

תרגיל 10 בקורס תוכנה 1 סמסטר א' תשס"ח

הוראות הגשה:

1. קראו בעיון את קובץ נוהלי הגשת התרגילים אשר נמצא באתר הקורס.
2. הגשת התרגיל תעשה ע"י המערכת VirtualTAU (<http://virtual.tau.ac.il>). הוראות שימוש במערכת ניתן למצוא ב-<http://virtual2002.tau.ac.il/upload/misc/main1.html>
3. הצורה הרצויה להגשת התרגיל היא יצירת קובץ zip אשר יכיל: (1) קבצי ה-java של התכניות שהתבקשת לכתוב, (2) קובץ WORD אשר בו העתק של כל קובץ java.

חלק 1: הורשה

ברצוננו לתאר בתוכנה מגוון של ביטויים חשבוניים. ביטוי חשבוני הוא ישות הניתנת לשערוך כגון מספר או פעולה חשבונית המופעלת על שניים או שלושה ביטויים חשבוניים (כן – ההגדרה רקורסיבית). כל ביטוי חשבוני יודע להדפיס את עצמו.

עליך להגדיר את הישויות הבאות (מחלקות קונקרטיות, מחלקות מופשטות ומנשקים):

1. Expression – ישות המייצגת **ביטוי** כלשהו.
2. Literal – ישות המתארת מספר בודד (double).
3. BinaryOp – ישות המתארת **פעולה** בינארית (פעולה על שני ביטויים).
4. TernaryOp – ישות המתארת **פעולה** טרינארית (פעולה על שלושה ביטויים).
5. Sum – ישות המתארת סכום.
6. Product – ישות המתארת מכפלה.
7. Exponent – ישות המתארת חזקה.
8. CondExp – ישות המתארת פעולה על שלושה פרמטרים a, b, c שהיא $a ? b : c$. שערוך הביטוי מתבצע כך: אם ערכו של a שונה מ-0.0 יוחזר ערכו של b אחרת יוחזר ערכו של c .

בדקו את עצמכם ע"י קוד הלקוח הבא:

```
public class Client {  
    public static void main(String[] args) {  
        Expression l1 = new Literal(1.0);  
        Expression l2 = new Literal(2.0);  
        Expression l3 = new Literal(3.0);  
        Expression sum = new Sum(l1, l2);  
        Expression e1 = new Exponent(l3, sum);  
  
        Expression prod = new Product(l1, l2);  
        Expression exp = new Exponent(l2, l3);  
        Expression e2= new CondExp(sum, prod, exp);  
  
        System.out.println(e1 + " = " + e1.eval());  
        System.out.println(e2 + " = " + e2.eval());  
    }  
}
```

פלט התוכנית הוא:

$(3.0) ^ {((1.0) + (2.0))} = 27.0$

$((1.0) + (2.0)) ? ((1.0) * (2.0)) : ((2.0) ^ (3.0)) = 2.0$

הערות:

- כל אחת מ-8 הישויות לעיל תכיל את המתודה:

```
public double eval()
```

- כל הפעולות לעיל פועלות על ביטויים (Expression), שערך של ביטוי מתבצע ע"י הפונקציה eval
- ביטויים מורכבים ישוערכו ע"י הפעלה רקורסיבית של eval על הארגומנטים
- תשובתך צריכה לבטא **שימוש חוזר** ברכיבי תוכנה ולמזער את **שכפול הקוד**. הדבר יעשה, בין השאר, ע"י בחירה נכונה של מחלקות קונקרטיות, מחלקות מופשטות ומנשקים. קוד עובד הוא תנאי הכרחי אך לא מספיק במקרה זה.
- שימו לב למתודה toString ולהדפסות הסוגריים. תזכורת: כאשר אופרטור ה- '+' ופונקציית ההדפסה מוצאים עצם שאינו String במקום שבו אמור היה להימצא String, מופעלת המתודה toString() של אותו העצם. במחלקה Object קיים מימוש ברירת מחדל של מתודה זו.

הציגו תרשים מחלקות המתאר את היחס בין הישויות שהגדרתם (מכונה גם עץ הורשה או היררכית מחלקות).

המתכנתת אוֹמָה כתבה זה עתה את המחלקה MyCar המציגה פרטים של מכוניות שונות.
להלן המחלקה וכן דוגמא לשימוש במחלקה:

```
public class MyCar {

    public static final int LAMBORGHINI = 2;

    public static final int SUSITA = 4;

    public static final int RENAULT = 5;


    private int type;

    public MyCar(int type) {

        this.type = type;

    }

    public String getManufacturerName() {

        switch (type) {

            case LAMBORGHINI:
                return "LAMBORGHINI";

            case RENAULT:
                return "RENAULT";

            case SUSITA:
                return "SUSITA";

            default:

                return "Manufacturer Unknown";

        }

    }

    public int getNumOfDoors() {

        switch (type) {

            case LAMBORGHINI:
                return LAMBORGHINI;

            case RENAULT:
                return RENAULT;

            case SUSITA:
                return SUSITA;

            default:

                return -1;

        }

    }

}
```

דוגמא לשימוש במחלקה:

```
public class MyCarDemo {

    public static void main(String[] args) {

        MyCar bimba = new MyCar(MyCar.LAMBORGHINI);

        System.out.println("bimba was manufactured by " +
            bimba.getManufacturerName());

        System.out.println("bimba has " + bimba.getNumOfDoors() +
            " door" + (bimba.getNumOfDoors() == 1 ? "" : "s"));

    }

}
```

א.

1) צייני את הבעיה הגדולה ביותר לדעתך במחלקה MyCar. אם יש בעיות נוספות צייני גם אותן. מדוע לדעתך הקוד של אומה לוקה בבעיות שציינת?

2) עזרי לאומה לשכתב את המחלקה בעזרת ירושה בשני אופנים שונים כדי לפתור את הבעיות שציינת בסעיף הקודם. ממשי בקוד את כל רכיבי התוכנה.

- i. מימוש אחד צריך למזער את כמות הקוד (רמז: מחלקת אב משותפת)
- ii. המימוש האחר צריך למזער את הזיכרון הדרוש למופעי המחלקה (רמז: ע"י דריסה)

ב.

1) צייני את הבעיה הגדולה ביותר העולה מפתרון הסעיף הקודם בעזרת ירושה. אם יש בעיות נוספות צייני גם אותן. מדוע ירושה במקרה זה גורמת לבעיות שציינת? (רמז חלקי: האם יש הבדל בהתנהגות בין מחלקות של מכוניות שונות?)

2) עזרי לאומה לשכתב את המחלקה MyCar ע"י שימוש במחלקה אחת בלבד (ללא ירושה) כדי לפתור את הבעיות שציינת בסעיף הקודם. ממשי בקוד את המחלקה במלואה.

ג. שכתבי את המחלקה MyCar ע"י הגדרתה כ enum.

ד. האם קיימות נסיבות (כגון אילוצים על המחלקה, הקשרי שימוש, דרישות עיצוב של המערכת וכו') שבהן יש העדפה לאסטרטגיית מימוש מסוימת על פני אחרות (מבין 5 הגישות לעיל כפי שעולה מהמחלקה המקורית ו-4 החלופות)? דוני בקצרה במקרים השונים.