Lambda và Closure là 1 dạng hàm đặc biệt(hàm vô định), nó giúp người dùng giải quyết được bài toán cho phép người lập trình chủ động trong việc tạo nhanh 1 hàm vô định để giải quyết vấn đề mà không cần thiết phải tạo thủ công 1 hàm hoàn chỉnh. Thay vì người dùng phải định nghĩa hàm riêng để dùng thì ta có thể dùng Lambda và Closure để viết trực tiếp

Lambda: Chỉ sử dụng biến nội tại

$test = function($param1,…){ … } ;

$test($param1,…);

Closure: Nhận được biến bên ngoài

$var; $test = function($param1,…) use($var){ … } ;

$test($param1,…);

Sử dụng tham chiếu để thay đổi giá trị $e khi giá trị $var thay đổi trong hàm => use(**&**$var)

Hàm array\_map($callback,$arr) : Lặp tất cả phần tử của mảng $arr truyền vào hàm callback. Coi phần tử mảng là tham số truyền vào.

Sử dụng Closure trong hướng đối tượng

Magic method \_\_call($method,$param): Được gọi khi ta gọi 1 phương thức không tồn tại trong đối tượng, $method là tên phương thức không tồn tại

Magic method \_\_invoke($param) : Được gọi khi gọi 1 đối tượng như 1 hàm

Is\_callable($this->$method): Kiểm tra $this->$method có phải 1 function không

Call\_user\_func(callback,$param): Gọi 1 hàm với đối số truyền vào là tên hàm và tham số

Call\_user\_func\_array(callback,$array) : Gọi 1 hàm với đối số truyền vào là tên hàm ( hoặc 1 class – array(‘Ten\_Ns\Ten\_Lop’, ‘Ten\_phuong\_thuc’)) và tham số là 1 mảng

Func\_get\_args() : Lấy tất cả đối số trong hàm đưa vào mảng;

Func\_num\_args(): Đếm số đối số người dùng truyền vào;

*\*callback: Hàm, hàm vô định, hàm định nghĩa trực tiếp*