Giriş

Nesne Yönelimli Programlama

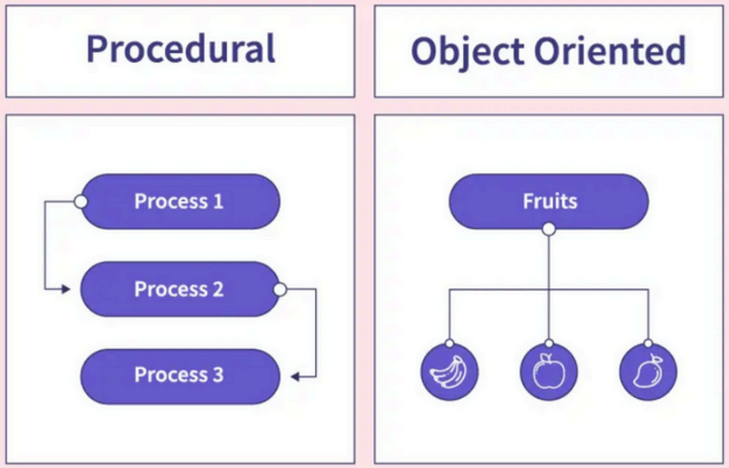
volkanulutas@gmail.com

Procedural vs Object Oriented Programming

Procedural programming is about writing procedures or methods that perform operations on the data, while object-oriented programming is about creating objects that contain both data and methods.

Object-oriented programming has several advantages over procedural programming:

- OOP is faster and easier to execute
- OOP provides a clear structure for the programs
- OOP helps to keep the Java code DRY "Don't Repeat Yourself", and makes the code easier to maintain, modify and debug
- OOP makes it possible to create full reusable applications with less code and shorter development time
- **Tip: The "Don't Repeat Yourself" (DRY) principle is about reducing the repetition of code. You should extract out the codes that are common for the application, and place them at a single place and reuse them instead of repeating it.**

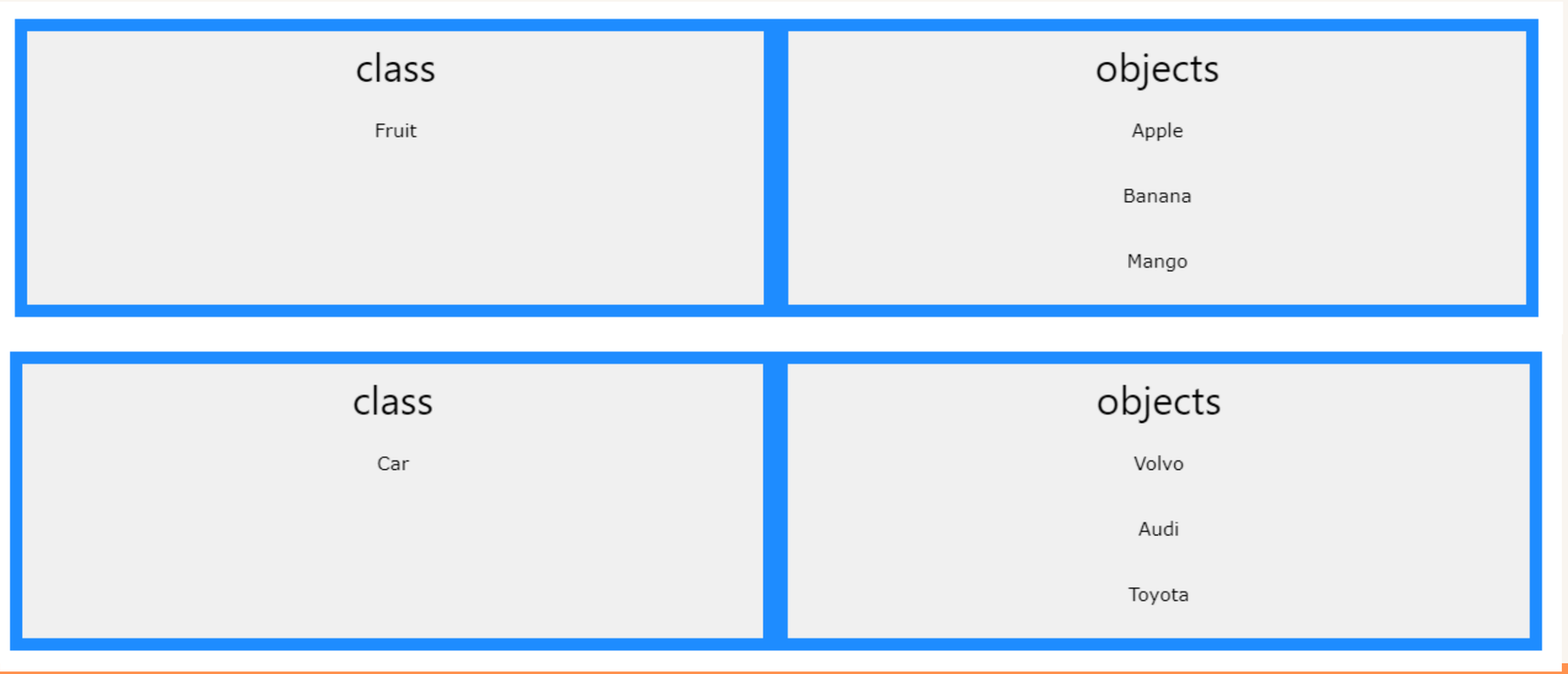


What is “Class” and “object”?

- a **class** is a **template** for objects. Only definition of an object. For example: defining a person, a car, an animal but only definiton for your program.
- an object is an **instance** of a class. **Usable things** of a definition, class.

Template, definiton.

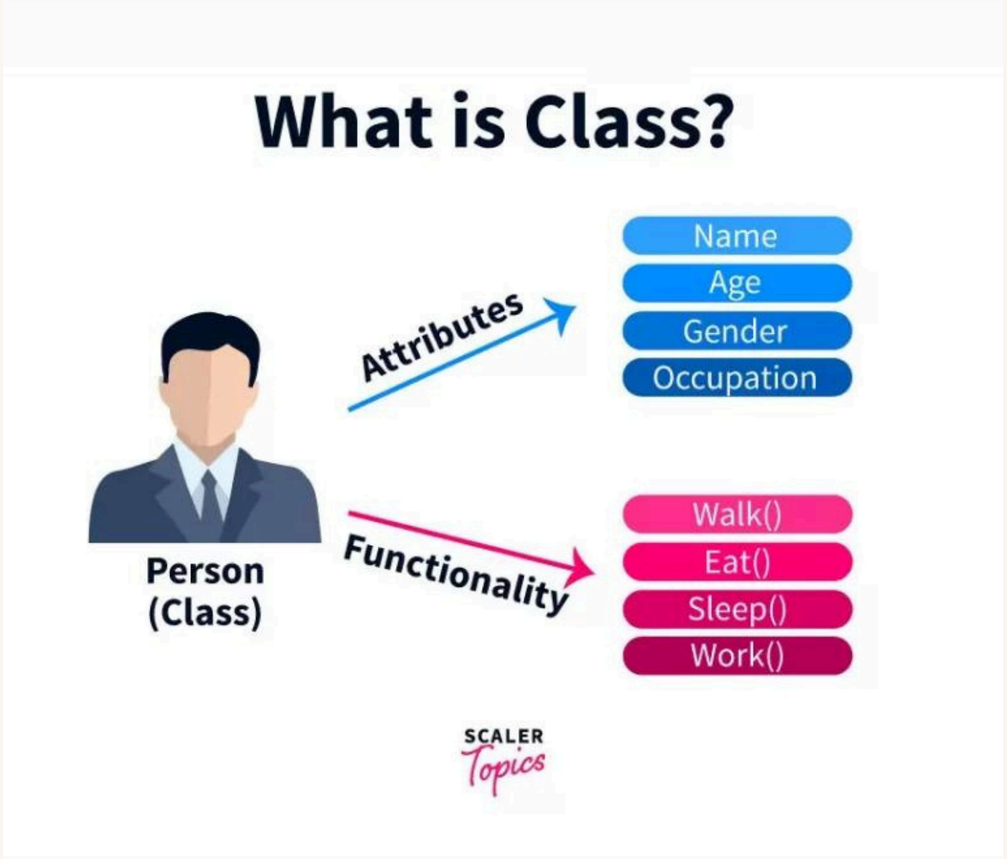
Running instance



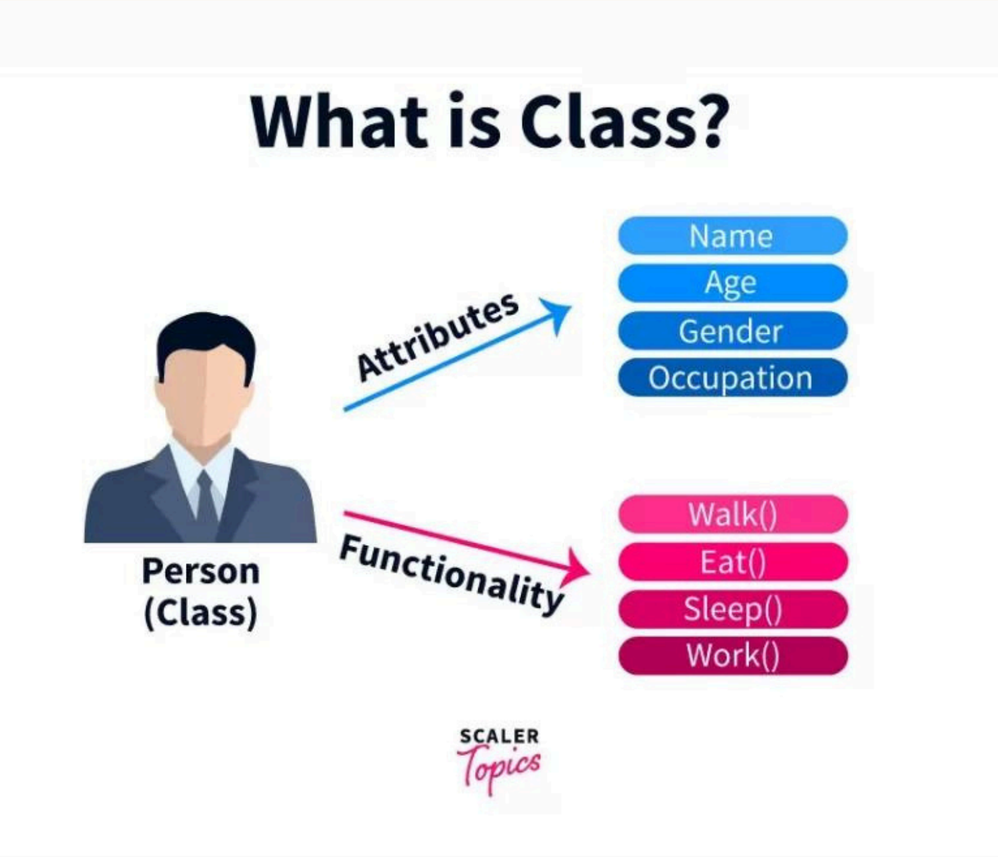
What is “Class” and “object”?

Class: Defining your object, It is a template.

- **Class** store two main aspect. Attributes (Nesnenin özellikleri), and functionality (behaviour, nesnenin davranışı.)



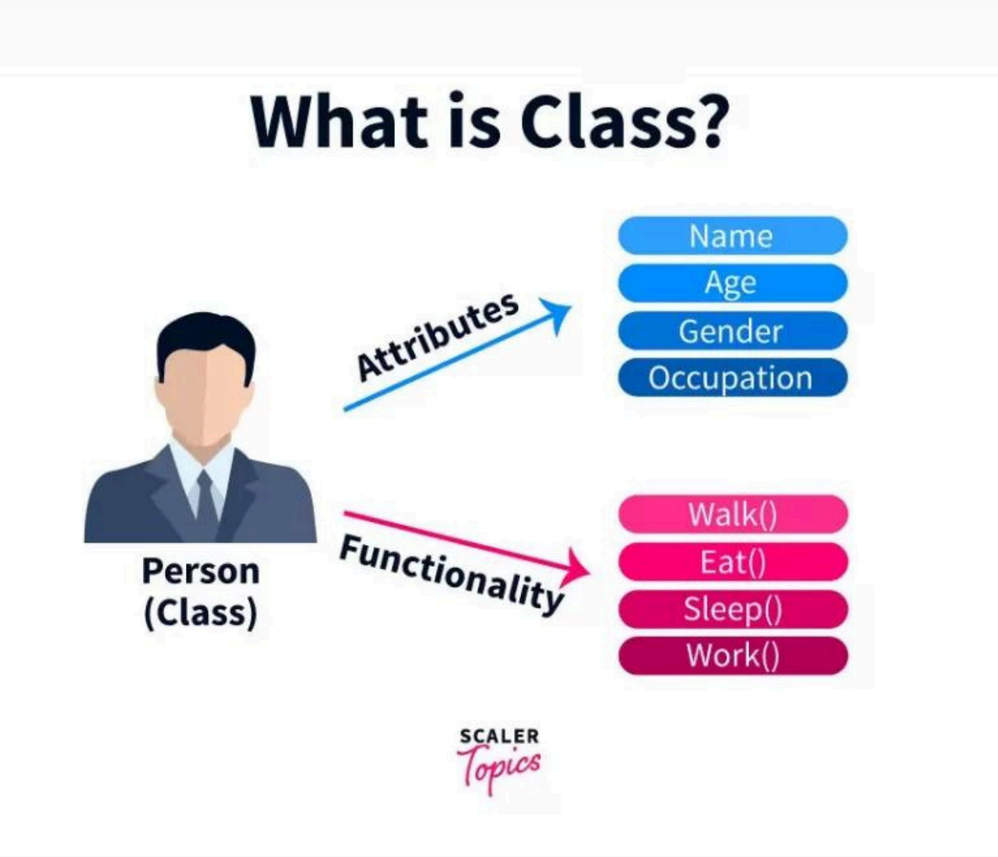
Object: instance of a new thing from your template.



Name: Duru
Age: 23
Gender: Female
Occupation: Student

Functionality, behaviour (Yetenekleri,davranışları)
walk()
eat()
sleep()
work()

Object: instance of a new thing from your template.



Name: Duru
Age: 23
Gender: Female
Occupation: Student

Functionality, behaviour, methods (Yetenekleri,davranışları)
walk()
eat()
sleep()
work()

Student V1

```
public class StudentV1 {
    public int id;

    public String name;

    public String surname;

    public String email;

    public int age;

    public double gpu;

    public Date enrollmentDate;

    public Date graduationDate;

    public String address;

    public String department;
}
```

create a class.
create two objects name “student1” and “student2".
assign attribute and getting attributes
create an method listing attributes called “viewDetails”.

Encapsulation - I

Encapsulation

The meaning of Encapsulation, is to make sure that "sensitive" data is hidden from users. To achieve this, you must:

declare class variables/attributes as private
provide public get and set methods to access and update the value of a private variable

Get and Set

You learned from the previous chapter that private variables can only be accessed within the same class (an outside class has no access to it). However, it is possible to access them if we provide public get and set methods.

The get method returns the variable value, and the set method sets the value.

Syntax for both is that they start with either get or set, followed by the name of the variable, with the first letter in upper case:

Encapsulation - II

```
public class Person {
private String name; // private = restricted access.
// Getter
public String getName() {
return name;
}
// Setter
  public void setName(String newName) {this.name = newName;}
}
```

GET, SET olmazsa access olmaz! Kontrol class'i tanımlayan kişide olur.

Get and Set

You learned from the previous chapter that private variables can only be accessed within the same class (an outside class has no access to it). However, it is possible to access them if we provide public get and set methods.

The get method returns the variable value, and the set method sets the value.

Syntax for both is that they start with either get or set, followed by the name of the variable, with the first letter in upper case:

Encapsulation - III: StudentV2

```
public class StudentV2 {
    private int id;

    private String name;

    private String surname;

    private String email;

    private int age;

    private double gpu;

    private Date enrollmentDate;

    private Date graduationDate;

    private String address;
    private String department;
}
```

create a class. create two objects name “student1” and “student2". assign attribute and getting attributes create an method listing attributes called “viewDetails”.

Access Modifiers

Modifier	Description	Try it
`public`	The code is accessible for all classes	Try it »
`private`	The code is only accessible within the declared class	Try it »
default	The code is only accessible in the same package. This is used when you don't specify a modifier. You will learn more about packages in the Packages chapter	Try it »
`protected`	The code is accessible in the same package and **subclasses**. You will learn more about subclasses and superclasses in the Inheritance chapter	Try it »

Giriş

Nesne Yönelimli Programlama

volkanulutas@gmail.com