

תוכנה 1 – תרגיל 5

הוראות הגשה:

- קראו בעיון את קובץ נוהל הגשת התרגילים אשר נמצא באתר הקורס.
- התרגיל יוגש דרך מערכת ה-VirtualTAU בלבד! (<http://virtual2002.tau.ac.il>).
- יש להגיש קובץ zip יחיד הנושא את שם המשתמש, קובץ ה zip יכיל:
 - א. קובץ פרטים אישיים בשם details.txt המכיל שם ומספר תעודת זהות
 - ב. קבצי ה-Java של התכניות (אין להגיש קבצי .class)
 - ג. קובץ טקסט עם העתק של כל קבצי ה Java. פורמט: txt או doc (לא אופיס 2007)

חלק א: כתיבת מחלקות בהינתן תיאור של השירותים.

עליכם לממש מחלקה בשם DisjointSets המייצגת קבוצה S של disjoint sets $S = \{S_1, S_2, \dots, S_n\}$ כך שכל S_i הינה קבוצה של מספרים שלמים אי-שליליים, והחיתוך בין הקבוצות ריק (http://en.wikipedia.org/wiki/Disjoint_sets). המחלקה תתמוך בשירותים הציבוריים הבאים:

```
/**
 * Create a singleton set containing x (i.e. {x}) and add it
 * to this object.
 * @pre x is not in any of the sets Si
 * @post this = $prev(this) U {{x}}
 */
public void makeSet(int x);

/**
 * Return true if and only if x and y belong to the same
 * set in this object.
 * @pre x in Si, y in Sj
 * @return true iff i == j
 */
public boolean equiv(int x, int y);

/**
 * Find the different sets that x and y belongs to. Remove
 * them from this object and add their union
 * @pre x in Si, y in Sj, i != j
 * @post this = $prev(this) - Si - Sj U {Sij} where Sij := Si U Sj
 */
public void joinSets(int x, int y);

/**
 * Return true if and only if x is in a set of this object
 * @pre true ("no precondition")
 * @return exists i s.t. x in Si
 */
public boolean inASet(int x);
```

אפשרות אחת לממש את המחלקה היא ע"י ייצוג כל קבוצה S_i של S בתור עץ. כל צומת (הקשורה למספר שלם אי שלילי x) מצביעה להורה. בדרך זו, שורש של עץ מייצג בצורה ייחודית את הסט S_i . מערך בשם `parent` יכול להחזיק את כל המצביעים האלה. ספציפית, לכל מספר x , הערך של `parent[x]` הוא מספר צומת האב בעץ או -1 אם x לא שייך לאף סט S_i . שורש כל עץ מצביע לעצמו. אורך המערך צריך להיות גדול יותר מכל מספר x שנמצא כרגע בסט S_i כלשהוא. לפיכך, אם נוסף מספר גדול יותר מהמקסימלי עד כה, נצטרך להחליף את המערך במערך חדש גדול יותר.

כל אובייקט של המחלקה `DisjointSets` מייצג קבוצה של קבוצות זרות, של שלמים אי-שליליים

$\{S_1, \dots, S_n\}$. ייצוג זה מוגדר פורמלית ע"י פונקצית העזר `r(x)`:

```
if parent[x] = -1 then r(x) = -1
else if parent[x] == x then r(x) = x
else r(x) = r(parent[x])
```

כעת, נוכל להגדיר את המיפוי `parent`, שדה של האובייקט, לקבוצה של קבוצות: $S(\text{this}) = \{S_1, S_2, \dots, S_n\}$ כך ש:

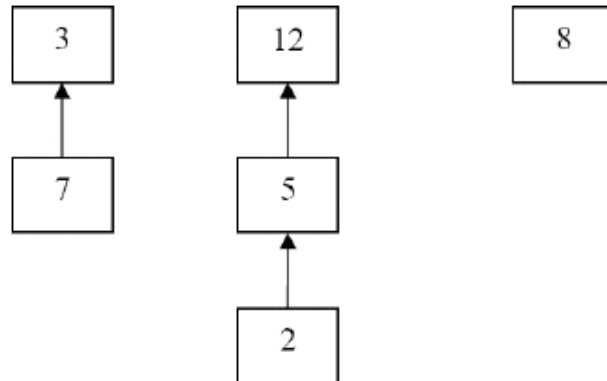
For all x , [for all i , $1 \leq i \leq n$, $x \notin S_i$] iff $[x \geq \text{parent.length or } \text{parent}[x] = -1]$

For all $0 \leq x, y \leq \text{parent.length}$, $x, y \in S_i$ (x and y are in the same set) iff $r(x) = r(y) \neq -1$

למשל, לאובייקט של `DisjointSets` כאשר מערך ה `parent` הוא:

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

העצים יהיו:

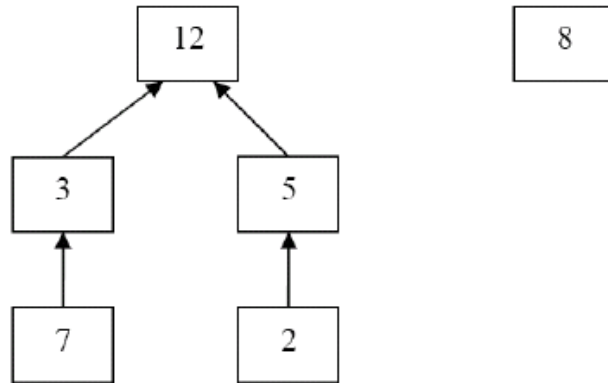


$S = \{ \{3,7\} \{12,5,2\} \{8\} \}$ ו-

אם נקרא ל- $\text{equiv}(3,5)$ false. אם נפעיל $\text{joinSets}(7,5)$ נקבל:

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

העצים יהיו:



ו- $S = \{\{3,7,12,5,2\} \{8\}\}$.

אם נפעיל כעת $\text{makeSet}(4)$ נקבל ש- $S = \{\{3,7,12,5,2\} \{8\} \{4\}\}$

המשימה:

ממשו את המחלקה DisjointSets בצורה יעילה תוך שימוש במערך parent (התבססו על התכנית החלקית מאתר הקורס). שימו לב שניתן להגדיר שירותי עזר (לדוגמא, אולי יעזור להגדיר פונקציה שבודקת שמערך ה-parents ארוך מספיק להכיל את כל האיברים).

חלק ב:

In this exercise, you will get the code of the class [Circle](#). Notice that the instance variables of class Circle include the radius, and two double values x0 and y0 which represent the center coordinates of the circle. We wish to change the implementation of class Circle so that instead of these two variables, we will have an instance of class [Point](#) as a member instance variable representing the center (download [Point.java](#)).

Remember that changing the implementation details should not affect the API of the class! This means that this change should not be noticed by outside users of the class. In particular you are not allowed to modify any of the methods signatures. You will surely need to change the implementation of some of the methods. Use the [Circle.java](#) file given to you and modify it to implement your solution.

To test your modified Circle class, we supply you with a class [CircleViewer](#) which you can use to display your circles. [CircleTest.java](#) is a simple program that uses CircleViewer. This program creates two circles and a CircleViewer object, and adds both circles to the viewer, using the method addCircle() of the viewer.

The CircleViewer "remembers" the circles it should display (by storing a reference to it), and whenever the update() method is called on it, it uses the [Circle](#) interface to retrieve the state of all the circles it should display (i.e., calls getCenter(), getRadius()), and draws them appropriately. To use the CircleViewer class, download [shapes.zip](#) and include it in your project as explained below.

Write your own simple program called SimpleCircleTest that tests your circle implementation class. Remember to test all the methods in the class, and use the CircleViewer class.

- בשאלה נעשה שימוש בקבצי jar. כאשר ברצונכם להוסיף קובץ jar (או zip) לפרויקט ב-Eclipse:
- קליק ימני על הפרויקט
- בחירה ב Build Path ואח"כ Add External Archives
- בחירה בקובץ אותו רוצים להוסיף לפרויקט

הקובץ [hw5partb.zip](#) מכיל את הקבצים הנדרשים בשאלה

חלק ג:

שאלה זו עוסקת בתכנון ובהגדרת טיפוס נתונים עבור פולינום (polynomial – רב איבר) שמקדמיו שלמים.

- **פולינום** הוא ביטוי מהצורה: $c_0 + c_1x + c_2x^2 + c_3x^3 + \dots + c_nx^n$.
- c_0, c_1, \dots, c_n הם מספרים שלמים, והם נקראים **המקדמים** של הפולינום (coefficients).
- המעריך הגדול ביותר בפולינום (של איבר שהמקדם שלו אינו אפס) נקרא **הדרגה** (degree) של הפולינום.

לדוגמא: הביטוי $2x^3 - 5x^2 + 13$ הוא פולינום שדרגתו 3 ומקדמיו הם: 2, -5, 0, 13.

בהמשך התרגיל אתם מתבקשים לממש את המחלקה Polynomial.

להלן תזכורת מתמטית:

סכום של שני פולינומים הוא פולינום שהמקדמים שלו הינם סכומים של המקדמים המתאימים (לפי חזקות) של שני הפולינומים הנתונים. למשל, עבור $a(x) = 6x^3 + 3x^2 + 8$ ו- $b(x) = 2x^2 + 4x + 1$ נקבל: $a(x) + b(x) = 6x^3 + 5x^2 + 4x + 9$.

מכפלה של שני פולינומים הינה פולינום, אשר מתקבל כתוצאה של "פתיחת סוגריים" רגילה בביטוי:
 $a(x) \cdot b(x) = (a_nx^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0) \cdot (b_mx^m + b_{m-1}x^{m-1} + \dots + b_1x + b_0)$

לדוגמה, עבור $a(x) = 3x^2 + x + 2$ ו- $b(x) = x^3 + 4x$ נקבל:
 $a(x) \cdot b(x) = (3x^2 + x + 2) \cdot (x^3 + 4x) = 3x^{2+3} + 3 \cdot 4x^{2+1} + x^{1+3} + 4x^{1+1} + 2x^3 + 8x =$
 $= 3x^5 + 12x^3 + x^4 + 4x^2 + 2x^3 + 8x = 3x^5 + x^4 + 14x^3 + 4x^2 + 8x.$

ובמקרה הכללי:

$$\begin{aligned} p(x) &= p_0 + p_1x + p_2x^2 + \dots + p_mx^m \\ q(x) &= q_0 + q_1x + q_2x^2 + \dots + q_nx^n \\ p(x)q(x) &= p_0q_0 + (p_0q_1 + p_1q_0)x + (p_0q_2 + p_1q_1 + p_2q_0)x^2 \\ &\quad + (p_0q_3 + p_1q_2 + p_2q_1 + p_3q_0)x^3 \\ &\quad + (p_0q_4 + p_1q_3 + p_2q_2 + p_3q_1 + p_4q_0)x^4 + \dots \\ &\quad + (p_0q_{n+m} + p_1q_{n+m-1} + p_2q_{n+m-2} \dots \\ &\quad \quad + p_{n+m-1}q_1 + p_{n+m}q_0)x^{n+m}. \end{aligned}$$

1. ממשו את המחלקה Polynomial. אם לצורך הגדרת המחלקה יש צורך בשרותי עזר נוספים שדות, בנאים או טיפוסים חדשים ממשו גם אותם.
2. עבור כל שרות הוסיפו תיעוד של השרות במונחי **המצב המופשט** של הטיפוס כפי שמתואר בראש המנשק. את תשובותיכם מלאו במקום המיועד מעל לחתימת כל אחת מהשרותים.
3. מהי פונקציית ההפשטה של המימוש אותו בחרתם? השלימו את $AF(\text{this}) = \dots$ כהערה בראש המחלקה.

4. מהו משתמר הייצוג של המימוש אותו בחרתם? השלימו את `@representation_inv:...` כהערה בראש המחלקה.

5. הגדירו וממשו מחלקה `TestPolynomial` (בעלת פונקציית `main`):
- a. הגדירו מערך של פולינומים באורך ארבע ומלאו אותו בפולינומים (hard coded)
 - b. בקשו מהמשתמש (השתמשו במחלקה `Scanner`):
 - i. לבחור פולינום (לפי המיקום במערך)
 - ii. לבחור את אחת מהפעולות המוגדרות על פולינומים (`degree, coeff, add,`)
(`mul, sub, minus`)
 - iii. לפי הפעולה, אם יש צורך, יש לבחור את הפולינום עליו תתבצע הפעולה (לפי המיקום במערך) או את הדרגה (עבור `coeff`)
 - iv. לחשב את בקשת המשתמש ולהדפיס את התוצאה למסך.

```

/** פולינום מקובע (immutable) שמקדמיו שלמים
 * @abst:  $c_0 + c_1x + c_2x^2 + c_3x^3 + \dots + c_nx^n$ 
 *
 * AF(this) =
 *
 * @representation_invariant:
 *
 */
public class Polynomial {

    /** השרות מחזיר את דרגת הפולינום
     *
     * @abst:
     */
    public int degree ( ){
        // implementation missing...
    }

    /** d מעריך של האיבר של המקדם את המקדם של האיבר עם מעריך d
     *
     * @abst:
     */
    public int coeff (int d){
        // implementation missing...
    }

    /** q השרות מחזיר את הסכום של הפולינום הנוכחי ושל הארגומנט q
     *
     * @abst:
     */
    public Polynomial add (Polynomial q){
        // implementation missing...
    }

    /** q השרות מחזיר את המכפלה של הפולינום הנוכחי ושל הארגומנט q
     *
     * @abst:
     */
    public Polynomial mul (Polynomial q){
        // implementation missing...
    }

    /** מחזיר את תוצאת חיסור q מהפולינום הנוכחי
     *
     * @abst:
     */
    public Polynomial sub (Polynomial q){
        // implementation missing...
    }

    /** מחזיר פולינום שמקדמיו נגדיים (מינוס) למקדמי הפולינום הנוכחי
     *
     * @abst:
     */
    public Polynomial minus ( ){
        // implementation missing...
    }

    /** מחזיר מחרוזת המתארת את הפולינום הנוכחי
     *
     * @abst:
     */
    public String toString ( ){
        // implementation missing...
    }
}

```