                vertex = polygon.getCursor();
                System.out.println("(" + vertex.x + "," + vertex.y + ")");
            } while (polygon.goToNext());
        }
    }

}
```

דרישות המעבדה – חלק ב':

כתבו מחלקה בשם PointListCommandLine התומכת בפקודות הבאות:

פקודה	פעולה
add x y	מוסיף נקודה **לסוף** הרשימה
curr	מראה את הנקודה המוצבעת ע"י הסמן
next	מזיז את הסמן לנקודה הבאה
prev	מזיז את הסמן לנקודה הקודמת
start	מזיז את הסמן לתחילת הרשימה
end	מזיז את הסמן לסוף הרשימה
empty	שאילתא: האם הרשימה ריקה ?
full	שאילתא: האם הרשימה מלאה ?
clear	מחק את כל הרשימה
quit	סיים את התוכנית

כך ש(לדוגמה) עבור הקלט הבא:

```
add 1 2
add 2 3
start
curr
next
curr
quit
```

יתקבל הפלט:

```
true
(2 ,1)
true
(3 ,2)
```

סדר העבודה ופרטים טכניים

- הורידו לתוך Eclipse מתוך GitHub את הפרויקט היחיד ב
  https://github.com/ykanizo/DSLab2022-2023Public
  - השתמשו ב File->Import->Git->CloneURI בתוך התפריט של Eclipse.
  - אם אתם עובדים ב VDI, מומלץ לשנות את המיקום המוצע לפרויקט בתיקייה כלשהי בכונן H.

פורמט קובץ ההגשה ובדיקתו:

פורמט: יש להגיש קובץ ZIP בשם

43_lab01_123456789_987654321.zip

(כמובן, יש להחליף את המספרים עם מספרי ת.ז. של המגישים)

על הקובץ להכיל את כל קבצי ה JAVA שכתבתם. שימו לב: הקובץ לא יכיל את התיקיה שבה הקבצים נמצאים, רק את הקבצים עצמם (אם לא ברור מה ההבדל, ראו סרטון הדגמה מטה).

ניתן ליצור את הקובץ בשרת המכללה ע"י הפקודה הבאה:

```
zip  ~/43_lab01_123456789_987654321.zip *.java
```

שיש לתת כאשר אתם נמצאים בתיקייה src של הפרויקט. הקובץ יווצר בתיקיית הבית שלכם.

בדיקה: בדקו את הקובץ שיצרתם בתוכנת הבדיקה בקישור:
https://cs.telhai.ac.il/homework/

ראו סרטון הדגמה של השימוש בתוכנת הבדיקה.

חשוב !!!
בדיקת ההגשות תבוצע ברובה ע"י תוכנית הבדיקה האוטומטית הנ"ל. תוצאת הבדיקה תהייה בעיקרון זהה לתוצאת הבדיקה הנ"ל שאתם אמורים לערוך בעצמכם . כלומר, אם ביצעתם את הבדיקה באתר החוג, לא תקבלו הפתעות בדיעבד. אחרת, ייתכן שתרגיל שעבדתם עליו קשה ייפסל בגלל פורמט הגשה שגוי וכו'. דבר שהיה ניתן לתקנו בקלות אם הייתם מבצעים את הבדיקה. היות ואין הפתעות בדיעבד, לא תינתן אפשרות של תיקונים, הגשות חוזרות וכד'.

הגשה שלא מגיעה לשלב הקומפילציה תקבל ציון 0.
הגשה שלא מתקמפלת תקבל ציון נמוך מ- 40 לפי סוג הבעיה.
הגשה שמתקמפלת תקבל ציון 40 ומעלה בהתאם לתוצאות הריצה, ותוצאת הבדיקה הידנית של הקוד (חוץ ממקרה של העתקה).
תכנית הבדיקה האוטומטים מכילה תוכנה חכמה המגלה העתקות. מקרים של העתקות יטופלו בחומרה.