

OOP in Dart

Contents

1. Functions in Dart
2. OOP in Dart

Functions in Dart (1)

- They are created when some **statements** are **repeatedly** occurring in the program. The function helps **reusability of the code** in the program.

- Syntax

```
returnType functionName ([param1, param2, ...]) {  
    // function body  
}
```

- Example

```
// this function add two numbers  
int add(int a, int b) {  
    int sum = a + b;  
    return sum;  
}  
  
void main() {  
    int num1 = 10;  
    int num2 = 20;  
  
    int total = add(num1, num2);  
    print("The sum is $total.");  
}
```

Functions in Dart (2)

- Providing **default value** on parameter

```
1 void add (int num1, int num2, [int num3=0]) {  
2     ⚡ int sum;  
3     sum = num1 + num2 + num3;  
4  
5     print("The sum is $sum");  
6 }  
7  
Run | Debug  
8 void main(){  
9     add(10, 20);  
10    add(10, 20, 30);  
11 }
```

Functions in Dart (3)

- **Named parameters** allow you to **specify the names of the arguments** when calling a function. This makes the **code more readable** and helps **avoid errors** that can arise from **passing arguments in the wrong order**.
- To define a function with named parameters, enclose the parameters in curly braces {}.

```
void printUserInfo({String? name, int? age}) {  
  print('Name: $name');  
  print('Age: $age');  
}
```

```
void printUserInfo(String name, int age) {  
  print('Name: $name');  
  print('Age: $age');  
}
```

- Calling a Function with Named Parameters
- Calling Function without Named Parameters

```
void main() {  
  printUserInfo(name: 'John', age: 25);  
  printUserInfo(age: 25, name: 'John');  
}
```

VS

```
void main() {  
  printUserInfo('John', 25);  
  printUserInfo(25, 'John');  
}
```

Functions in Dart (4)

- Making Named Parameters **Required**

```
void printUserInfo({required String name, required int age}) {  
  print('Name: $name');  
  print('Age: $age');  
}
```

- **Nullable** for Named Parameters

```
void printUserInfo({String? name, int? age}) {  
  print('Name: $name');  
  print('Age: $age');  
}
```

Functions in Dart (5)

- Here's a complete example demonstrating all these concepts.

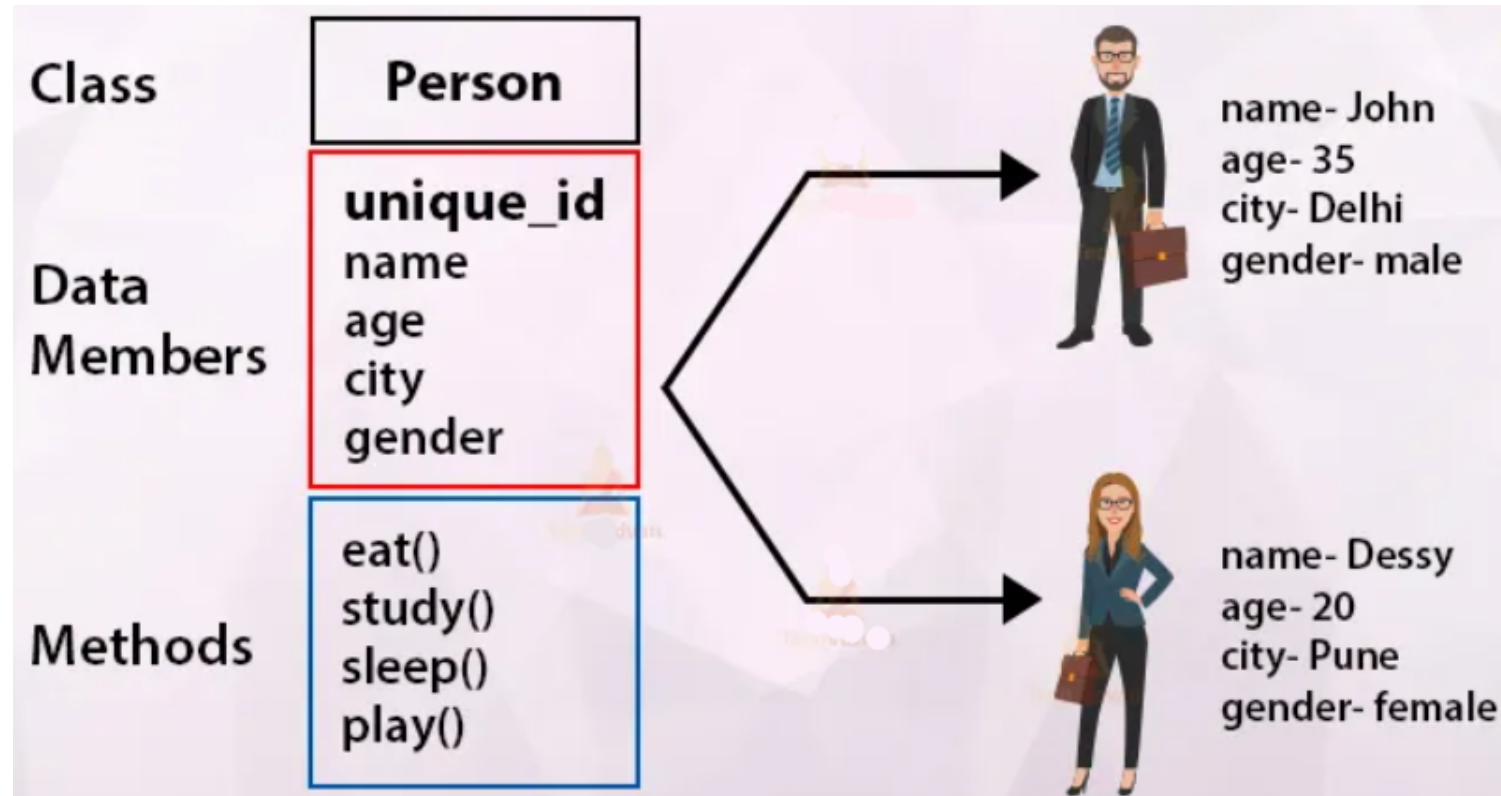
```
void printUserInfo({String name = 'Unknown', int age = 0, String? city}) {  
  print('Name: $name');  
  print('Age: $age');  
  
  if (city != null) {  
    print('City: $city');  
  }  
}  
  
void main() {  
  // Using default values  
  printUserInfo();  
  
  // Overriding default values  
  printUserInfo(name: 'Alice', age: 25, city: 'New York');  
  
  // Omitting optional named parameter  
  printUserInfo(name: 'Bob', age: 30);  
}
```

OOP in Dart (1)

- **Object-oriented programming (OOP)** is a programming method that **uses objects** to design and program
- Features of OOP
 1. Class, Object
 2. Constructor
 3. Inheritance, Encapsulation, Abstraction

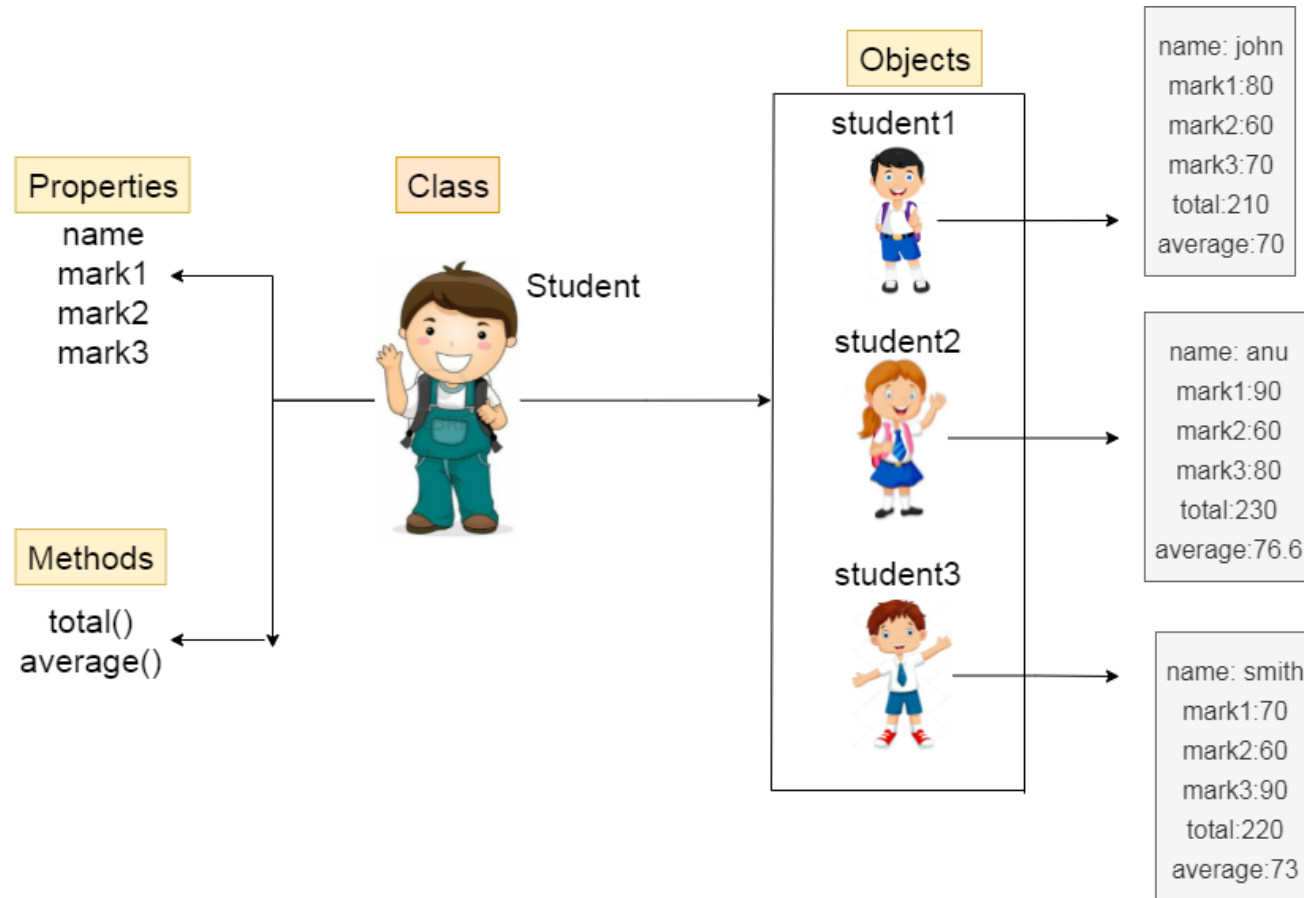
OOP in Dart (2)

- Example



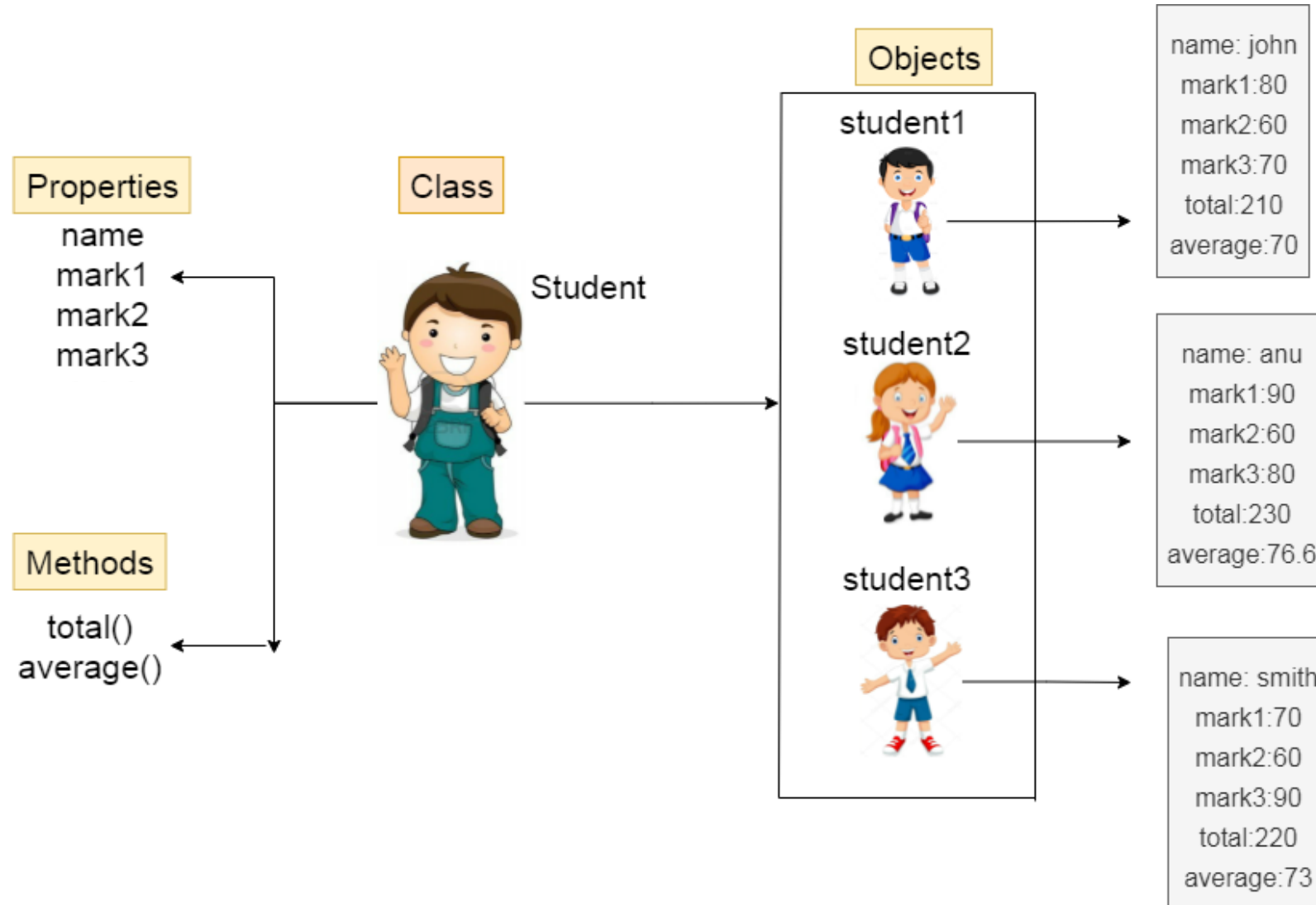
Class in Dart (1)

- A class is a **blueprint** or a template **used to create objects**. It defines the attributes (**properties**) and **methods** that the objects created from the class will have.
- In simpler terms, **a class is a way to define the structure and behavior of an object.**



Class in Dart (2) - How to Identity a Class

- **Class name:** Main noun
- **Properties:** Nouns as modifiers of main noun
- **Methods:** Verbs related to main noun



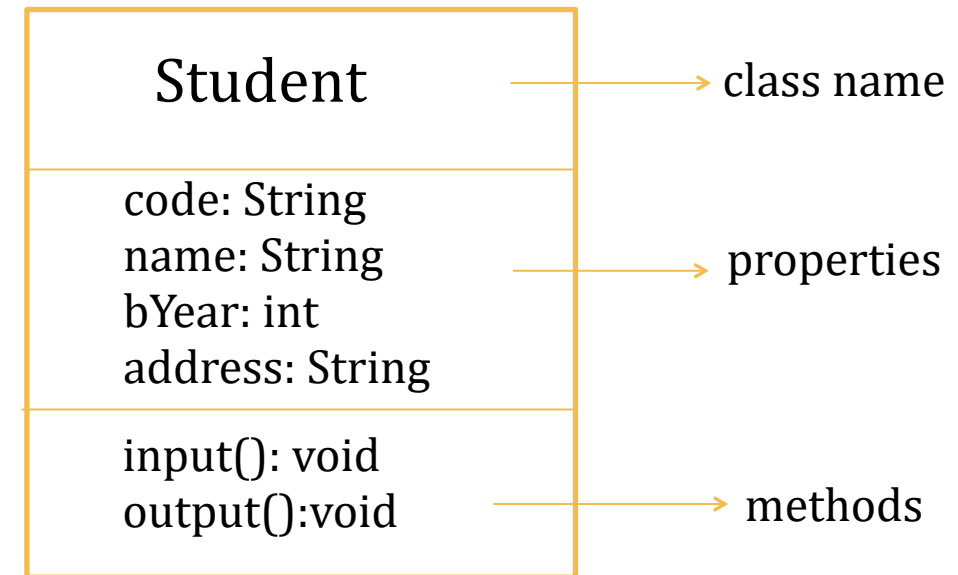
Class in Dart (3) - Hints for class design

- A UML class diagram is used to represent the Student class

Main noun: Student

Auxiliary nouns: code , name, bYear, address

verbs: input(), output()



Class in Dart (4)

- You can declare a class in dart using the **class** keyword followed by class name and braces {}.
- Syntax

```
class ClassName {  
    // properties  
    // methods/functions  
}
```

Example

```
class Student {  
    String code;  
    String name;  
    int birthYear;  
    String? address;  
  
    void output() {  
        print("Code: $code");  
        print("Name: $name");  
        print("Birth Year: $birthYear");  
        print("Address: $address");  
    }  
}
```

Properties

Methods

Class in Dart (5)

- Challenge:
 - Create a class Book with three properties: name, author, and price.
 - Also, create a method called display, which prints out the values of the three properties.
 - Where:
 - Name, author: String
 - Price: number

Object in Dart (1)

- An object is an **instance** of a class.
- You can create **multiple objects** of the **same class**
- Syntax:

```
// Create an object  
ClassName objectName = ClassName();  
var objectName = ClassName();
```

```
// Access property  
objectName.propertyName;
```

```
// Access method  
objectName.methodName([argument]);
```

```
class Car {  
  String? name;  
  String? color;  
  int? numberOfSeats;  
  
  void start() {  
    print("$name Car Started.");  
  }  
}  
  
void main(){  
  // Here car is object of class Car.  
  Car car = Car();  
  car.name = "BMW";  
  car.color = "Red";  
  car.numberOfSeats = 4;  
  car.start();  
  
  // Here car2 is another object of class Car.  
  Car car2 = Car();  
  car2.name = "Audi";  
  car2.color = "Black";  
  car2.numberOfSeats = 4;  
  car2.start();  
}
```

Object in Dart (2)

- Challenge:
 - Create a class **Camera** with properties: **name**, **color**, **megapixel**. Create a method called **display** which prints out the values of the three properties.
 - Create **two objects** of the class Camera and call the method display.
 - Where:
 - Name, color: String
 - Megapixel: number

*Keeping up those **inspiration** and the **enthusiasm** in the **learning path**.
Let confidence to bring it into **your career path** for getting gain the **success**
as your expectation.*

Thank you

Contact

- Name: R2S Academy
- Email: daotao@r2s.edu.vn
- Phone/Zalo: 0919 365 363
- FB: <https://www.facebook.com/r2s.tuyendung>
- Website: www.r2s.edu.vn

Questions and Answers