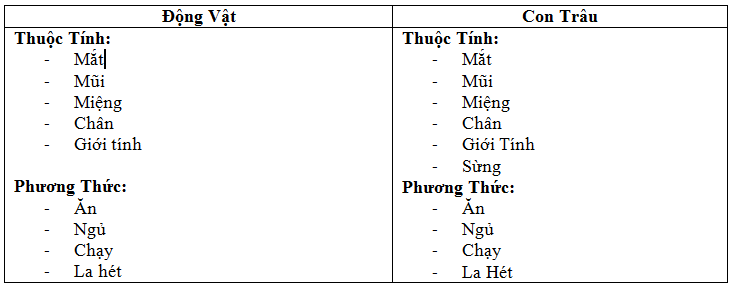
**1. Tính kế thừa là gì?**

Giả sử tôi có 2 lớp **Động Vật** và lớp **Con Trâu** có những thuộc tính và phương thức sau:

Nhìn vào hình ta thấy 2 đối tượng **Động Vật** và **Con Trâu** có những đặc điểm chung và riêng. đối tượng Con trâu chỉ có một đặc tính khác đó là nó có sừng, còn đối tượng động vật thì không có. Vậy ta có nhận xét như sau: Con trâu cũng là một động vật nên nó có những đặc tính của động vật, hay nói cách khác nó có những đặc điểm và hành động của động vật, vì vậy con trâu có thể kế thừa nhưng đặc điểm và phương thức của động vật.

Đấy là ta đang nói đến **tính kế thừa của các đối tượng trong thế giới thực**, còn trong lập trình thì như thế nào ?

Nếu khai báo thông thường thì ta sẽ tạo hai class như sau:

**Lớp Động Vật**:

|  |  |
| --- | --- |
| 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 | class DongVat  {      // Thuộc Tính      var $mat = '';      var $mui = '';      var $mieng = '';      var $chan = '';      var $gioitinh = '';        // Hàm, phương thức      function an(){          // lệnh      }        function ngu(){          // lệnh      }        function chay(){          // lệnh      }        function lahet(){          // lệnh      }  } |

**Lớp Con Trâu**:

|  |  |
| --- | --- |
| 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 | class ConTrau  {      // Thuộc Tính      var $mat = '';      var $mui = '';      var $mieng = '';      var $chan = '';      var $gioitinh = '';        // Thuộc tính này khác với động vật      var $sung = ''; // con trâu có sừng        // Hàm, phương thức      function an(){          // lệnh      }        function ngu(){          // lệnh      }        function chay(){          // lệnh      }        function lahet(){          // lệnh      }  } |

Rõ ràng 2 đối tượng Động Vật và Con Trâu có những [thuộc tính và phương thức](https://freetuts.net/lop-thuoc-tinh-phuong-thuc-cua-doi-tuong-28.html) giống nhau tại sao ta lại phải khai báo 2 đối tượng dài như vậy? Vì thế trong lập trình hướng đối tượng cung cấp chúng ta một tính chất để giải quyết vấn đề này đó là **tính kế thừa**.

Cú pháp như sau:

|  |  |
| --- | --- |
| 1  2 | class classCon extends classCha {  } |

Sau khi khai báo như vậy thì classCon sẽ có tất cả những thuộc tính, phương thức của classCha nên nó không cần phải viết lại.

Bài toán trên sẽ viết lại dưới dạng kế thừa như sau:

|  |  |
| --- | --- |
| 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 | // Lớp Cha  class DongVat  {      // Thuộc Tính      var $mat = '';      var $mui = '';      var $mieng = '';      var $chan = '';      var $gioitinh = '';        // Hàm, phương thức      function an(){          // lệnh      }        function ngu(){          // lệnh      }        function chay(){          // lệnh      }        function lahet(){          // lệnh      }  }    // Lớp Con  class ConTrau extends DongVat {        // Tất cả các thuộc tính khác đều kế thừa từ cha      // nên không cần viết lại      // chỉ riêng thuộc tính này là cha ko có nên ta phải khai báo      var $sung = '';        // Tất cả các hàm, phương thức đều kế thừa từ cha      // nên không cần viết lại  } |

**2. Gọi các phương thức và thuộc tính của lớp cha**

Sau khi kế thừa thì lớp có sẽ có những thuộc tính và phương thức ở lớp cha, vậy làm thế nào để lớp con có thể truy xuất đến những thuộc tính và phương thức đó? Ta sẽ tìm hiểu thông qua hai cách gọi đó là gọi từ bên trong lớp con và gọi từ bên ngoài lớp.

**Gọi bên trong lớp con**

Như khái niệm tính kế thừa, lớp con kế thừa từ lớp cha nên tất cả các phương thức và thuộc tính đều coi như là của nó (*coi như chứ không phải là của nó hẳn*) nên cách gọi cũng như cách nó gọi các phương thức thuộc tính của nó, đó là từ khóa $this->thuoctinh, $this->phuong\_thuc()**.**

Tuy nhiên để phân biệt hàm nào của cha, [hàm](https://freetuts.net/xay-dung-ham-trong-php-11.html) nào của con người ta hay gọi bằng từ khóa: parent::thuoctinh, parent::phuong\_thuc()**.**

**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 | <strong>/ Lớp Động Vật  class DongVat  {      // Thuộc tính      var $mat = '';      var $mui = '';        // Phương Thức      function an()      {          echo 'Dong Vat Dang An';      }  }    // Lớp Con Trâu  class ConTrau extends DongVat  {      function gioi\_thieu()      {          $this->mat = 'Đây là cái mặt';          $this->mui = 'Đây là cái mũi';          parent::an(); // xuất ra chuỗi "Động Vật Đang Ăn"      }  }</strong> |

**Gọi từ bên ngoài lớp**

Cách gọi cũng như cách 1 của phần **gọi bên trong lớp con**, tuy nhiên không được sử dụng cách 2 (parent::thuoctinh, parent::phuong\_thuc()).

**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 | // Lớp Động Vật  class DongVat  {      // Thuộc tính      var $mat = '';      var $mui = '';        // Phương Thức      function an()      {          echo 'Dong Vat Dang An';      }  }    // Lớp Con Trâu  class ConTrau extends DongVat  {      function gioi\_thieu()      {          $this->mat = 'Đây là cái mặt';          $this->mui = 'Đây là cái mũi';          parent::an(); // xuất ra chuỗi "Động Vật Đang Ăn"      }  }    // Chương Trình  $contrau = new ConTrau();    // Gọi đến hàm gioi\_thieu trong lớp Con Trâu  // nên xuất ra màn hình chuỗi "Động Vật Đang Ăn"  $contrau->gioi\_thieu();    // Trong hàm giới thiệu có gán giá trị cho 2  // thuộc tính mắt và mũi, giờ ta xuất ra màn hình  // xem giá trị nó là gì    echo $contrau->mat;  echo $contrau->mui; |