



Java OOP

10128

Class and Object

Pini shlomi

Class

- Java is an OOP language
- Object has attributes and methods
- Class is a “blueprint” for creating object
- Attribute is a variable that keep the data of a specific object
- Method is a collection of commands for executing action with a common idea
- Object is an instantiation of class
- Create Class by using the **class** keyword

```
public class Person {  
}
```

- Create object/ instance from class

```
Person p1 = new Person();
```

Class

- Class attributes are the fields like name, age
- Methods are the class functions
- Keep attribute and method names with meaning
- Person class

```
public class Person {  
    public String name = "David";  
    public int age = 25;  
}
```

- Create object/ instance from class

```
Person p1 = new Person();  
System.out.println(p1.name); => David  
System.out.println(p1.age);  => 25
```

Create several objects from class

```
public class CreateSeveralPersons {  
  
    public static void main(String[] args) {  
        Person p1 = new Person();  
        Person p2 = new Person();  
        p2.name = "Mor";  
        p2.age = 23;  
  
        System.out.println(p1.name); => David  
        System.out.println(p1.age);  => 25  
  
        System.out.println(p2.name); => Mor  
        System.out.println(p2.age);  => 23  
    }  
}
```

Access modifier

- Access modifier is used to set the access level for classes, attributes and methods
- Class:
 - **public** - The class is accessible by any other class
 - Default - The class is only accessible by classes in the same package
- Method , Attribute
 - **public** - The code is accessible by any other class
 - **private** - The code is accessible within declared class
 - Default - **protected** The code is only accessible by classes in the same package

Non -Access modifier

- Class:
 - **final** - The class cannot be inherited by other classes
 - **abstract** - The class cannot be used to create
- Method , Attribute
 - **final** - cannot be overridden/modified
 - **static** - belongs to the class, rather than an object
 - **abstract** - used in an abstract class and can only be used on methods
 - **Transient** - methods and attribute are skipped when serializing the object containing them
 - **Synchronized** - method can only be accessed by one thread at a time
 - **volatile** - The value of a attribute is not cached thread-locally, and is always read from the "main memory"

Getters and Setters

- By default, all class attributes will have **private** access modifier
- Create methods to attribute data access
- All class attributes have initial default values
- Get:
 - Function `get< attributeName>` return the attribute value
 - Be aware – **never** return reference of attribute of object type
- Set:
 - Function `set< attributeName>` set value to attribute
 - Check attribute data validation
 - Be aware – **never** set reference to attribute of object type

תרגיל 1 :

כתבו מחלקה חדשה Clock, תכונותיה הן שעות, דקות ושניות.

1. כתבו שיטות get ו-set והקפידו על בדיקת תקינות הנתונים.

2. כתבו את השיטה tick כך שתקדם את השניות

3. כתבו את השיטה show

תרגיל 1 - המשך :

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

a. הגדירו אובייקט c1 מטיפוס המחלקה ושימו בו נתונים באמצעות שיטות set.

b. הציגו את נתוני הזמן שלו

c. קדמו את השעה ב-10 שניות

d. הציגו את נתוני הזמן שלו

e. אפסו את הזמן שלו

f. הציגו את נתוני הזמן שלו

תרגיל 2 :

כתבו מחלקה חדשה , Car , תכונותיה הן מספר הרכב ומהירותו הנוכחית

1. כתבו שיטות get לשתי התכונות.

2. כתבו שיטת set לעדכון מספר הרכב

3. כתבו את השיטות הבאות: (methods)

a. להאיץ – העלאת המהירות ב-1 קמ"ש

b. להאט – הורדת המהירות ב-1 קמ"ש

c. לעצור – עצירת הרכב

d. הצגת נתוני הרכב

תרגיל 2 - המשך :

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

a. הגדירו אובייקט מטיפוס המחלקה, קלטו מהשתמש את מספר הרכב והציגו את נתוניו.

b. שאלו את המשתמש כמה להאיץ את הרכב, האיזו את הרכב והציגו שוב את נתוניו.

c. האטו את מהירות הרכב ב-10 קמ"ש והציגו את שוב את נתוניו.

d. עצרו את הרכב והציגו את נתוניו.

תרגיל 3 :

a. כתבו את המחלקה Person עם השדות הבאים : שם, גובה ומשקל.

b. כתבו שיטות get ו-set.

c. הציגו את נתוני האובייקט.

d. בקובץ התוכנית הוסיפו פונקציה חדשה שמקבלת מערך של Person וממיינת בסדר יורד את המערך לפי המשקל.

e. בקובץ התוכנית הוסיפו פונקציה חדשה שמקבלת מערך של Person וממיינת בסדר יורד את המערך לפי הגובה.

תרגיל 4 :

כתבו מחלקה חדשה Date, תכונותיה הן שנה, חודש ויום מסוג מספר שלם.

1. כתבו שיטות get ו-set, הקפידו על בדיקת תקינות הנתונים ועל סדר השימוש, כלומר בצעו ראשית השמה של שנה, לאחר מכן חודש ורק בסוף יום.

a. עבור שנה שתהיה בטווח של 1800-2500, ברירת מחדל 2023

b. עבור חודש בטווח של 1-12 כולל, ברירת מחדל 1

c. עבור יום בטווח של 1 ועד מספר הימים בחודש כתלות בחודש ובשנה, בדקו אם השנה מעוברת כאשר החודש הוא פברואר. ברירת מחדל 1

d. עבור נתונים שגויים הציבו את ערכי ברירת המחדל.

2. ממשו את השיטה toString להצגת התאריך בפורמט "DD/MM/YYYY".

3. כתבו את השיטה addDay המוסיפה יום אחד לתאריך.

4. כתבו את השיטה subDay המחסירה יום אחד מהתאריך.

תרגיל 4 המשך :

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

a. הגדירו אובייקט d1 מטיפוס המחלקה והציבו בו נתונים באמצעות שיטות set.

b. הציגו את נתוני התאריך שלו.

c. הוסיפו יום לתאריך והציגו אותו.

d. החסירו יום לתאריך והציגו אותו.

e. הגדירו אובייקט d2 מטיפוס המחלקה והציבו בו נתונים באמצעות שיטות set.

f. הציגו את נתוני התאריך שלו.



תרגיל 4 המשך :

דוגמא ל-output

```
create date object...  
show date: 28/02/2020
```

```
Adding one day  
Before : 28/02/2020  
After  : 29/02/2020
```

```
show date: 31/12/2020
```

```
Adding one day  
Before : 31/12/2020  
After  : 01/01/2021
```

```
Subtract one day  
Before : 01/01/2021  
After  : 31/12/2020
```