

הגדרת מחלקות ויצירת אובייקטים

class

קובץ שמגדיר טיפוס

יש להגדיר את הדברים הבאים:

1. **מאפיינים** – להגדרת מאפיינים של האובייקט
2. **בנאים** – פונקציות מיוחדות שמשמשות ליצירת אובייקטים מהמחלקה
3. **מתודות** – פונקציות שמגדירות את הפעולות של האובייקטים אחרי שנוצרו

דוגמה:

הגדרת מחלקה שמייצגת נקודה (אובייקט שמציין מיקום – על המפה, על מסך וכדומה)

מאפיינים:

x – מיקום על ציר ה X

y – מיקום על ציר ה Y

בנאים:

בנאי ליצירת נקודה במיקום (0, 0)

בנאי ליצירת נקודה במיקום אחר

מתודות:

הדפסת מיקום

הזזת הנקודה ימינה

הזזת הנקודה שמאלה

הזזת הנקודה למטה

הזזת הנקודה למעלה

איפוס הנקודה

הגדרת המחלקה:

```
Point.java ×
1 package d.oop.points;
2
3 public class Point {
4
5 }
```

הגדרת מאפיינים

```
Point.java X
1 package d.oop.points;
2
3 public class Point {
4
5     // 1. attributes
6     int x;
7     int y;
8
9 }
```

הגדרת בנאים

```
Point.java X
1 package d.oop.points;
2
3 public class Point {
4
5     // 1. ATTRIBUTES
6     int x;
7     int y;
8
9     // 2. CONSTRUCTORS
10
11     // constructor 1 - function for object creation
12     Point() {
13
14     }
15
16     // constructor 2 - function for object creation
17     Point(int x, int y) {
18         this.x = x;
19         this.y = y;
20     }
21
22 }
23
```

שימוש בבנאים ליצירת אובייקטים יראה כך:

```
public static void main(String[] args) {
    Point p1 = new Point(); // create a point at (0, 0)
    Point p2 = new Point(2, 4); // create a point at (2, 4)

    System.out.println(p1.x + ", " + p1.y);
    System.out.println(p2.x + ", " + p2.y);
}
```

```
// 3. METHODS - defines what actions a point can do

void printLocation() {
    System.out.println("(" + x + ", " + y + ")");
}

void moveRight() {
    x++;
}

void moveLeft() {
    x--;
}

void moveUp() {
    y++;
}

void moveDown() {
    y--;
}
```

שימוש במתודות מתוך תכנית

```
public static void main(String[] args) {

    // create 2 Point instances (objects)
    // to create the instances we call a constructor
    Point p1 = new Point();
    Point p2 = new Point(5, 3);

    // use the instances - call methods
    p1.printLocation();
    p2.printLocation();

    System.out.println("=====");
    // print the current location
    p1.printLocation();
    // move right 3 steps
    p1.moveRight();
    p1.moveRight();
    p1.moveRight();
    // move up 2 steps
    p1.moveUp();
    p1.moveUp();
    // print the current location
    p1.printLocation();
}
```

OOP Principles

1. Encapsulation – הסתרת המידע הרגיש של ה class באמצעות המילה השמורה private ומתן מגישה באמצעות מתודות get/set
2. inheritance
3. polymorphism

Libraries – ספריות

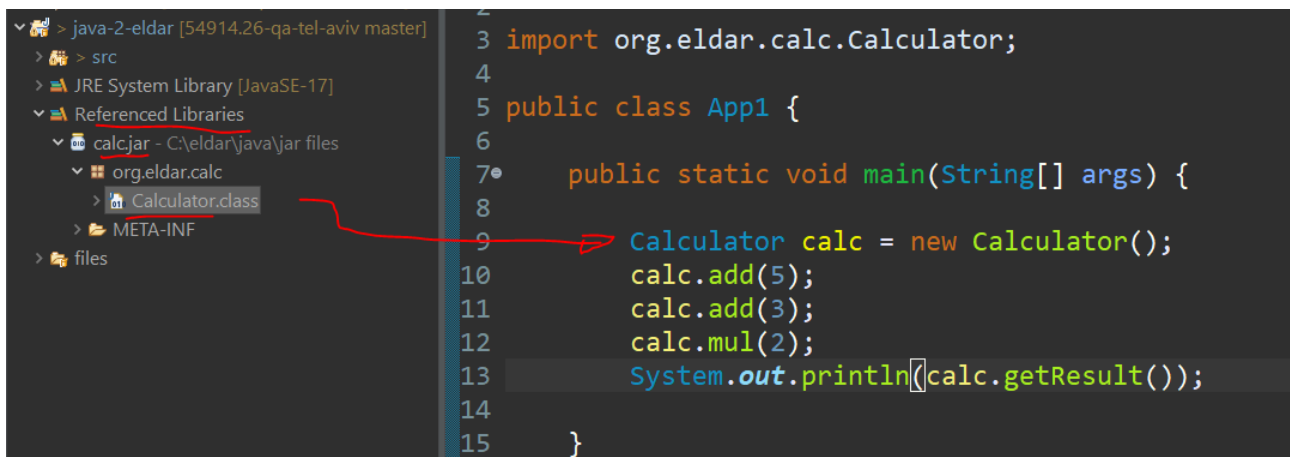
איך ליצור ספריות קוד

יוצרים פרויקט עם packages ועם classes שמכילים מתודות וכדומה.

מייצאים את הפרויקט לספרייה – קובץ דחוס בפורמט ZIP עם סיומת JAR (ראשי תיבות של Java (Archive)

איך להשתמש בספריות בספרייה

יש לייבא את ה jar לפרויקט שבו נרצה להשתמש בספרייה.
עכשיו אפשר לעבוד עם המחלקות שמוגדרות ב JAR



```
1 2
3  import org.eldar.calc.Calculator;
4
5  public class App1 {
6
7      public static void main(String[] args) {
8
9          Calculator calc = new Calculator();
10         calc.add(5);
11         calc.add(3);
12         calc.mul(2);
13         System.out.println(calc.getResult());
14
15     }
```