

תוכנה 1 – סתיו תשע"ג

תרגיל מספר 6

הנחיות כלליות:

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

- הגשת התרגיל תעשה במערכת ה moodle בלבד (<http://moodle.tau.ac.il/>).
- יש להגיש קובץ zip יחיד הנושא את שם המשתמש ומספר התרגיל (לדוגמא, עבור המשתמש zvainer יקרא הקובץ zvainer_hw6.zip). קובץ ה-zip יכיל:
 - א. קובץ פרטים אישיים בשם details.txt המכיל את שמכם ומספר ת.ז.
 - ב. קבצי ה-java של התוכניות אותם התבקשתם לממש.
 - ג. קובץ טקסט אחד עם העתק של כל קבצי ה-java.
 - ד. קובץ answers עם התשובות לשאלות

חלק א' (45 נק')

בחלק זה נתרגל כתיבה של מחלקות המממשות מנשק.

המנשק IDate2013 המופיע למטה מייצג תאריך (יום וחודש) בשנת 2013. אנו נכתוב עבור המנשק שלושה מימושים מקובעים המתבססים על ייצוג פנימי שונה.

1. DayMonthDate2013 אשר מכילה שני שדות בדיוק, אחד מייצג את החודש (מקבל ערך בין 1 ל-12), והשני מייצג את היום בחודש (מקבל ערך בין 1 למס' הימים באותו חודש).
2. StringDate2013 אשר מכילה שדה אחד בדיוק, המייצג את המחרוזת בפורמט YYYY-MM-DD, לדוגמא, "2013-05-30".
3. DaysDate2013 אשר מכילה שדה אחד בדיוק, המייצג את היום בשנה (מקבל ערך בין 0 ל-364).

עבור כל אחת מן המחלקות כבר נתון שלד. עליכם:

1. לכתוב בנאי, אותו תגדירו כרצונכם.
2. לממש את מתודות המנשק. שימו לב: עליכם לעבוד בכל מחלקה עם הייצוג הפנימי שלה כדי לחשב את הערך המוחזר, גם על חשבון חיסכון בקוד. לדוגמא, ב- StringDate2013 נשווה תאריכים בעזרת מחרוזות, ואילו ב- DaysDate2013 נשווה תאריכים בעזרת מספרי היום בשנה. ניתן לעבור מייצוג אחד לשני רק לצורך פלט. למשל, כדי להחזיר מחרוזת המייצגת תאריך. בשום מקרה אסור למחלקה אחת ליצור מופע של מחלקה אחרת או להשתמש בקוד שלה!
3. לממש את המתודה הסטטית המתאימה ב- Date2013Factory אשר יוצרת מופע של המחלקה המתאימה.

מותר ורצוי:

- להשתמש בקבועים הנתונים במנשק.
- להניח כי הקלט לבנאי מייצג תאריך תקין ב-2013. עם זאת, כאשר אתם קוראים לבנאים מתוך הקוד עליכם לוודא שאתם מעבירים קלט תקין.
- כאמור, מטרתנו אינה לחסוך בקוד. במקרים בהם אין ברירה ונוצר ממש שכפול קוד בין המחלקות, ניתן למנוע את השכפול בעזרת מחלקת עזר נפרדת עם שירותי עזר סטטיים.
- אין חובה לכתוב חוזה, מכיוון שהחוזה כבר נתון או נובע מתיעוד המנשק. אם בכל זאת אתם רואים צורך בהבהרת הנחה מסוימת שעשיתם בקוד, ניתן להוסיף חוזה.

אסור:

- לשנות את המנשק הנתון.
- לשנות את הגדרות השדות בשלדי המחלקות הנתונים או את ערכי השדות (מחלקות מקובעות).
- להוסיף שדות (כולל סטטיים) למחלקות הנתונות. אם יש צורך בקבועים ניתן להוסיפם במחלקת עזר.
- כאמור, לקרוא ממחלקה אחת למחלקה אחרת מבין ה-3 שמממשות את המנשק, כולל יצירת מופע.
- ליצור תאריכים לא חוקיים.
- להשתמש במחלקה קיימת שעובדת עם תאריכים כגון `java.util.Date`.

הסבר על פעולתה של כל מתודה נמצא בתיעוד המנשק, קראו הסבר זה בעיון. המנשק ושלדי המחלקות נמצאים באתר הקורס.

```
package date2013;

public interface IDate2013 {

    final int[] DAYS_IN_MONTHS = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31,
        30, 31 }; //the i'th cell contains the number of days in
        //month i+1
    static final char SEPRARTOR = '-'; // the delimiter for the date string
    static final int YEAR2013 = 2013;

    /**
     * Returns the date in format YYYY-MM-DD, for example 2013-05-31
     */
    String toString();

    /**
     * Returns the day of month for this date, ranging from 1 to the number
     * of days in that month.
     * E.g., for 2013-05-31 the method returns 31
     */
    int getDayOfMonth();

    /**
     * Returns the day of year for this date, ranging from 0 to 364.
     * E.g., for 2013-05-31 the method returns 150
     */
    int getDayOfYear();
}
```

```
/**
 * Returns the month for this date, ranging from 1 to 12.
 * E.g., for 2013-05-31 the method returns 5
 */
int getMonth();

/**
 * Returns a date that is one day after this date.
 * If the requested date exceeds the last day of the year, it returns
 * null.
 * E.g., for 2013-05-31 the method returns a date that represents
 * 2013-06-01
 */
IDate2013 advanceOneDay();

/**
 * Returns a date that is one month after this date.
 * If the requested date exceeds the last day of the year, it returns
 * null.
 * If the requested day does not exist in the next month, it returns
 * the last day of the next month
 * E.g., for 2013-05-31 the method returns a date that represents
 * 2013-06-30
 */
IDate2013 advanceOneMonth();

/**
 * Returns true iff other is not null and this date comes before or
 * equal to the other date
 * E.g., for 2013-05-31 and other representing 2013-07-01 it returns
 * true
 */
boolean beforeOrEquals(IDate2013 other);
}
```

חלק ב' (45 נק')

בחלק זה נתרגל כתיבת מחלקות המממשות מנשק נתון. לאחר מכן, נשתמש בהן בעזרת תכונת הפולימורפיזם של תכנות מונחה עצמים ב-Java.

נתון המנשק IShape:

```
public interface IShape {

    // Returns the shape's area
    public float getArea();

    // Returns the shape's perimeter
    public float getPerimeter();

    // Returns a string representation of the shape.
    // The returned string would include the shape's name, its fields
    // and their corresponding values.
    public String toString();

}
```

1. עליכם לכתוב שלוש מחלקות המממשות את המנשק IShape. להלן פירוט חתימת הבנאי של כל אחת מהן (כל בנאי מקבל כפרמטרים את הנתונים הנדרשים להגדרת הצורה אותה הוא יוצר):

```
// x and y are the coordinates of the centre
public Circle (int x, int y, int radius)

// x and y are the coordinates of the bottom left corner
public Rectangle (int x, int y, int width, int height)

// x and y are the coordinates of the bottom left corner
public Square (int x, int y, int width)
```

- על כל מחלקה להכיל שדות פרטיים אליהם ניתן לגשת עם מתודות getter ו-setter ציבוריות.
- כל אחת משלוש המחלקות מממשת את המנשק IShape ועל כן צריכה לממש את המתודות המוגדרות בו.
- המתודות getArea ו-getPerimeter יחזירו את שטח והיקף הצורה בהתאמה (שימו לב שלצורך החישוב יש להשתמש בקבוע Math.PI לקבלת הערך פאי).
- המתודה toString תחזיר מחרוזת המייצגת את הצורה ותכיל את שם הצורה, ואת כל השדות שבה לצד הערכים שלהם. דוגמא למחרוזת שתוחזרנה עבור מעגל, מלבן וריבוע בהתאמה:

"Circle: X=100, Y=150, Radius=30"

"Rectangle: X=100, Y=150, Width=50, Height=50"

"Square: X=100, Y=150, Width=50"

2. כתבו תוכנית בשם ShapeDimensionCalculator המקבלת כארגומנט יחיד בשורת הפקודה מסלול קובץ שימש לפלט. התוכנית תקבל מהמשתמש (דרך ה-`System.in`) סדרה של נתוני צורות, תחשב עבור כל צורה את שטחה ואת היקפה, ולבסוף תדפיס דו"ח מסכם לקובץ הפלט.

עבור כל צורה שנבחרה ע"י המשתמש, תבקש התוכנית את הנתונים הנדרשים לבנייתה.

דוגמא להרצת התוכנית:

עבור שורת הפקודה הבאה: (קלט משתמש בכחול)

```
java ShapeDimensionCalculator output.txt
```

```
Shape Dimension Calculator

Please choose shape type:
C - Circle
S - Square
R - Rectangle
E - End
C
Please enter X coordinate:
100
Please enter Y coordinate:
150
Please enter radius:
30
Shape added: [Circle: X=100, Y=150, Radius=30]

Please choose shape type:
C - Circle
S - Square
R - Rectangle
E - End
S
Please enter X coordinate:
200
Please enter Y coordinate:
230
Please enter width:
50
Shape added: [Square: X=200, Y=230, Width=50]

Please choose shape type:
C - Circle
S - Square
R - Rectangle
E - End
E

Report saved to file output.txt.
```

לאחר ריצת התוכנית כמודגם לעיל יכיל קובץ הפלט output.txt את הטקסט הבא:

```
Shape Dimension Calculator
Circle: X=100, Y=150, Radius=30   Area: 2827.4333882308138   Perimeter: 188.49555921538757
Square: X=200, Y=230, Width=50   Area: 2500   Perimeter: 200
Total Number of Shapes: 2
Total Area Sum: 5327.433388230814
Total Perimeter Sum: 388.4955592153876
```

הערות:

- המשתמש יכול להכניס 20 צורות לכל היותר. לאחר הכנסת הצורה ה-20 התוכנית תצא אוטומטית משלב הכנסת הנתונים ותייצר את הדו"ח המסכם לקובץ הפלט.
- ניתן להניח שהמשתמש מכניס קלט חוקי (אותיות או מספרים שלמים כנדרש).
- יש לאחסן את הצורות במערך מסוג IShape ולבצע את החישובים בגמר קליטת הנתונים.
- בקובץ הפלט יש 3 עמודות עבור כל צורה המופרדות ע"י טאבים. העמודה הראשונה היא תוצר של מתודת ה-`toString` של הצורה, השניה מציגה את השטח והשלישית את ההיקף.

3. להלן מנשק נוסף בשם `IMovable`:

```
public interface IMovable {
    // Moves the shape by the given x and y offsets
    public void move (int xOffset, int yOffset);
}
```

המתודה `move` מקבלת היסט X והיסט Y ומוסיפה אותם לערכי ה-X וה-Y של הצורה.

- שנו את המחלקות `Circle`, `Rectangle` ו-`Square` כך שיממשו גם את המנשק `IMovable` ע"י מימוש המתודה `move` בכל אחת מהמחלקות.
- כתבו מתודה בשם `moveShapes` המקבלת מערך צורות המקיימות את המנשק הנ"ל, היסט X והיסט Y. המתודה תזיז את מיקומי כל הצורות במערך לפי ערכי היסט הנתונים. יש לממש את המתודה בתוך המחלקה `ShapeDimensionCalculator`. חתימת המתודה תהיה:

```
public void moveShapes(IMovable[] movableShapes, int xOffset, int yOffset);
```

שימו לב: בחלק זה יש להגיש את הקבצים הבאים (ניתן להוסיף קבצי עזר בהם השתמשתם):

ShapeDimensionCalculator.java, Circle.java, Rectangle.java, Square.java

חלק ג' (10 נק')

נתון קטע הקוד הבא:

```

1 String s = "This is my String";
2 s = s.replaceAll("is", "in");
3 s = s + "ggg";
4 s = new String("bbb" + "fff" + "nnn");
5 s = "bb" + "bfffnnn";

```

להלן מס' שאלות הנוגעות להפנמת מחרוזות ולמודל הזיכרון. ענו על השאלות הבאות ב"נכון" או "לא נכון" בקובץ answers.

1. אחרי שורה 1, המחרוזת "This is my String" נמצאת במאגר המחרוזות
2. אחרי שורה 2, המחרוזת "This is my String" נמצאת במאגר המחרוזות
3. אחרי שורה 2, המחרוזת "Thin in my String" נמצאת במאגר המחרוזות
4. אחרי שורה 2, המחרוזת "is" נמצאת במאגר המחרוזות
5. אחרי שורה 2, ייתכן שהמחרוזת "This is my String" נמצאת ב-Heap
6. אחרי שורה 3, s מצביע למחרוזת שנמצאת במאגר המחרוזות
7. אחרי שורה 3 יש ב-stack 3 משתנים
8. אחרי שורה 4 המחרוזת "bfffnnn" נמצאת במאגר המחרוזות
9. אחרי שורה 4 s מצביע למחרוזת שנמצאת במאגר המחרוזות
10. אחרי שורה 5 s מצביע למחרוזת שנמצאת במאגר המחרוזות