

Lập trình hướng đối tượng (OOP)





Function: hàm

- Hàm là chương trình thực hiện một tác vụ cụ thể.

```
function function_name($vars) {  
    //đoạn code  
    return $val;  
}
```

- Nếu hàm không có trả về giá trị thì không có dòng **return**.



Function: hàm

- Viết hàm kiểm tra 1 số có phải là số chẵn không
- Viết hàm tính diện tích hình chữ nhật
- Viết các hàm tính tổng, hiệu, tích, thương 2 số



Giới thiệu

- Lập trình hướng đối tượng (Object-Oriented Programming, viết tắt là OOP) là một phương pháp mới trên bước đường tiến hóa của việc lập trình máy tính, nhằm làm cho chương trình trở nên linh hoạt, tin cậy và dễ phát triển.
- Cho phép nhóm các action tượng tự nhau vào các class giúp mã lệnh không lặp lại và dễ dàng để bảo trì.

OOP

The logo features the text "VinaENTER" in a stylized font, with "Vina" in orange and "ENTER" in white. The text is positioned inside a white arrow pointing to the right, which is set against a blue background. The entire graphic is enclosed in a silver-colored frame that resembles a hanging sign.

VinaENTER

OOP:

- Tính đóng gói: class, object
- Tính trừu tượng: interface và abstract
- Tính thừa kế: extend
- Tính đa hình: overload và override

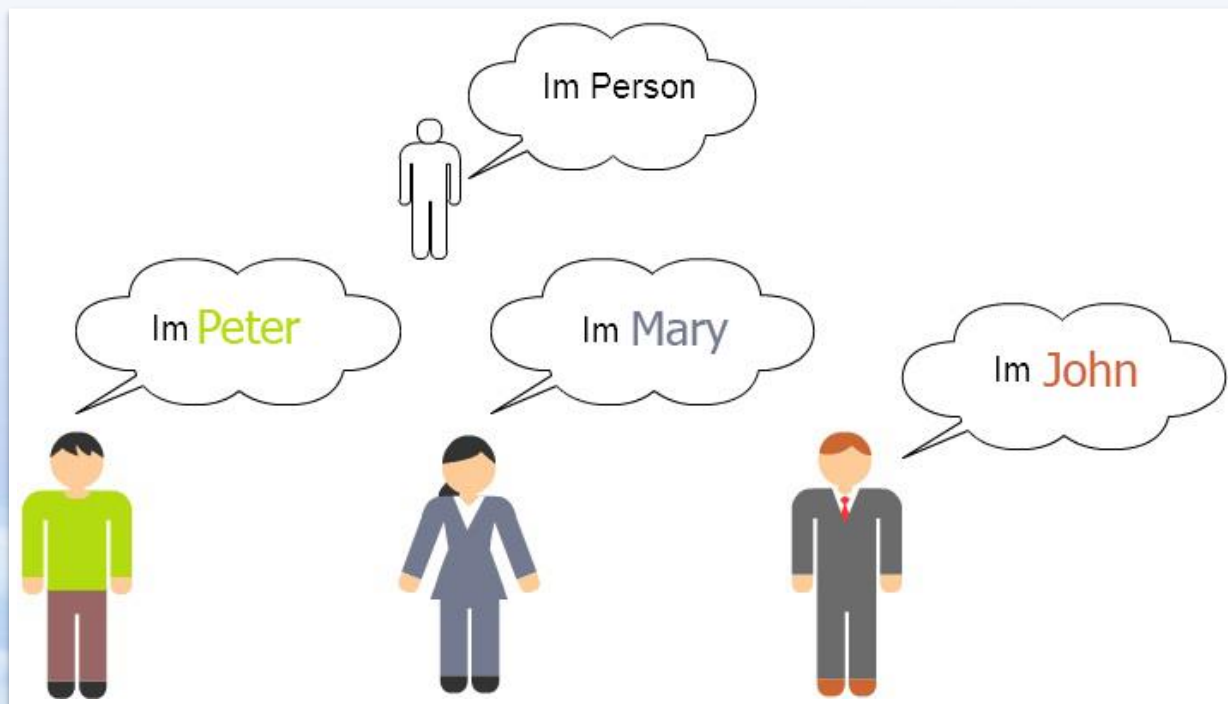
Properties:

- public
- private
- protected
- static
- ...



Class: lớp

- Lớp nghĩa là một khung kịch bản của một đối tượng. Trong lớp có các biến gọi là các thuộc tính (*properties*), và có các hàm gọi nó là phương thức (*method*)





Cấu trúc 1 class

Sử dụng class:

- Tạo đối tượng (object) mới
\$objCal1 = new Calculator(5, 3);
- Dùng đối tượng trở tới method
echo "
Tổng: " . \$objCal1->sum();
echo "
Hiệu: " . \$objCal1->sub();

- Tương tự:

```
$objCal2 = new Calculator(15, 5);  
echo "<br>Tổng: " . $objCal2->sum();  
echo "<br>Hiệu: " . $objCal2->sub();
```

- Truy cập vào thuộc tính:

```
echo "Số 1 obj1: " . $objCal1->so1;  
echo "Số 1 obj2: " . $objCal2->so1;
```

```
1 <?php  
2 class Calculator{  
3     //thuộc tính lớp  
4     public $so1;  
5     public $so2;  
6  
7     //phương thức khởi tạo của lớp  
8     public function __construct($so1, $so2){  
9         $this->so1 = $so1;  
10        $this->so2 = $so2;  
11    }  
12  
13    //các phương thức lớp  
14    public function sum(){ //cộng 2 số  
15        return $this->so1 + $this->so2;  
16    }  
17    public function sub(){ //trừ 2 số  
18        return $this->so1 - $this->so2;  
19    }  
20    public function mul(){ //nhân 2 số  
21        return $this->so1 * $this->so2;  
22    }  
23    public function div(){ //chia 2 số  
24        return $this->so1 / $this->so2;  
25    }  
26 }  
27 ?>
```



Bài tập 1

- Tạo class `GiangVien` gồm các thuộc tính: `hoTen`, `namSinh`, `lopHoc`
- Tạo 2 đối tượng là `gv1` và `gv2`. Gán giá trị cho các thuộc tính và in giá trị các thuộc tính từng giảng viên ra màn hình!



Bài tập 2

- Tạo class DiemThiKhoiA gồm các thuộc tính: Họ tên(hoTen), điểm các môn Toán(diemToan), Lý(diemLy), Hóa(diemHoa).
 - Viết phương thức khởi tạo
 - Viết phương thức tính điểm trung bình (diemTB), biết:
$$\text{diemTB} = (\text{diemToan} + \text{diemLy} + \text{diemHoa}) / 3$$
 - Triển khai các class đã tạo với các thuộc tính được gán mặc định



extends: kế thừa

- Một class có thể được kế thừa từ một class nào đó đã được khai báo trước. Class được kế thừa gọi là class cha, class kế thừa gọi là class con.
- Cú pháp:

```
class sClass extends pClass{  
    //khởi lệnh;  
}
```
- Class con sẽ được kế thừa tất cả các thuộc tính và phương thức bên trong class cha(trừ các thuộc tính hoặc phương thức được khai báo với từ khóa private).



Bài tập 3

- Tạo class `DiemThiUuTien` kế thừa từ class `DiemThiKhoiA`, có thêm thuộc tính Điểm ưu tiên (`diemUuTien`).
 - Viết phương thức tính điểm trung bình (`diemTB`), biết:
$$\text{diemtb} = (\text{diemToan} + \text{diemLy} + \text{diemHoa}) / 3 + \text{diemUuTien}$$
- Triển khai các class đã tạo với các thuộc tính được gán mặc định