

תוכנה 1

תרגיל מספר 6

כתיבת מחלקות לפי מפרט

הנחיות כלליות:

קראו בעיון את קובץ נהלי הגשת התרגילים אשר נמצא באתר הקורס.

- הגשת התרגיל תיעשה במערכת ה-moodle בלבד (<http://moodle.tau.ac.il/>).
- יש להגיש קובץ zip יחיד הנושא את שם המשתמש ומספר התרגיל (לדוגמא, עבור המשתמש aviv יקרא הקובץ aviv_hw6.zip). קובץ ה-zip יכול:
 - א. קובץ פרטים אישיים בשם details.txt המכיל את שמכם ומספר ת.ז.
 - ב. קבצי ה-java של התוכניות אותם התבקשתם לממש.

שימו לב: עבור תרגיל, חלקים א'-ב' נתונים לכם שלדי מחלקות באתר הקורס. כאשר אתם משלימים שלד אתם רשאים להוסיף מתודות ומחלקות עזר כרצונכם. אולם, אסור לכם לשנות את כותרות המתודות הנתונות או את החוזה המוגדר עבורן.

המקומות שבהם נדרשת השלמת קוד מסומנים ב- **TODO** // . תוכלו להיעזר ברשימת ה-tasks ב-Eclipse על מנת לראות את כל המקומות בקוד בהם מופיע הסימון הנ"ל. לשם כך, לחצו על `Window > Show view > Tasks` ב-Eclipse. לחיצה כפולה על אחת השורות ב-view שנפתח תוביל אתכם לנקודה הרלוונטית בקוד.

בתרגיל זה נכתוב, בשני שלבים, מחלקות לייצוג פולינומים מרובי-משתנים, למשל

$$15xy - 52x^2z + 19uvx^3y^2z - 2$$

- **משתנה** (variable) מיוצג ע"י אות קטנה בין a-z.
- כל מחובר בפולינום נקרא **מונום** (monomial). כל מונום מורכב ממכפלה של משתנים בחזקות שונות (degrees), וממקדם שהוא מס' שלם. למשל, $-52x^2z$ הוא מונום בו המקדם הוא -52, ומוכפלים בו המשתנה x בחזקת 2 והמשתנה z בחזקת 1. לשם הפשטות, נניח שכל משתנה מופיע במכפלה פעם אחת לכל היותר, ושהם מופיעים במכפלה בסדר אלפביתי.
- **חיבור:** אם"ם לשני מונומים יש אותן חזקות עבור כל משתנה, ניתן לחבר אותם למונום יחיד עם אותן חזקות, ע"י חיבור המקדמים שלהם. למשל, $-52x^2z + 84x^2z = 32x^2z$.
- נניח כי $x^0 = 1$ לכל משתנה x.
- פולינום (polynomial) הוא סכום של מונומים.

חלק א' (40%) – מונום

באתר הקורס נתון לכם שלד המחלקה `sw1.polynomials.Monomial`

תחילה, החליטו כיצד לייצג את המידע המקודד במונום (המקדם של המונום והחזקה של כל משתנה) באופן נוח לביצוע חישובים, והגדירו שדות מופע בהתאם.

לאחר מכן, השלימו את מתודות המחלקה לפי הגדרת המונום שלעיל ולפי ההנחיות \ החוזה של כל מתודה, תוך שימוש בשדות שהגדרתם. מותר להוסיף מתודות עזר ושדות כרצונכם, אך אין לשנות את חתימת המתודה או את החוזה המוגדר עבורה. להלן הסבר קצר נוסף בעברית על מתודות המחלקה.

```
public Monomial(int coefficient)
    בנאי המקבל כקלט את מקדם המונום ויוצר מונום שבו החזקות של כל המשתנים הן 0.
```

```
public int getCoefficient()
    שאילתה צופה המחזירה את מקדם המונום
```

```
public void setCoefficient(int coefficient)
    פקודה המשנה את ערך המקדם של המונום
```

```
public boolean isVariable(char variable)
    שאילתה צופה המחזירה האם המשתנה הנתון מופיע במונום.
```

```
public int getDegree(char variable)
    שאילתה צופה המקבלת כקלט שם משתנה ומחזירה את חזקתו במונום.
```

```
public void setDegree(char variable, int degree)
    פקודה המקבלת כקלט משתנה ואת חזקתו החדשה ומשנה את המונום בהתאם.
```

```
public boolean hasSameDegrees(Monomial other)
    שאילתה צופה המקבלת כקלט מונום אחר ומשווה אותו למונום הנוכחי (this). מחזירה true אם"ם עבור כל משתנה בין a-z, דרגתו זהה ב-this וב-other.
```

```
public Monomial getCopy()
    שאילתה מפיקה המחזירה "עותק בטוח לשימוש" של המונום, כך שבמונום המוחזר אותו מקדם ואותן חזקות כמו ב-this, אך שינוי המונום החדש לא ישפיע על this ולהפך. יצירת עותקים תהיה שימושית עבורנו בעבודה עם פולינומים בהמשך.
```

```
public int evaluate(int[] assignment)
    שאילתה צופה אשר מחשבת את ערך המונום עבור השמה נתונה. ההשמה מתקבלת כמערך שלמים בגודל 26, כך שבאינדקס 0 ההשמה למשתנה a, באינדקס 1 ההשמה למשתנה b וכך הלאה. למשל, עבור ההשמה אשר ממפה את המשתנה x לערך 8, וממפה כל משתנה אחר למספר כלשהו, המונום  $2x$  יחזיר את הערך 16. שימו לב, אנו מניחים כי  $0^0 = 1$ .
```

```
public String toString()
    שאילתה המחזירה ייצוג של המונום כמחרוזת, לפי הקונבנציה המתמטית, ותוך שימוש בסימן ^ לייצוג חזקה. לדוגמא,
    15xy      → 15xy
    -52x2z   → -52x^2z
    19uvx3y2z → 19uvx^3y^2z
    -2       → -2
    במידה והמקדם הינו 1 או -1, אין צורך להדפיס את המקדם של המונום.
```

חלק ב' (40%) – פולינום

באתר הקורס נתון לכם שלד המחלקה `sw1.polynomials.Polynomial`

גם כאן, עליכם לבחור את הייצוג הפנימי (השדות) של הפולינום, ולהשלים את המתודות המוגדרות בשלד המחלקה. **שימו לב** - עליכם להקפיד בחלק זה על "בטיחות המידע" נגד שינויים חיצוניים: כאשר אנחנו מעבירים מונום m כקלט לפולינום p , או מחזירים אותו כפלט, לא נרצה ששינוי m ישפיע על p , ולהפך. בנוסף, כאשר אנחנו מפעילים על פולינום p שאילתה מפיקה המחזירה פולינום q , לא נרצה ששאילתה זו תשנה את ערכו של p , או ששינוי p ישפיע על q , ולהפך. מומלץ להיעזר לשם כך במתודה `Monomial.copy`.

```
public Polynomial(Monomial[] monomials)
```

בנאי המקבל כקלט מערך של מונומים ומייצר פולינום שהוא סכום המונומים הנ"ל לפי הסדר שבו הם מופיעים במערך. כלומר, בהינתן הקלט $\{x, y\}$ נבנה את הפולינום הנוכחי להיות $x+y$.

```
public int getMonomialCount()
```

שאילתה צופה המחזירה את מס' המונומים בפולינום.

```
public Monomial getMonomial(int index)
```

שאילתה צופה המחזירה את המונום במקום `index` (החל מ-0) בפולינום הנוכחי, בהנחה שזהו אינדקס תקין עבור מונום בפולינום הנוכחי. שימו לב לבטיחות המידע ביחס למונום המוחזר!

```
public Polynomial add(Polynomial other)
```

שאילתה מפיקה המקבלת כקלט פולינום נוסף, ומחזירה את הסכום של הפולינום הנוכחי `this` ושל `other`. יש לפשט את הביטוי, למשל הוספת $3x$ לפולינום x תיצור את הפולינום $4x$. שימו לב לבטיחות המידע ביחס לפולינום המוחזר!

```
public Polynomial multiply(Polynomial other)
```

שאילתה מפיקה המקבלת כקלט פולינום נוסף, ומחזירה את המכפלה של הפולינום הנוכחי `this` ושל `other`. יש לפשט את הביטוי, למשל מכפלת $3x$ בפולינום $2x$ תיצור את הפולינום $6x^2$. שימו לב לבטיחות המידע ביחס לפולינום המוחזר!

```
public int evaluate(int[] assignment)
```

שאילתה צופה אשר מחשבת את ערך הפולינום עבור השמה נתונה. ההשמה מתקבלת כמערך שלמים בגודל 26, כך שבאינדקס 0 ההשמה למשתנה a , באינדקס 1 ההשמה למשתנה b וכך הלאה. למשל, עבור ההשמה אשר ממפה את המשתנה x לערך 8, וממפה כל משתנה אחר למספר כלשהו, הפולינום $2x$ יחזיר את הערך 16. שימו לב, אנו מניחים כי $0^0 = 1$.

```
public String toString()
```

פקודה אשר מחזירה ייצוג של הפולינום הנוכחי כמחרוזת, שבה מחוברים הייצוגים של המונומים כמחרוזות. לדוגמא, $15xy - 52x^2z + 19uvx^3y^2z - 2 \rightarrow 15xy-52x^2z+19uvx^3y^2z-2$. שימו לב למקרי קצה! למשל, פולינום עם 0 מונומים שקול לפולינום עם מונום יחיד בעל מקדם 0. במידה והמקדם הינו 1 או -1, אין צורך להדפיס את המקדם של המונום.

בדקו את עצמכם: באתר הקורס נתונה לכם לצד התרגיל גם

המחלקה `sw1.polynomials.TestPolynomials` אשר מכילה שימוש בסיסי בפולינומים ומונומים. אין צורך להגיש את המחלקה, ומומלץ לשנות אותו ולהוסיף לה בדיקות משלכם (למשל, של הדוגמאות הנתונות למעלה).

חלק ג' (20%) – חידות ג'אוה תיקון מחלקות קיימות

בכל סעיף של חלק זה תקבלו חבילה ובה מספר מחלקות. חבילות אלה מצורפות לתרגיל באתר הקורס. עליכם לשנות את הקוד בהתאם להנחיות, כדי לקבל את התוצאה הנדרשת. יש להגיש את כל המחלקות (כולל אלה שלא שיניתם בהם דבר, וכמובן, הקוד המתוקן).

1. החבילה `il.ac.tau.cs.sw1.riddle.a` מכילה שתי מחלקות, A ו-B.

B היא תכנית המקבלת כארגומנט מס' שלם. עליכם לשנות את הקוד בתוך `printA()` במחלקה-A בלבד כך שבהרצת פונקציית ה-`main` ב-B יודפס המספר שניתן כקלט בין A1 ל-A2. לדוגמא, אם הקלט הוא 15, יודפס:

```
B
A1
15
A2
```

- מותר: לשנות את הקוד בתוך `printA.A`.
 - אסור: לשנות את B, את חתימת `printA()`, וקוד ב-A שנמצא מחוץ ל-`printA()`.
2. החבילה `il.ac.tau.cs.sw1.riddle.b` מכילה שלוש מחלקות, A, B ו-C.

C היא תכנית המקבלת כארגומנטים שלוש מחרוזות. עליכם להשלים את מימוש המתודות `printA`, `printA2` ו-`printA3` ב-A כך שבהרצת פונקציית ה-`main` ב-C יודפסו 3 המחרוזות בין הכוכביות. לדוגמא, אם הקלט לתכנית הוא `hello world bye`, יודפס:

```
hello
***
world
***
bye
```

- מותר: לשנות את הקוד בתוך `printA`, `printA2` ו-`printA3`.
- אסור: לשנות את B ו-C, את חתימות המתודות `printA`, `printA2` ו-`printA3`, וקוד ב-A שנמצא מחוץ למתודות הנ"ל.

בהצלחה!