

חלק 1 – ממשקים – INTERFACE

(א) יש להגדיר מחלקה COMPLEX המציגה מספרים מרוכבים. המחלקה מכילה תכונות ושיטות הבאות:

(a - חלק ממשי של מספר מרוכב, b - חלק מדומה של מספר מרוכב)

```
public class Complex{  
    double a, b;
```

```
// בנאי
```

```
    public Complex(double a, double b){...}
```

♦ שיטה לחישוב מודול של מספר מרוכב $z = (a, b)$, לפי הנוסחה $||z| = \sqrt{a^2 + b^2}$

```
    public double module(){ . . }
```

♦ שיטה לחישוב סכום של שני מספרים מרוכבים $z_1 = (a_1, b_1)$, $z_2 = (a_2, b_2)$ לפי הנוסחה:

```
//  $z_1 + z_2 = (a_1 + a_2, b_1 + b_2)$ 
```

```
    public void add (Complex z){ . . .}
```

♦ שיטה לחישוב כפל של שני מספרים מרוכבים $z_1 = (a_1, b_1)$, $z_2 = (a_2, b_2)$, לפי הנוסחה:

```
//  $z_1 \cdot z_2 = (a_1 a_2 - b_1 b_2, a_1 b_2 + a_2 b_1)$ 
```

```
    public void mul (Complex z){ . . .}
```

♦ נגדיר השוואה של שני מספרים מרוכבים לפי המודולים שלהם: מספר z_1 קטן ממספר z_2 אם

$|z_1| < |z_2|$, מספרים z_1 ו- z_2 שווים אם שווים המודולים שלהם $|z_1| = |z_2|$.

שיטה להשוואה של שני מספרים מרוכבים מחזירה:

-1, כאשר מספר Z גדול ממספר שממנו השיטה מופעלת,

0, כאשר מספר Z שווה ממספר שממנו השיטה מופעלת,

+1, כאשר מספר Z קטן ממספר שממנו השיטה מופעלת,

```
    public int compare(Complex z){ . . .}
```

(ב) נתון ממשק Sortable

```
interface Sortable {
    int compare(Object left, Object right);
    Object valueAt(int position);
    void setValue(Object value, int position);
    int size();
} // interface Sortable
```

יש לכתוב מחלקה **SortComplex** אשר מממשת את הממשק הנתון. המחלקה מכילה מערך של מספרים מרוכבים, והבנאי מקבל מערך של מספרים מרוכבים כפרמטר:

```
private Complex[] cArr;

public SortComplex(Complex[] c){.....}
```

(ג) יש להגדיר מחלקה **ObjectSort** המכילה שתי שיטות:

- שיטה סטטית `public static void sort(Sortable item)` המקבלת עצם ממחלקה אשר מממשת את הממשק **Sortable** וממיינת אותו.
- שיטה סטטית `public static boolean checkSort(Sortable item)` המקבלת עצם ממחלקה אשר מממשת את הממשק **Sortable** והמחזירה **true** כאשר המערך ממורן, אחרת היא מחזירה **false**.

(ד) לאחר מכן יש להגדיר מחלקת **TestSort** עם **main** אשר יוצרת עצם ממחלקה **SortComplex** ממיינת אותו באמצעות שיטה **SortObject.sort** ובודקת האם המערך ממורן.

הערה: למיון מערך של מספרים שלמים יש להשתמש בחד ממיונים מהירים שלמדנו בקורס קודם:

נספח: תוכנית TEST:

```
public class TestSort {
    public static void main(String[] args) {
        Complex []c = new Complex[10];
        for(int i=0; i<c.length; i++){
            double a = Math.random()*c.length;
            double b = Math.random()*c.length;
            c[i]= new Complex(a,b);
        }
        SortComplex sc = new SortComplex(c);
        SortObject.sort(sc);
        System.out.println("is sorted: "+SortObject.checkSort(sc));
        for(int i=0; i<c.length; i++){
            System.out.println(((Complex)sc.valueAt(i)).module());
        }
        System.out.println();
    }
}
```

חלק 2 – Exceptions

תרגיל זה כבר עשיתם בקורס מבוא לחישוב. עכשיו עליכם להוסיף מחלקה `SquareEquationException` שתטפל בקלט לא תקין.

ובכל מקרה שהפתרון אינו קיים צריך לזרוק חריגה עם הודעה מתאימה ולחזור לקליטת מקדמים.

חלק ראשון:

על המשתמש להזין שלושה מספרים ממשיים (a,b,c) , שמייצגים את מקדמי משוואה ריבועית:

$$ax^2 + bx + c = 0$$

לאחר מכן על התוכנית להדפיס:

אם יש שני פתרונות שונים (ממשיים) את שני הפתרונות: $x1=...$, $x2=...$

אם קיים פתרון יחיד למשוואה: $x1=x2=...$

אם למשוואה אין פתרון צריך לזרוק חריגה עם הודעה מתאימה. תכנית בדיקה:

```
public static void main(String[] args) {  
    SquareEquation.sqEquation();  
}
```

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:  
SquareEquationException: Error: NO real roots!  
Enter 0 or any number to Exit or 1 to solve aX^2+bX+c=0:  
    at SquareEquation.sqEq(SquareEquation.java:26)  
    at SquareEquation.sqEquation(SquareEquation.java:42)  
    at SquareEquation.main(SquareEquation.java:53)  
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: 3
 $0.0X^2+0.0X+3.0=0$ :
Enter 0 or any number to Exit or 1 to solve  $aX^2+bX+c=0$ :
SquareEquationException: Error, no answer!!
    at SquareEquation.sqEq(SquareEquation.java:14)
    at SquareEquation.sqEquation(SquareEquation.java:42)
    at SquareEquation.main(SquareEquation.java:53)
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  $aX^2+bX+c=0$ :
SquareEquationException: x1 can be any number - trivial!
    at SquareEquation.sqEq(SquareEquation.java:15)
    at SquareEquation.sqEquation(SquareEquation.java:42)
    at SquareEquation.main(SquareEquation.java:53)
0
Ex3b - Done!

```

חלק שני:

מעטפת התוכנית, בחלק זה עליכם לכתוב את המסגרת: שתציג למשתמש תפריט לבחירתו:
 0 – יציאה מהתוכנית (או כל מספר אחר ששונה מ 1).
 1 – סעיף ראשון (חישוב של פתרונות משוואה ריבועית)
 לאחר סיום הסעיף (הראשון) על התוכנית לחזור ולהציג את התפריט ההתחלתי.