**1. Kế thừa lồng là gì?**

Kế thừa lồng  hay còn gọi là[**kế thừa**](https://freetuts.net/tinh-ke-thua-trong-php-30.html) nhiều lớp, nghĩa là một lớp được kế thừa từ nhiều lớp. Lớp B kế thừa lớp A, lớp C kế thừa lớp B và thế là ta có luồng kế thừa  A -> B -> C. Một ví dụ khác thực tế hơn, giả sử Ông mình sinh ra Ba mình, Ba mình sinh ra mình như vậy mình kế thừa dòng máu của Ông mình và Ba mình.

**Ví dụ:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | class A {    }    class B extends A {    }    class C extends B {    } |

Nhìn ví dụ thật đơn giản phải không nào, nhưng đi sâu vào bên trong nó không đơn giản tí nào đâu. Vấn đề đặt ra là khi kế thừa nhiều cấp như vậy thì các biến (thuộc tính), các hàm (phương thức) ở các lớp cha được sử dụng ở lớp con như thế nào ? Để hiểu rõ hơn vấn đề này ta sẽ đi qua phần hai.

**2. Các mức độ truy cập trong kế thừa lồng**

Như ta đã biết có [3 mức độ truy cập](https://freetuts.net/cac-muc-truy-cap-private-protected-va-public-31.html) hay dùng nhất đó là **private, public và protected**. Cũng như trong kế thừa một cấp, tất cả những biến (thuộc tính) và hàm (phương thức) ở dạng public thì ở tất cả các lớp con dù ở cấp độ nào nó cũng có thể truy xuất vào được. Nếu ở dạng **private** thì chỉ dùng trong lớp đó. Nếu ở dạng **protected** thì tất cả các lớp kế thừa nó dù là lồng bao nhiêu lần thì cũng có thể sử dụng. Định nghĩa này chúng ta đã được học ở bài trước và cũng được giữ nguyên cho bài **kế thừa lồng** này.

**Ví dụ:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69 | // Lớp A  class A  {      protected $protected\_A = 'Protected';      private $private\_A = 'Private';      public $public\_A = 'Public';        private function showPrivate()      {          echo $this->private\_A;      }        protected function showProtected()      {          echo $this->protected\_A;      }        public function showPublic()      {          echo $this->public\_A;      }  }    // Lớp B Kế Thừa Lớp A  class B extends A  {      public function ClassB()      {          echo $this->protected\_A;      }  }    // Lớp C Kế Thừa Lớp B  class C extends B  {      public function showInfo()      {          // Lệnh này đúng vì nó truy xuất vào thuộc tính protected          $this->protected\_A = 'Nguyễn Văn A';            // Lệnh này đúng vì nó truy xuất vào thuộc tính public          $this->public\_A = 'Nguyễn Văn B';            // Lệnh này sai vì nó truy xuất vào thuộc tính private          $this->private\_A = 'Lệnh sai';      }  }    // ------------------//  // Chương trình chính//  // ------------------//    // Khởi tạo lớp C  // Lớp C được kế thừa từ lớp B  // mà lớp B kế thừa từ lớp A nên  // suy ra nó kế thừa từ 2 lớp A, B  $c = new C();    // Lệnh này đúng vì gọi đến hàm public của lớp cha A  $c->showPublic();    // Lệnh này sai vì nó gọi hàm protected của lớp cha A  $c->showProtected();    // Lệnh này sai vì nó gọi hàm private của lớp cha A  $c->showPrivate();    // Lệnh này đúng vì nó truy xuất vào hàm public của lớp cha B  $c->ClassB(); |