

# מטלת מנחה (ממ"ן) 15

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

נושא המטלה: רשימות מקושרות

חומר הלימוד למטלה: יחידה 11

משקל המטלה: 5 נקודות

מספר השאלות: 2

מועד אחרון להגשה: 8.2.2020

סמסטר: 2020א

(ת)

נגדיר: פולינום הוא ביטוי מהצורה  $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$

כאשר המקדמים  $a_i$  הם מספרים ממשיים.

דוגמאות לפולינומים:

$$r = 2.8x^{10} - 4.9x^3 + 6.5x^2 - 12.0$$

$$p = 8.0x^3 - 3.0x^2 - x + 7.0$$

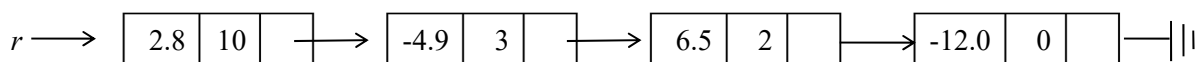
$$q = x^2 + 6.0x - 15.0$$

בממ"ן זה נייצג פולינום באמצעות רשימה מקושרת.

$$r = 2.8x^{10} - 4.9x^3 + 6.5x^2 - 12.0$$

למשל, הפולינום

יוצג כך:



שימו לב שהרשימה ממוינת בסדר יורד לפי חזקות!

כדי לעשות זאת, עליכם להגדיר שתי מחלקות.

## שאלה 1 - (20%)

המחלקה `PolyNode` מייצגת איבר אחד של פולינום.  
לכל אובייקט במחלקה שלושה שדות:

```
1. int _power           // חזקה
2. double _coefficient  // המקדם
3. PolyNode _next       // מצביע לאיבר הבא
```

למחלקה זו עליכם להגדיר שלושה בנאים:

```
1. public PolyNode (int power, double coefficient)
```

בנאי המקבל חזקה ומקדם, שדה ה-`_next` יאותחל ל-`null`.  
שימו לב שחזקה לא יכולה להיות שלילית בפולינום. במידה והשיטה מקבלת חזקה שלילית אזי תכונת החזקה תקבל ערך 0 והמקדם יקבל ערך 0.

```
2. public PolyNode(int power, double coefficient, PolyNode next)
```

בנאי המקבל את שלושת התכונות של האובייקט ומאתחל אותן לפי ערכי הפרמטרים.

- שימו לב שחזקה לא יכולה להיות שלילית בפולינום. במידה והשיטה מקבלת חזקה שלילית אזי תכונת החזקה תקבל ערך 0 והמקדם יקבל ערך 0. שימו לב שפה `aliasing` הוא לא טעות. יש לעדכן את המידע (`next`) עצמו ולא עותק של המצביע.

```
3. public PolyNode( PolyNode p )
```

- בנאי העתקה. שימו לב שפה `aliasing` הוא לא טעות. יש להעתיק את המידע (`next`) עצמו ולא עותק של המצביע.

השיטות במחלקה `PolyNode` הן:

- `public int getPower()` - שיטה המחזירה את החזקה.
- `public double getCoefficient()` - שיטה המחזירה את המקדם.
- `public PolyNode getNext()` - שיטה המחזירה מצביע לאיבר הבא. שימו לב שפה `aliasing` הוא לא טעות. יש להחזיר את המצביע `next` ולא עותק של המצביע.
- `public void setPower(int power)` - שיטה המקבלת חזקה ומעדכנת את תכונת החזקה. שימו לב שחזקה לא יכולה להיות שלילית בפולינום. אם מתקבלת חזקה שלילית אין לבצע שינוי בחזקה.
- `public void setCoefficient (double coefficient)` - שיטה המקבלת מקדם ומעדכנת את תכונת המקדם.

- `public void setNext( PolyNode next)` - שיטה המקבלת מצביע ומעדכנת את תכונת המצביע לאיבר הבא. שימו לב שפה `aliasing` הוא לא טעות. יש לעדכן את המידע `(next)` עצמו ולא עותק.

- `public String toString()` - שיטה המחזירה מחרוזת המייצגת את האיבר.

### דגשים לשיטה `toString` -

- ✓ אם המקדם שווה 0 יש להחזיר מחרוזת ריקה.
- ✓ אם החזקה שווה 0 יש להחזיר את המקדם בלבד.
- ✓ אם המקדם שווה 1 או מינוס 1 אין להחזיר את המקדם אבל כמובן, אם המקדם שווה מינוס 1 יש להחזיר סימן מינוס. במידה שהמקדם שווה 1 או מינוס 1 וגם החזקה שווה 0 יש להחזיר את המספר 1 או מינוס 1 בלבד בהתאמה.
- ✓ אם החזקה שווה 1 יש להחזיר את המקדם (במידה שהמקדם אינו 1 או מינוס 1) ו- $x$  ואין להוסיף חזקה.
- ✓ יש להחזיר מקדם (במידה שהמקדם אינו 1 או מינוס 1), אחר כך  $x$  (במידה שהחזקה אינה 0) ואחר כך את החזקה (במידה והחזקה אינה 1).

### דוגמאות לפלט:

אם החזקה היא 2 והמקדם 6.5 אזי השיטה תחזיר את המחרוזת `"6.5x^2"`

אם החזקה היא 2 והמקדם -6.5 אזי השיטה תחזיר את המחרוזת `"-6.5x^2"`

אם החזקה היא 0 והמקדם 6.5 אזי השיטה תחזיר את המחרוזת `"6.5"`

אם החזקה היא 2 והמקדם 1 אזי השיטה תחזיר את המחרוזת `"x^2"`

אם החזקה היא 2 והמקדם -1 אזי השיטה תחזיר את המחרוזת `"-x^2"`

אם החזקה היא 0 והמקדם 1 אזי השיטה תחזיר את המחרוזת `"1"`

אם החזקה היא 0 והמקדם -1 אזי השיטה תחזיר את המחרוזת `"-1"`

אם החזקה היא 1 והמקדם 3 אזי השיטה תחזיר `"3.0x"`

אם החזקה היא 1 והמקדם -1 אזי השיטה תחזיר `"-x"`

### עליכם לדייק בתשובתכם.

### שימו לב:

- השיטות שתכתבו צריכות להיות יעילות ככל הניתן. תשובה שאינה יעילה מספיק כלומר, שתהיה בסיבוכיות גדולה יותר מזו הנדרשת לפתרון הבעיה (במקום או בזמן) תקבל מעט נקודות בלבד.
- אל תשכחו לתעד את השיטות שכתבתם בתיעוד פנימי וב-API.
- כתבו מה סיבוכיות הזמן וסיבוכיות המקום של כל שיטה בתוך ה-API של השיטה.

## שאלה 2 - (80%)

המחלקה **Polynom** מייצגת פולינום באמצעות רשימה מקושרת.

במחלקה זו מותר להגדיר **אך ורק תכונה פרטית אחת** שתצביע להתחלת הרשימה. עליכם לכתוב בנאים ושיטות לפי המפורט להלן:

<code>public Polynom ()</code>	בנאי ריק - יוצר רשימה ריקה
<code>public Polynom (PolyNode p)</code>	בנאי המקבל כפרמטר אובייקט מהמחלקה <code>PolyNode</code> ושם אותו כאיבר הראשון ברשימה
<code>public PolyNode getHead()</code>	שיטה המחזירה את הצומת בראש הפולינום
<code>public void setHead( PolyNode p )</code>	שיטה המקבלת איבר מטיפוס <code>PolyNode</code> ומעדכנת את האיבר בראש הרשימה
<code>public Polynom addNode (PolyNode p)</code>	שיטה המקבלת כפרמטר אובייקט מהמחלקה <code>PolyNode</code> ומוסיפה אותו לפולינום (במקום המתאים יש לשמור על הפולינום ממויין). השיטה מחזירה את הפולינום החדש.
<code>public Polynom multByScalar (int num)</code>	שיטה המקבלת כפרמטר מספר שלם (סקלר), ומכפילה את הפולינום בסקלר זה ומחזירה את הפולינום החדש.
<code>public Polynom addPol (Polynom other)</code>	שיטה המקבלת כפרמטר פולינום <code>other</code> ומוסיפה אותו לפולינום שעליו מופעלת השיטה. השיטה מחזירה את פולינום הסכום.
<code>public Polynom multPol (Polynom other)</code>	שיטה המקבלת כפרמטר פולינום <code>other</code> ומכפילה אותו בפולינום שעליו מופעלת השיטה. השיטה מחזירה את פולינום המכפלה.
<code>public Polynom differential ()</code>	שיטה המחשבת את הנגזרת של הפולינום עליו היא מופעלת ומחזירה את פולינום הנגזרת.
<code>public String toString()</code>	שיטה המחזירה ייצוג מחרוזתי של הפולינום.

**דגשים לשיטה `toString`:**

שימו לב שאיברים (מונומים) שהמקדם שלהם הוא 0 לא יודפסו. לדוגמא, הפולינום שלהלן:

$$r = 2.8x^{10} - 4.9x^3 + 6.5x^2 - 12.0$$

ייוצג במחרוזת כך :

$$2.8x^{10}-4.9x^3+6.5x^2-12.0$$

קודם יופיע מקדם (במידה שהמקדם אינו 1 או מינוס 1 (אלא אם החזקה שווה 0 ואז יופיע המספר 1 או מינוס 1 בלבד)), ואחר כך  $x$  (במידה שהחזקה אינה 0) ואחר כך החזקה (במידה שהחזקה אינה 1). בין כל שני איברים מופיע סימן (+ או -) ושוב המקדם של האיבר הבא. באיבר הראשון אין רווח לפני המקדם ואין רווח אחרי האיבר האחרון. אם האיבר הראשון הוא חיובי לא יופיע סימן + אבל אם האיבר הראשון הוא שלילי יופיע סימן -. אם הרשימה ריקה או שכל המקדמים ברשימה שווים לאפס אזי תוחזר מחרוזת ריקה.

**עליכם לדייק בתשובתכם.**

**תזכורת מתמטית:**

- **מכפלה בסקלר** של פולינום  $p$  בסקלר  $s$  היא הפולינום המתקבל על-ידי הכפלת כל איבר של הפולינום  $p$  בסקלר  $s$ .

**דוגמא:**

$$\begin{aligned}(8.0x^3 - 3.0x^2 - x + 7.0) \times (-2.0) &= \\ &= -16.0x^3 + 6.0x^2 + 2.0x - 14.0\end{aligned}$$

- **סכום** של 2 פולינומים  $p, q$  הוא הפולינום המתקבל על-ידי כינוס מקדמים של איברים בעלי חזקה זהה.

**דוגמא:**

$$\begin{aligned}p + q &= (8.0x^3 - 3.0x^2 - x + 7.0) + (x^2 + 6.0x - 15.0) = \\ &= 8.0x^3 - 2.0x^2 + 5.0x - 8.0\end{aligned}$$

- **מכפלה** של 2 פולינומים  $p, q$  היא הפולינום המתקבל על-ידי מכפלת כל איבר של הפולינום  $p$  בכל איבר של הפולינום  $q$ , וכינוס מקדמים של איברים בעלי חזקה זהה.

**דוגמא:**

$$\begin{aligned}p * q &= (8.0x^3 - 3.0x^2 - x + 7.0) * (x^2 + 6.0x - 15.0) \\ &= 8.0x^3(x^2 + 6.0x - 15.0) - 3.0x^2(x^2 + 6.0x - 15.0) - \\ &\quad - x(x^2 + 6.0x - 15.0) + 7.0(x^2 + 6.0x - 15.0) \\ &= 8.0x^5 + 48.0x^4 - 120.0x^3 - 3.0x^4 - 18.0x^3 + 45.0x^2 - \\ &\quad - x^3 - 6.0x^2 + 15.0x + 7.0x^2 + 42.0x - 105.0 \\ &= 8.0x^5 + 45.0x^4 - 139.0x^3 + 46.0x^2 + 57.0x - 105.0\end{aligned}$$

- **נגזרת של פולינום**  $p$  היא הפולינום המתקבל על-ידי חישוב של כל אחד המונומים כך : אם

$$\text{המונום הוא } a \cdot x^b \text{ אזי מונום הנגזרת הוא } (a \times b) \cdot x^{b-1}$$

**דוגמא:**

$$p = 8.0x^3 - 3.0x^2 - x + 7.0$$

$$p' = 24.0x^2 - 6.0x - 1$$

### כמה הבהרות לממ"ן:

1. aliasing - שימו לב שעד כה התייחסנו אל aliasing כטעות, בדרך כלל מפני שרצינו שליטה מלאה על הנתונים במטרה להגן על המחלקה שלנו מפני שינויים "מבחוץ". כאשר מדובר במבני נתונים דינמיים ורשימות בפרט העקרון אינו תקף יותר, להפך, הרשימה היא מקום לאחסון נתונים ולכן אנחנו נשמור את המידע המקורי ולא העתק שלו.

2. כל הפעולות על הפולינום, הוספת איבר (addNode) כפל בסקלר (multByScalar), חיבור (addPol), כפל פולינומים (multPol) ונגזרת (differential) צריכות להתבצע על הפולינום עצמו. לדוגמא, לאחר קריאה לשיטה multByScalar הפולינום שעליו הופעלה השיטה ישתנה בהתאם. הסיבה שהשיטה מחזירה גם את הפולינום החדש ולא העתק שלו ולא void למשל היא שנוכל לשרשר פעולות. לדוגמא, נניח ש- p1, p2 הם פולינומים. נוכל לבצע את הפקודה:

```
p1.addPol (p2.multByScalar (5.0)) ;
```

לאחר ביצוע פקודה זו, הפולינום p2 ישתנה ויוכפל ב-5, והפולינום p1 ישתנה ויכיל את החיבור של עצמו עם  $p2 * 5.0$

רמז: כדי להחזיר את הפולינום עצמו ולא העתק שלו מחזירים this.

3. בכל הפעולות על הפולינום, הוספת איבר (addNode) כפל בסקלר (multByScalar), חיבור (addPol), כפל פולינומים (multPol) ונגזרת (differential) אין להשאיר רשימה המכילה יותר מחוליה אחת בעלת אותה חזקה.

לדוגמא: אם הפולינום היה  $3x^4 + 5x^3 + 8$

ומפעילים עליו addNode(p) כאשר p מכיל  $2x^3$

אזי המחרוזת שתתקבל מ-toString עבור הרשימה המקושרת שמייצגת את הפולינום תהיה

$$3x^4 + 7x^3 + 8$$

בנוסף, במידה שיווצר צומת עם מקדם של 0, אתם יכולים להחליט אם ברצונכם למחוק את האיבר מהרשימה או להשאיר אותו ופשוט לא להוסיף את האיבר למחרוזת המוחזרת מ-toString.

לדוגמא: אם הפולינום היה  $3x^4 - 2x^3 + 7$

ומפעילים עליו addNode(p) כאשר p מכיל  $2x^3$

אזי המחרוזת שתתקבל מ-toString עבור הרשימה המקושרת שמייצגת את הפולינום תהיה

$$3x^4 + 7$$

4. בכל הפעולות על הפולינום המקבלות עוד Polynom כפרמטר, חיבור (addPol) וכפל פולינומים (multPol) השיטה תשנה רק את הפולינום שעליו הופעלה השיטה, ולא תשנה את הפולינום שמתקבל כפרמטר. במידה שהועבר פרמטר שהוא null אין לשנות את הפולינום שעליו הופעלה השיטה.

5. יש לשמור על פולינום ממוין בכל השיטות כמו-כן ניתן להניח כי כל הפולינומים שמתקבלים כפרמטרים לשיטות הם ממוינים.

שימו לב,

- השיטות שתכתבו צריכות להיות יעילות ככל הניתן. תשובה שאינה יעילה מספיק כלומר, שתהיה בסיבוכיות גדולה יותר מזו הנדרשת לפתרון הבעיה (במקום או בזמן) תקבל מעט נקודות בלבד. אין צורך להקפיד על כתיבת השיטה multPol בסיבוכיות הנמוכה ביותר.
- אל תשכחו לתעד את השיטות שכתבתם בתיעוד פנימי וב-API.
- לא שמנו API מפורט באתר הקורס. עליכם לכתוב אותו בעצמכם.
- כתבו מה סיבוכיות הזמן וסיבוכיות המקום של כל שיטה בתוך ה API של השיטה. אין צורך לציין את סיבוכיות השיטה multPol.
- באתר הקורס בספריית הקבצים להורדה תוכלו למצוא את הקבצים PolynomTester.java ו-PolyNodeTester.java ובהם מחלקות שמבצעות בדיקות למחלקות PolyNode ו-Polynom. חובה להיעזר בקבצים אלו כדי לבדוק את המחלקות שכתבתם. הטסטר והמחלקות צריכות לרוץ ללא שגיאות קומפילציה. שמנו באתר גם את הקבצים PolynomTester output ו-PolyNodeTester output ובהם הפלטים הדרושים. יש להשוות אותם עם הפלטים שלכם. הם צריכים להיות זהים.

## הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו ששמות השיטות והמחלקות יהיו בדיוק לפי הוראות הממ"ן, וכן שההדפסות יהיו בדיוק כפי שמופיע במטלה.
3. הקפידו לתעד בתיעוד פנימי וב-API את כל השיטות שיש במחלקות PolyNode, Polynom.
4. עליכם להגיש את הקבצים Polynom.java, PolyNode.java עטפו אותם בקובץ zip ושלחו. אין לשלוח קבצים נוספים.

## בהצלחה

