



Java OOP

10128

Constructor

Pini shlomi

Constructor

- A constructor is a special method that is used to **initialize objects**
- The constructor method must have the **same name** as the class name and without returning type
- The constructor is called **only once** when an object of a class is created.

Types:

- Default constructor
- Constructor with parameters
- Copy constructor

Default constructor

- All classes have constructors **by default.**
- If you do not create a class constructor yourself, Java creates one for you.
- If you create a constructor, the default constructor **not exist anymore,** if you need it, you need to create it.

```
public class Box {  
    private double width, height, depth;  
    // default constructor unshown in class  
    // public Box () {  
    // }  
}
```

Create Object by using a default constructor

```
Box b1 = new Box ();
```

Constructor with parameters

- Constructors can also take parameters, which is used to initialize attributes.
- Overloading** Constructors , We can create several constructors with difference parameters

```
public class Box {  
    private double width, height, depth;  
    public Box() {  
    }  
    public Box(double width, double height, double depth) {  
        this.width = width;  
        this.height = height;  
        this.depth = depth;  
    }  
}  
  
Box b1 = new Box(); // using constructor without parameters  
Box b2 = new Box(2.5, 3.25, 4.5); // using constructor with parameters
```

Overloading Constructors Using `this()`

```
public class Box {  
    private double width, height, depth;  
    public Box() {  
        this(0, 0, 0); // we can ignore this line, why?  
    }  
    public Box(double width, double height, double depth) {  
        this.width = width;  
        this.height = height;  
        this.depth = depth;  
    }  
    public Box(double width) {  
        this(width, width, width);  
    }  
}  
  
Box b1 = new Box(); // using constructor without parameters  
Box b2 = new Box(2.5, 3.25, 4.5); // using constructor with all parameters  
Box b3 = new Box(2.5); // using constructor with one parameter
```

Validate Data

- Constructor set initial value to attribute.
- Use set method to set the value.
- Be aware the set method validate the value.

```
public class Box {  
    private double width, height, depth;  
    public Box() {  
        this(0, 0, 0); // again, we can ignore this line, why?  
    }  
    public Box(double width, double height, double depth) {  
        setWidth(width);  
        setHeight(height);  
        setDepth(depth);  
    }  
    public Box(double width) {  
        this(width, width, width);  
    }  
}
```

תרגיל 1:

- מהו בנאי (Constructor) ומה מטרתו?
- מהם סוגי הבנאים הקיימים ב-Java?
- מה יקרה אם לא נגדיר בנאי למחלקה?

```
public class Exe {  
    public static void main(String[] args) {  
        Car c = new Car();  
    }  
}  
  
public class Car {  
    private String model;  
    private int year;  
  
    public Car(String model, int year) {  
        this.model = model;  
        this.year = year;  
    }  
}
```

תרגיל 2:

- מה יקרה אם ננסה
ליצור מופע של
המחלקה Car ללא
פרמטרים?
כיצד ניתן לפתור
זאת?

Object Reference

```
public static void main(String[] args) {  
    Car[] cars = new Car[3];  
    System.out.println(Arrays.toString(cars));  
    cars[0] = new Car("Kia", 2019);  
    cars[1] = new Car("Subaru", 2025);  
    cars[2] = new Car("Opel", 2022);  
    System.out.println("After init");  
    System.out.println(Arrays.toString(cars));  
    Car firstCar = cars[0];  
    firstCar.setYear(2020);  
    System.out.println("After changed first car");  
    System.out.println(Arrays.toString(cars));  
}
```

[null, null, null]

After init

[Kia, 2019, Subaru, 2025, Opel, 2022]

After changed first car

[Kia, 2020, Subaru, 2025, Opel, 2022]

```
class Car {  
    private String model;  
    private int year;  
  
    public Car(String model, int year) {  
        this.model = model;  
        this.year = year;  
    }  
  
    public void setYear(int year) {  
        this.year = year;  
    }  
  
    @Override  
    public String toString() {  
        return model + ", " + year;  
    }  
}
```

Copy Constructor

- Constructor that creates an object using another object of the **same class**
- We passing another object as a parameter

```
public class Box {  
    private double width, height, depth;  
  
    ... other constructors  
  
    public Box(Box other) {  
        why do not need to use setter's functions?  
        this.width = other.width;  
        this.height = other.height;  
        this.depth = other.depth;  
    }  
}
```

תרגיל 3:

כתבו מחלקה חדשה Adult, המייצגת אדם בוגר.

1. נתוני המחלקה הינם שם, גובה ומקצוע.
2. כתבו בנאי המקבל את שלושת הנתונים.
3. כתבו בנאי המקבל רק שם וגובה.
4. כתבו בנאי העתקה.
5. ממשו שיטות get ו-set והקפידו על בדיקת תקינות הנתונים.
6. ממשו את השיטה toString, במידה ואין מקצוע אין להדפיס תכונה זאת.

תרגיל 3 - המשך:

ב-main בצעו את הפעולות הבאות תוך הדפסת הודעות מתאימות למסך.

a. צרו אדם p1 תוך שימוש בבנאי של שלושת הנתונים.

b. צרו אדם p2 תוך שימוש בבנאי שמקבל רק שם וגובה.

c. צרו אדם p3 על בסיס p1.

d. הדפיסו את נתוני שלושת האובייקטים.

e. עדכנו את המקצוע של p1 ובדקו כיצד הושפע p3.

תרגיל 4: מה יקרה בהרצת הקוד הבא?

```
public class Test {  
    private int x;  
  
    public Test() {  
        this(5);  
    }  
  
    public Test(int x) {  
        x = x;  
    }  
  
    public void printX() {  
        System.out.println(x);  
    }  
}
```

```
public static void main(String[] args) {  
    Test obj = new Test();  
    obj.printX();  
}
```

תרגיל 5: מה יקרה בהרצת הקוד הבא?

```
class Employee {  
    private String name;  
    private double salary;  
  
    public Employee() {  
        this.name = "Unknown";  
        this.salary = 5000;  
    }  
  
    public Employee(String name, double salary) {  
        this.name = name;  
        this.salary = salary;  
    }  
  
    public Employee(Employee other) {  
        this.name = other.name;  
        this.salary = other.salary;  
    }  
  
    @Override  
    public String toString() {  
        return "Employee: " + name + ", Salary: " + salary;  
    }  
}
```

```
public static void main(String[] args) {  
    Employee e1 = new Employee();  
    Employee e2 = new Employee("David", 7000);  
    Employee e3 = new Employee(e2);  
  
    System.out.println(e1);  
    System.out.println(e2);  
    System.out.println(e3);  
}
```

תרגיל 6: מה יקרה בהרצת הקוד הבא?

```
public class Circle {  
    private double radius;  
  
    public Circle() {  
        this.radius = 1.0;  
    }  
  
    public void Circle(double r) {  
        this.radius = r;  
    }  
  
    public double getRadius() {  
        return radius;  
    }  
}
```

```
public static void main(String[] args) {  
    Circle c1 = new Circle(5.0);  
    System.out.println(c1.getRadius());  
}
```

תרגיל 7: מה יקרה בהרצת הקוד הבא?

```
public class A {  
    private int x;  
  
    public A() {  
        this(10);  
    }  
  
    public A(int x) {  
        this.x = x;  
        System.out.println("Constructor A: " + x);  
    }  
}
```

```
public static void main(String[] args) {  
    A a1 = new A();  
    A a2 = new A(20);  
}
```


תרגיל 8 : מה יקרה בהרצת הקוד הבא?

```
public class B {  
    private int y;  
  
    public B(int y) {  
        this.y = y;  
    }  
  
    public void setY(int y) {  
        this.y = y;  
    }  
  
    public void printY() {  
        System.out.println(y);  
    }  
}
```

```
public static void main(String[] args) {  
    B b1 = new B(5);  
    B b2 = b1;  
    b2.setY(10);  
    b1.printY();  
}
```

תרגיל 9 : מה יקרה בהרצת הקוד הבא?

```
public class C {  
    private int num;  
  
    public C() {  
        this(100);  
    }  
  
    public C(int num) {  
        this.num = num;  
    }  
  
    public int getNum() {  
        return num;  
    }  
}
```

```
public static void main(String[] args) {  
    C[] arr = new C[3];  
    for (int i = 0; i < arr.length; i++) {  
        arr[i] = new C(i * 10);  
    }  
    System.out.println(arr[0].getNum());  
    System.out.println(arr[1].getNum());  
    System.out.println(arr[2].getNum());  
}
```

תרגיל 10 : מה יקרה בהרצת הקוד הבא?

```
public class D {  
    private String name;  
  
    public D(String name) {  
        this.name = name;  
    }  
  
    public void changeName(String newName) {  
        this.name = newName;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```

```
public static void main(String[] args) {  
    D d1 = new D("Alice");  
    D d2 = new D(d1.getName());  
    d1.changeName("Bob");  
    System.out.println(d1.getName());  
    System.out.println(d2.getName());  
}
```

תרגיל 11 : מה יקרה בהרצת הקוד הבא?

```
public class E {  
    private int value;  
  
    public E(int value) {  
        this.value = value;  
    }  
  
    public void updateValue(E obj) {  
        obj.value += 5;  
    }  
  
    public int getValue() {  
        return value;  
    }  
}
```

```
public static void main(String[] args) {  
    E e1 = new E(10);  
    E e2 = new E(20);  
    e1.updateValue(e2);  
    System.out.println(e1.getValue());  
    System.out.println(e2.getValue());  
}
```

תרגיל 12 : מה יקרה בהרצת הקוד הבא?

```
public class F {  
    private int[] data;  
  
    public F() {  
        this.data = new int[]{1, 2, 3};  
    }  
  
    public int[] getData() {  
        return data;  
    }  
  
    public void modifyArray(int[] arr) {  
        arr[0] = 99;  
    }  
  
    public void printArray() {  
        System.out.println(Arrays.toString(data));  
    }  
}
```

```
public static void main(String[] args) {  
    F Arrays.toString = new F();  
    f.modifyArray(f.getData());  
    f.printArray();  
}
```

תרגיל 13 :

- כתבו מחלקה חדשה , Rectangle תכונותיה הן אורך ורוחב.
1. כתבו בנאי שלא מקבל פרמטרים ומאתחל את האורך והרוחב ל-10.
 2. כתוב בנאי נוסף שמקבל אורך ורוחב.
 3. כתבו שיטות get לשתי תכונותיו
 4. כתבו שיטה getPerimeter לחישוב היקף המלבן.
 5. כתבו שיטה getArea לחישוב שטח המלבן.
 6. ממשו את השיטה toString שתדפיס את צורת המלבן עם כוכביות *.
 7. כתבו שיטה showByChar שמקבלת תו ומדפיסה את צורת המלבן עם התו שהתקבל.

תרגיל 13 המשך :

ב-main בצעו את הפעולות הבאות תוך הדפסת הודעות מתאימות למסך.

a. יצרו מלבן עם ערכי ברירת המחדל והציגו אותו.

b. יצרו מלבן עם הבנאי שמקבל אורך ורוחב והדפיסו אותו.

c. הציגו את היקף המלבן.

d. הציגו את שטח המלבן.

e. הציגו את המלבן עם התו \$.

תרגיל 14:

Attributes – מאפיינים

- -name: String
- -weight: float
- -color: String, Default value Black
- -breed: String, Default value Terrier

Constructors – בנאים

- + Dog()
- + Dog(String name, float weight)
- + Dog(String name, float weight, String color)
- + Dog(String name, float weight, String color, String breed)

Methods – מתודות

- + Getters & Setters
- + toString(): String

1. כתבו מחלקה חדשה **Dog**
2. צרו מספר אובייקטים תוך שימוש בכל הבנאים והדפיסו אותם, שימו לב לאופן כתיבת הבנאים, יש להשתמש ב-**this**.

תרגיל 15:

מאפיינים – Attributes

- -name: String
- -color: String, Default value : Silver
- -speed: int, Default value : 20
-

בנאים – Constructors

- + Fish()
- + Fish(String name, String color)
- + Fish(String name, String color, int speed)

מתודות – Methods

- + Getters & Setters
- + toString(): String

1. כתבו מחלקה חדשה **Fish**
2. צרו מספר אובייקטים תוך שימוש בכל הבנאים והדפיו אותם, שימו לב לאופן כתיבת הבנאים, יש להשתמש ב-this.