מבוא לתכנות מונחה עצמים מטלה 2

- תרגיל זה נעשה בזוגות.
- את התרגיל יש להגיש בנוהל ההגשה הרגיל לאתר של מידע אישי שלכם.
- שם הקובץ הוא שמם המלא ות.ז. של שני חברי הקבוצה.
- סוג הקבץ – zip או rar.

חלק א' משוואה ריבועית – שימוש בחריגות

בחלק זה על המשתמש יש לפתור משוואה ריבועית: $ax^2+bx+c=0$.

א) יש להזין שלושה מספרים ממשיים (a,b,c), שמייצגים את מקדמי משוואה ריבועית.

ב) יש לבדוק האם הפתרון קיים, כלומר קיימים שני מספרים ממשיים (לא בהכרח שונים) שמהווים את פתרון המשוואה.

ג) אם יש שני פתרונות שונים (ממשיים), יש להדפיס: x_1=..., x_2=...

אם קיים פתרון יחיד למשוואה, יש להדפיס: $x_1=x_2$=...

ד) כאשר למשוואה אין פתרון יש לטפל במצב זה ולזרוק חריגה מטיפוס SquareEquationException (חריגה של המשתמש) עם הודעה מתאימה:

a. כאשר ביטוי בתוך השורש שלילי יש להדפיס:

Error: NO real roots!

b. כאשר a=0, b=0, c=0 יש להדפיס:

x can be any number - trivial!

c. כאשר a=0, b=0, c≠0 יש להדפיס:

Error, no answer!!

הבהרה: ההודעות המתקבלות חייבות להיות זהות לחלוטין לנוסח המצוין לעיל.

מעטפת התוכנית. עליכם לכתוב את המסגרת שתציג למשתמש תפריט לבחירתו:

0 – יציאה מהתוכנית (או כל מספר אחר ששונה מ 1).

1 – לקליטת נתונים וחישוב פתרונות של משוואה ריבועית.

לאחר סיום החישובים וטיפול בחריגות על התוכנית לחזור ולהציג את התפריט ההתחלתי.

תוצאות ההרצה:

```
aX^2+bX+c=0: Enter a,b,c:
Enter a: -2.3
Enter b: 5.1
Enter c: -12.62
-2.3X^2+5.1X+-12.62=0:
matala_2Exceptions.SquareEquationException: Error: NO real roots!
Enter 0 or any number to Exit or 1 to solve aX^2+bX+c=0:
      at matala_2Exceptions.SquareEquation.sqEq(SquareEquation.java:28)
      at matala_2Exceptions.SquareEquation.sqEquation(SquareEquation.java:44)
      at matala_2Exceptions.SquareEquation.main(SquareEquation.java:55)
1
aX^2+bX+c=0: Enter a,b,c:
Enter a: -2.3
Enter b: 5.1
Enter c: 12.98
-2.3X^2+5.1X+12.98=0:
x1:-1.5128848463076623    x2:3.730276150655489
Enter 0 or any number to Exit or 1 to solve aX^2+bX+c=0:
1
aX^2+bX+c=0: Enter a,b,c:
```

```
Enter a: 1
Enter b: -5
Enter c: 6
1.0X^2+-5.0X+6.0=0:
x1:3.0    x2:2.0
Enter 0 or any number to Exit or 1 to solve aX^2+bX+c=0:
1
aX^2+bX+c=0: Enter a,b,c:
Enter a: 1
Enter b: -2
Enter c: 1
1.0X^2+-2.0X+1.0=0:
x1=x2:1.0
Enter 0 or any number to Exit or 1 to solve aX^2+bX+c=0:
1
aX^2+bX+c=0: Enter a,b,c:
Enter a: 0
Enter b: 2
Enter c: 5
0.0X^2+2.0X+5.0=0:
x1=:-2.5
Enter 0 or any number to Exit or 1 to solve aX^2+bX+c=0:
1
aX^2+bX+c=0: Enter a,b,c:
Enter a: 0
Enter b: 0
Enter c: 0
0.0X^2+0.0X+0.0=0:
Enter 0 or any number to Exit or 1 to solve aX^2+bX+c=0:
matala_2Exceptions.SquareEquationException: x1 can be any number - trivial!
      at matala_2Exceptions.SquareEquation.sqEq(SquareEquation.java:17)
      at matala_2Exceptions.SquareEquation.sqEquation(SquareEquation.java:44)
      at matala_2Exceptions.SquareEquation.main(SquareEquation.java:55)
1
aX^2+bX+c=0: Enter a,b,c:
Enter a: 0
Enter b: 0
Enter c: 3
0.0X^2+0.0X+3.0=0:
Enter 0 or any number to Exit or 1 to solve aX^2+bX+c=0:
matala_2Exceptions.SquareEquationException: Error, no answer!!
      at matala_2Exceptions.SquareEquation.sqEq(SquareEquation.java:16)
      at matala_2Exceptions.SquareEquation.sqEquation(SquareEquation.java:44)
      at matala_2Exceptions.SquareEquation.main(SquareEquation.java:55)
1
aX^2+bX+c=0: Enter a,b,c:
Enter a: 1
Enter b: 0
Enter c: 0
1.0X^2+0.0X+0.0=0:
x1=x2:0.0
Enter 0 or any number to Exit or 1 to solve aX^2+bX+c=0:
0
Bye-bye!
```

חלק ב' – מחלקות גנריות, איטרטורים, קומפרטורים

בחלק זה יש לכתוב מערכת המנהלת ציי רכבים. מחלקה מרכזית במערכת מייצגת את היישות העסקית "כלי רכב" ונראית כך:

```
package carManagmentSystem;
public class Vehicle {
    private int totalValue;          // שווי
    private int licencePlate;    // מספר רישוי רכב
           private String ownerName;       // שם בעלים

    public Vehicle(int totalValue,int licencePlate,String ownerName) {
        this.totalValue=totalValue;
        this.licencePlate = licencePlate;
        this.ownerName=ownerName;
    }
. . . . . .
}
```

* במידת ומצאתם לנכון הוסיפו Getters/Setters toString למחלקה.

כמוכן צי הרכבים כולל את המכוניות הבאות:

מספר רישוי	שווי	שם בעלים
5079930	240000	אבירם כוהן
3049377	98000	אביבה לוי
5079930	56000	חדווה קדרון
2023078	89000	אלי קופטר
9599	310000	אבי רון

ממש\י את ממשק `Comparable<T>` כך שמיון סט נתונים מסוג List<Vehicle> יתבסס על תכונת השווי בסדר *יורד*. וכן כתב\י פונקציה לבדיקה המקבלת סט כזה, ממיינת ומציגה אותו. (ניתן לכתוב תוכנית בדיקה גם דרך ה - main)

<u>דוגמה לבדיקת מחלקת</u> Vehicle:

```
public static void main(String[] args) {
      List<Vehicle> cars=new ArrayList<>();
      cars.add(new Vehicle(240000,5079930,"כוהן אבירם" ));
      cars.add(new Vehicle(98000,3049377,"לוי אביבה" ));
      cars.add(new Vehicle(56000,5079930,"קדרון חדווה" ));
      cars.add(new Vehicle(89000,20230786,"קופטר אלי"));
      cars.add(new Vehicle(310000,9599,"רון אבי"));
      Collections.sort(cars);
      for (Vehicle v : cars) {
            System.out.println(v);
      }
}
```

פלט:

```
totalValue: 310000, 9599, ownerName :רון אבי
totalValue: 240000, 5079930, ownerName :כוהן אבירם
totalValue: 98000, 3049377, ownerName :לוי אביבה
totalValue: 89000, 20230786, ownerName :קופטר אלי
totalValue: 56000, 5079930, ownerName :קדרון חדווה
```

א. הגדר\י מחלקה המייצגת צי רכב בשם CarFleet הכוללת:

1) תכונה בשם OrganizationName מסוג מחרוזת המתייחסת לשם הארגון.
2) תכונה בשם Cars מסוג List<Vehicle> המייצגת רשימת כלי רכב.
3) פונקציית בנאי המקבלת את שם הארגון ומאתחלת את המשתנים הנ"ל.
4) פונקציה בשם Add המוסיפה כלי רכב.
5) מחלקה פנימית בשם CarIterator המממשת את תבנית Iterator באמצעות ממשק `Iterator<T>` עבור Cars.
6) מתודה בשם iterator המחזירה אובייקט מסוג `Iterator<Vehicle>`.

ב. כתוב\י קוד המאתחל את CarFleet לרבות הוספת מכוניות, ומשתמש ב- Iterator לסריקת רשימת הרכבים בארגון ומציג את הרכבים ששם בעליהם כולל את תת המחרוזת "אבי".

דוגמה להרצת התכנית

```
public static void main(String[] args) {
            CarFleet fleet=new CarFleet ("Avis");
            fleet.Add(new Vehicle(240000,5079930,"כוהן אבירם" ));
            fleet.Add(new Vehicle(98000,3049377,"לוי אביבה" ));
            fleet.Add(new Vehicle(56000,5079930,"קדרון חדווה" ));
            fleet.Add(new Vehicle(89000,20230786,"קופטר אלי"));
            fleet.Add(new Vehicle(310000,9599,"רון אבי"));
            Iterator<Vehicle> it=fleet.iterator();
            Vehicle v=null;
            while (it.hasNext()) {
                  System.out.println(it.next());
            }
            System.out.println();
            it=fleet.iterator();
            while (it.hasNext()) {
                  v=it.next();
                  if(v.getOwnerName().contains("אבי"))
                        System.out.format("Car Licence %d costs %d%n",
                                    v.getLicencePlate(), v.getTotalValue());
            }
      }
```

פלט:

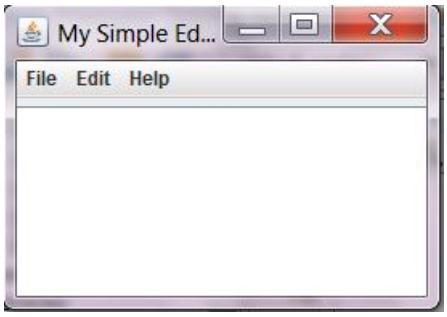
```
totalValue: 240000, 5079930, ownerName :כוהן אבירם
totalValue: 98000, 3049377, ownerName :לוי אביבה
totalValue: 56000, 5079930, ownerName :קדרון חדווה
totalValue: 89000, 20230786, ownerName :קופטר אלי
totalValue: 310000, 9599, ownerName :רון אבי

Car Licence 5079930 costs 240000
Car Licence 3049377 costs 98000
Car Licence 9599 costs 310000
```

חלק ג' - Graphic User Interface - ממשק גרפי

חלק ג' – Simple Editor , כתבן פשוט

בחלק זה עליך לממש כתבן פשוט בדומה ל- Notepad.
יש לצרף למטלה קובץ JAR ו - icon(שבו אתם משתמשים בחלק זה במטלה) .

הכתבן מכיל MenuBar עם שלושה תפריטים: File, Edit ו- Help



תפריט **File** מכיל שלוש אופציות:

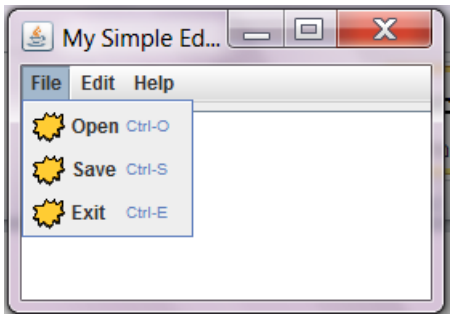
- אופציה לשמירת (Save) קובץ טקסט על דיסק ע"י FileSaveDialod,
- אופציה לקראית קובץ מדיסק ע"י FileOpenDialod
- ואופציית יציאה מהתכנית. לכל אופציה צריך לצרף קיצור דרך ותמנה קטנה מתאימה:

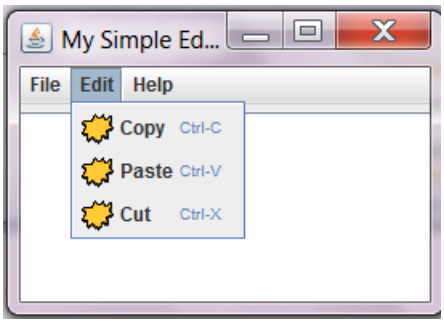
תפריט **Edit** מכיל שלוש אופציות:

- אופציה העתקת טקסט
- אופציה הדבקת טקסט
- אופציה לגזירת טקסט

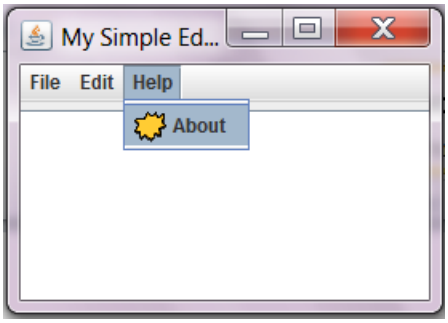
לכל אופציה צריך לצרף קיצור דרך ותמנה קטנה מתאימה:
בכל האופציות הנ"ל מאחר והם קיצורים מוגדרים במחשב נשתמש :

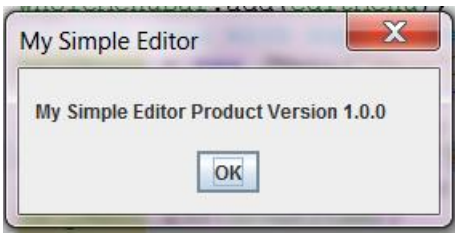
- Ctrl+A – להעתקה
- Ctrl+B- להדבקה
- Ctrl+H – לחיתוך.
- Ctrl+L – לפתיחה.
- Ctrl+M – לשמירה.
- Ctrl+G – ליציאה.



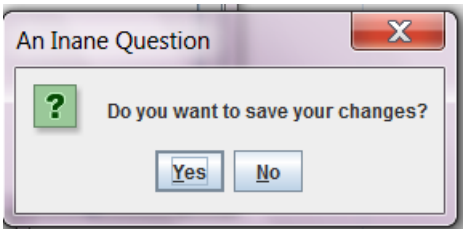


תפריט **Help**: בלחיצה על About צריך לקבל את ההודעה הבאה:





לפני יציאה מכתבן התכנית מציעה לשמור את הקובץ ע"י ההודעה הבאה:



להרצה תקינה של JAR צריך לשמור את ICON באותה תיקייה.

תיהנו !