

Bài 3

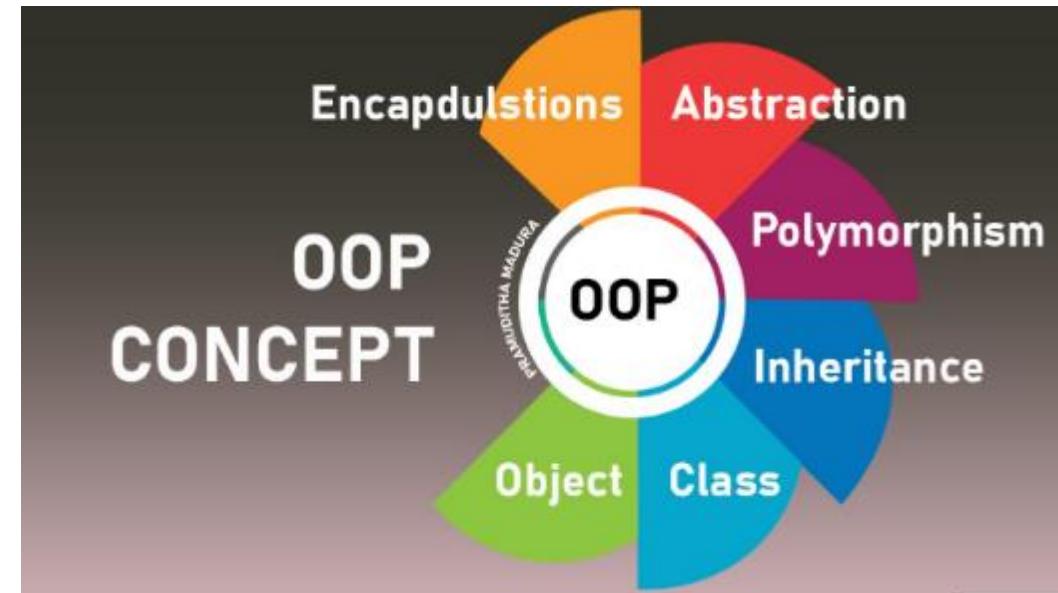
Classes và Objects

NỘI DUNG

- Lập trình hướng đối tượng (OOP)
- Đối tượng (object)?
- Lớp (Classes)
- Biến (Variables)
- Phương thức (Methods)
- Phương thức khởi tạo (Constructors)
- Tạo đối tượng (Creating Objects)
- Con trỏ **this**

Lập trình HĐT - OOP?

- Lập trình hướng đối tượng (Object-Oriented Programming - OOP) là một phương pháp lập trình dựa trên khái niệm “**đối tượng**” (object), trong đó mỗi đối tượng đại diện cho một thực thể trong thế giới thực bao gồm các thuộc tính (properties) và phương thức (methods).
- Các nguyên lý chính của LTHĐT bao gồm:



Đối tượng - Object?

Object là một thực thể cụ thể được tạo ra từ một lớp (class) gồm các trạng thái, hành vi.

Trạng thái (State):

Là tập hợp các thuộc tính (attributes) hay dữ liệu của object, thường được biểu diễn dưới dạng các biến. Ví dụ, một object "**Xe hơi**" có các thuộc tính như "năm sx", "hãng xe", "màu sắc", "số chỗ ngồi",...

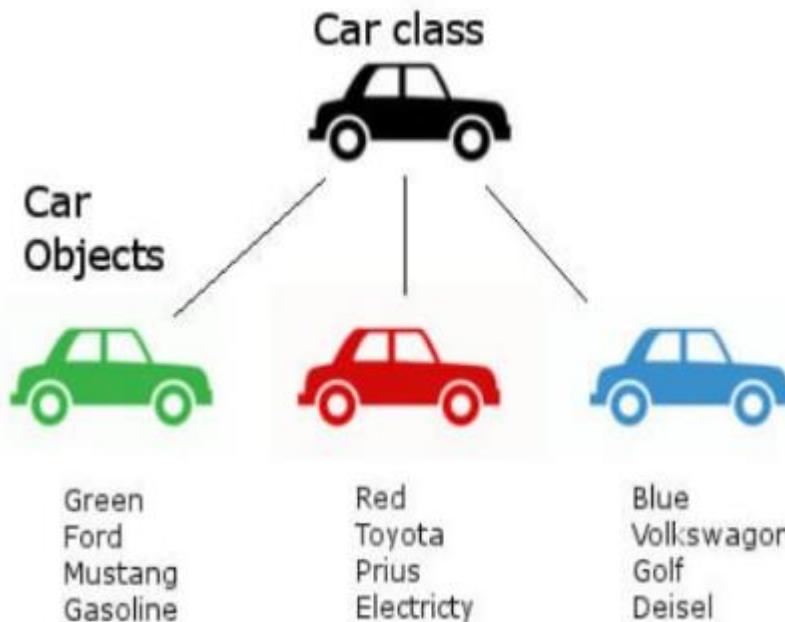
Hành vi (Behavior):

Là các phương thức (methods) mà object có thể thực hiện. Ví dụ, với object "**Xe hơi**", gồm các phương thức: "chạy", "dừng", "bật đèn", "mở cửa",...



Lớp - Class?

Lớp (class) là một bản thiết kế hoặc một khuôn mẫu để tạo ra các đối tượng. Nó định nghĩa các thuộc tính (dữ liệu) và phương thức (hành vi) chung mà đối tượng có.



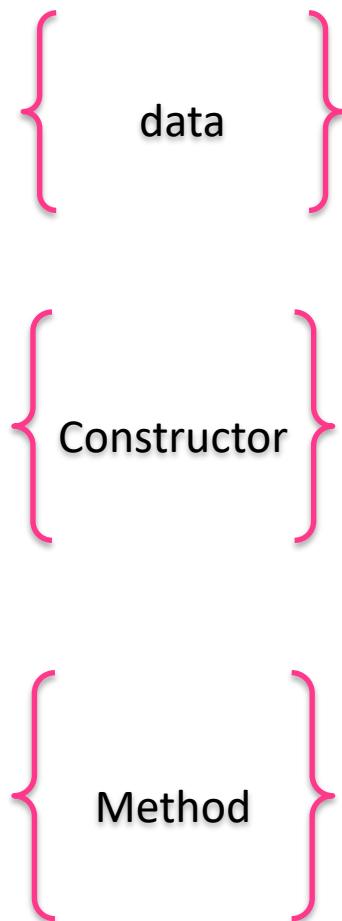
```
public class Car {  
  
    private String color;  
    private int size;  
    private int capacity;  
    private double weight;  
  
    public void start() {  
        // Start the engine  
    }  
  
    public void stop() {  
        // Stop the engine  
    }  
  
    public void accelerate() {  
        // accelerate  
    }  
}
```



Khai báo lớp

Syntax	Example
<pre>class ClassName { // fields // constructors // methods }</pre>	<pre>class Student { //fields String name; int age; // constructors // methods void output() { System.out.println(name); } }</pre>

Ví dụ lớp



```
1 package helloworld;
2 public class Student {
3     //Khai báo biến - các fields
4     String name;
5     int age;
6
7     //Phương thức khởi tạo
8     public Student(String name, int age) {
9         this.name = name;
10        this.age = age;
11    }
12
13     //Phương thức
14     public void Print(){
15         System.out.println("Name: " + name + " Age: " + age);
16     }
17 }
```

Biến - variables

- **Member variables (fields):** biến của lớp
- **Local variables:** biến trong phương thức hoặc biến trong khối lệnh
- **Parameters:** Tham số khai báo trong phương thức

Biến - variables

```
2  public class Student {  
3      //Khai báo biến - các fields  
4      String name;           → { class  
5      int age;               } variables  
6  
7      //Phương thức khởi tạo  
8      public Student(String name, int age) {  
9          this.name = name;  
10         this.age = age;  
11     }  
12  
13     //Phương thức  
14     public void Print(int year){  
15         int DoB;  
16         DoB = year - age;  
17         System.out.println("Name: " + name + " Age: " + age);  
18         System.out.println("She/He was born in " + DoB);  
19     }  
20 }
```

local variable

paramater

Qui ước đặt tên

- **Class name:** the first letter should be capitalized
Student, Person, FileReader, FileWriter, etc.
- **Variables name:** should be a **noun** in lowercase
name, firstName, lastName, yearOfBirth, etc.
- **Method name:** the first word should be a **verb** in lowercase
run, runFast, getBackground, getFinalData,
isEmpty, compareTo, etc.

Phương thức - Methods

- Phương thức (method) là một hàm được định nghĩa bên trong một class để xử lý hoặc thực hiện nhiệm vụ liên quan đến dữ liệu trong lớp

[quyền truy xuất] [kiểu DL] TênPT ([các tham số])

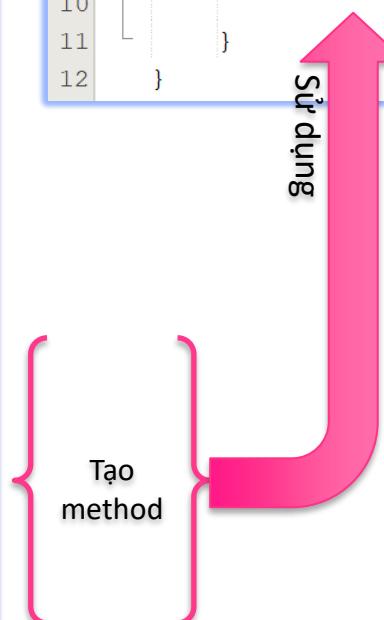
```
{  
    //các lệnh  
    [return biểu thức];  
}
```

```
//Phương thức  
public void Print(int year) {  
    int DoB;  
    DoB = year - age;  
    System.out.println("Name: " + name + " Age: " + age);  
    System.out.println("She/He was born in " + DoB);  
}
```

Sử dụng phương thức

```
2 public class Student {  
3     //Khai báo biến - các fields  
4     String name;  
5     int age;  
6  
7     //Phương thức khởi tạo  
8     public Student(String name, int age) {  
9         this.name = name;  
10        this.age = age;  
11    }  
12  
13    //Phương thức  
14    public void Print(int year){  
15        int DoB;  
16        DoB = year - age;  
17        System.out.println("Name: " + name + " Age: " + age);  
18        System.out.println("She/He was born in " + DoB);  
19    }  
20}
```

```
3 public class HelloWorld {  
4  
5     public static void main(String[] args) {  
6  
7         Student st = new Student("Messi", 37);  
8  
9         st.Print(2024);  
10    }  
11}  
12
```



Tạo 1 object
thuộc lớp
Student

Phương thức khởi tạo Constructor

- Constructor là phương thức đặc biệt được dùng để khởi tạo đối tượng khi nó được tạo ra từ một class. Constructor có các đặc điểm sau:
 - Tên của constructor trùng với tên của class.
 - Constructor không có kiểu trả về
 - Constructor được dùng để gán giá trị ban đầu cho các thuộc tính của đối tượng, cấp phát vùng nhớ
- Trong Java, có hai loại constructor:
 - Default constructor (hàm tạo mặc định) không có tham số.
 - Parameterized constructor (hàm tạo có tham số)

Phương thức khởi tạo

Constructor

```
2  public class Student {  
3      //Khai báo biến - các fields  
4      String name;  
5      int age;  
6  
7      //Phương thức khởi tạo không tham số  
8      public Student() {  
9          name = "Ronaldo";  
10         age = 40;  
11     }  
12  
13      //Phương thức khởi tạo có tham số  
14      public Student(String name, int age) {  
15          this.name = name;  
16          this.age = age;  
17      }  
18  
19      //Phương thức  
20      public void Print(int year){  
21          int DoB;  
22          DoB = year - age;  
23          System.out.println("Name: " + name + " Age: " + age);  
24          System.out.println("She/He was born in " + DoB);  
25      }
```

{ Phương thức khởi tạo
không tham số }

{ Phương thức khởi tạo
có tham số }

Tạo objects

```

1 package helloworld;
2 public class Student {
3     //Khai báo biến - các fields
4     String name;
5     int age;
6
7     //Phương thức khởi tạo không tham số
8     public Student() {
9         name = "Ronaldo";
10        age = 40;
11    }
12
13    //Phương thức khởi tạo có tham số
14    public Student(String name, int age) {
15        this.name = name;
16        this.age = age;
17    }
18
19    //Phương thức
20    public void Print(int year){
21        int DoB;
22        DoB = year - age;
23        System.out.println("Name: " + name + ", Age: " + age);
24        System.out.println("She/He was born in " + DoB);
25    }

```

```

3     public class HelloWorld {
4
5         public static void main(String[] args) {
6
7             Student st1 = new Student();
8
9             Student st2 = new Student("Messi", 37);
10
11            st1.Print(2024);
12
13            st2.Print(2024);
14
15        }
16
17    }

```

run:
 Name: Ronaldo, Age: 40
 She/He was born in 1984
 Name: Messi, Age: 37
 She/He was born in 1987
 BUILD SUCCESSFUL (total time: 0 seconds)

Con trả this

```
1 package helloworld;
2 public class Student {
3     //Khai báo biến - các fields
4     String name;
5     int age;
6
7     //Phương thức khởi tạo có tham số
8     public Student(String name, int age) {
9         this.name = name;
10        this.age = age;
11    }
12
13     //Phương thức
14     public void Print(int year){
15         int DoB;
16         DoB = year - age;
17         System.out.println("Name: " + name + ", Age: " + age);
18         System.out.println("She/He was born in " + DoB);
19     }
20     public void Display(){
21         this.Print(2000);
22     }
23 }
```

{ Con trả this để phân biệt tên biến instance và tham số của phương thức khi trùng tên }

{ Con trả this để gọi phương thức trong lớp }

HỎI ĐÁP

