

Lập trình hướng đối tượng

Nội dung

1. Tổng quan về lập trình hướng đối tượng
 - ▶ Tóm tắt sự tiến hóa của các mô hình khái niệm
 - ▶ Khái niệm
 - ▶ Đối tượng
 - ▶ Lớp đối tượng
2. Xây dựng lớp đối tượng (class)
 - ▶ Tạo lớp
 - ▶ Khai báo các thuộc tính của lớp
 - ▶ Các phương thức của lớp
 - ▶ Minh họa xây dựng lớp phân số
3. Sử dụng class
4. Kế thừa

Tóm tắt sự tiến hóa của các

Mô hình 1

- Code xử lý chính
- CT nguồn được lưu trong 1 tập tin (trang.php)

Mô hình 2

- Code xử lý chính
- Hàm tự định nghĩa
- CT nguồn được lưu trong 1 tập tin (trang.php)

CT nguồn được lưu
trong 1 tập tin
(trang.php)

Mô hình 4

- Cải tiến từ mô hình 3
- CT nguồn được phân rã thành nhiều tập tin

Mô hình 3

- Cải tiến từ mô hình 2
- CT nguồn được phân rã thành nhiều tập tin

Nội dung

1. Tổng quan về lập trình hướng đối tượng
 - ▶ Tóm tắt sự tiến hóa của các mô hình khái niệm
 - ▶ Khái niệm
 - ▶ Đối tượng
 - ▶ Lớp đối tượng
2. Xây dựng lớp đối tượng (class)
 - ▶ Tạo lớp
 - ▶ Khai báo các thuộc tính của lớp
 - ▶ Các phương thức của lớp
 - ▶ Minh họa xây dựng lớp phân số
3. Sử dụng class
4. Kế thừa

Khái niệm

- ▶ Lập trình hướng đối tượng (Object-Oriented Programming- OOP) là một phương pháp thiết kế và phát triển phần mềm dựa trên kiến trúc lớp (class) và đối tượng (object).

Nội dung

1. Tổng quan về lập trình hướng đối tượng
 - ▶ Tóm tắt sự tiến hóa của các mô hình khái niệm
 - ▶ Khái niệm
 - ▶ Đối tượng
 - ▶ Lớp đối tượng
2. Xây dựng lớp đối tượng (class)
 - ▶ Tạo lớp
 - ▶ Khai báo các thuộc tính của lớp
 - ▶ Các phương thức của lớp
 - ▶ Minh họa xây dựng lớp phân số
3. Sử dụng class
4. Kế thừa

Đối tượng (object)

- ▶ Là những thực thể tồn tại trong thế giới thực
- ▶ Ví dụ:
 - ▶ Con người
 - ▶ Sinh viên Nguyễn Văn An
 - ▶ Nhân viên Trần Thị Thảo
 - ▶ Đồ vật
 - ▶ Phòng học C41
 - ▶ Máy in Laser Jet 4300
 - ▶ Chứng từ
 - ▶ Hóa đơn HD01
 - ▶ Đơn đặt hàng DDH_14022008_01

Đối tượng (object)

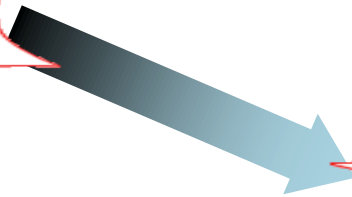
- ▶ Ví dụ: một chiếc xe hơi cụ thể với các thông tin về chiếc xe
 - ▶ Biển số xe
 - ▶ Hiệu xe
 - ▶ Màu sơn
 - ▶ Hãng sản xuất
 - ▶ Năm sản xuất



Đối tượng (object)

- ▶ Tiếp cận hướng đối tượng:
 - ▶ Là kỹ thuật cho phép biểu diễn tự nhiên các đối tượng trong thực tế với các đối tượng bên trong chương trình

**Đối tượng
trong thực tế**



**Đối tượng
trong chương
trình**

Đối tượng (object)

- ▶ Nhân cách hóa đối tượng

Hệ thống thực tế



Hệ thống tin học

nghiệp vụ f
đối tượng x

Đối tượng x
nghiệp vụ f

Đối tượng (object)

Hệ thống thực tế

hai



Hệ thống tin học

Ví dụ 1: Bài toán phương trình bậc hai

Giải phương trình bậc hai P

$$ax^2 + bx + c = 0$$

Đối tượng P, với

- Thông tin về đối tượng: hệ số **a**, **b** và **c**
- Hàm xử lý: **Giải phương trình**, ...

Đối tượng (object)

- ▶ Ví dụ 2: Tính tiền lương cho nhân viên

Hệ thống thực tế



Hệ thống tin học

Tính lương cho nhân viên NV: họ tên, ngày sinh, số con, hệ số lương, ...

Đối tượng NV, với

- Thông tin về đối tượng: họ tên, ngày sinh, số con, hệ số lương, ...
- Hàm xử lý: Tính lương, ...

Nội dung

1. Tổng quan về lập trình hướng đối tượng
 - ▶ Tóm tắt sự tiến hóa của các mô hình khái niệm
 - ▶ Khái niệm
 - ▶ Đối tượng
 - ▶ Lớp đối tượng
2. Xây dựng lớp đối tượng (class)
 - ▶ Tạo lớp
 - ▶ Khai báo các thuộc tính của lớp
 - ▶ Các phương thức của lớp
 - ▶ Minh họa xây dựng lớp phân số
3. Sử dụng class
4. Kế thừa

Lớp đối tượng (class)

- ▶ Là một khái niệm trong Lập trình hướng đối tượng, mô tả những thực thể có chung tính chất và hành vi. Class định nghĩa những thuộc tính và hành vi được dùng cho những đối tượng của lớp đó
- ▶ Kết quả của sự **TRỪU TƯỢNG HOÁ (Abstraction)** các đối tượng:
 - ▶ Cùng loại
 - ▶ Cùng các thông tin mô tả về đối tượng

Lớp đối tượng

Lớp đối tượng (class)

► Ví dụ:



Lớp đối tượng (class)

- ▶ Ví dụ: lớp nhân viên



Nội dung

1. Tổng quan về lập trình hướng đối tượng
 - ▶ Tóm tắt sự tiến hóa của các mô hình khái niệm
 - ▶ Khái niệm
 - ▶ Đối tượng
 - ▶ Lớp đối tượng
2. Xây dựng lớp đối tượng (class)
 - ▶ Tạo lớp
 - ▶ Khai báo các thuộc tính của lớp
 - ▶ Các phương thức của lớp
 - ▶ Minh họa xây dựng lớp phân số
3. Sử dụng class
4. Kế thừa

Tạo lớp

- ▶ Cấu trúc của một lớp bao gồm từ khóa **class**, **tên lớp**, cặp dấu **{}** để bao lại các câu lệnh bên trong lớp.
- ▶ Bên trong lớp có các thuộc tính và

```
class <tên lớp>
```

```
{ // khai báo các thuộc tính của lớp  
  // gán và lấy giá trị của thuộc tính  
  // các phương thức của lớp  
}
```

Nội dung

1. Tổng quan về lập trình hướng đối tượng
 - ▶ Tóm tắt sự tiến hóa của các mô hình khái niệm
 - ▶ Khái niệm
 - ▶ Đối tượng
 - ▶ Lớp đối tượng
2. Xây dựng lớp đối tượng (class)
 - ▶ Tạo lớp
 - ▶ Khai báo các thuộc tính của lớp
 - ▶ Các phương thức của lớp
 - ▶ Minh họa xây dựng lớp phân số
3. Sử dụng class
4. Kế thừa

Khai báo các thuộc tính của lớp

- ▶ Thuộc tính: thành phần lưu trữ các tính chất, đặc điểm của đối tượng.
- ▶ Ứng với mỗi thuộc tính: khai báo một biến và chọn một từ khóa phía trước phù hợp với mục đích sử dụng.
- ▶ Các từ khóa trước biến:
 - public: có thể sử dụng ở bất kỳ đâu.
 - private: chỉ có thể sử dụng trong class hiện hành
 - protected: có thể sử dụng trong các lớp kế thừa.

Khai báo các thuộc tính của

► **lớp**
Cú pháp:

```
<?php  
public <tên thuộc tính 1>;  
private <tên thuộc tính 1>;  
protected <tên thuộc tính 1>;  
...  
?>
```

Lớp do người dùng tự xây dựng

- ▶ Khai báo các thuộc tính của lớp
 - ▶ **Chú ý:** khi khai báo một lớp, có thể thiết lập những giá trị mặc định ban đầu cho tất cả các thuộc tính được tạo ra từ lớp

```
<?php
class PHAN_SO
{
    private $tu_so = 1;
    private $mau_so = 1;
    ...
}
?>
```

Nội dung

1. Tổng quan về lập trình hướng đối tượng
 - ▶ Tóm tắt sự tiến hóa của các mô hình khái niệm
 - ▶ Khái niệm
 - ▶ Đối tượng
 - ▶ Lớp đối tượng
2. Xây dựng lớp đối tượng (class)
 - ▶ Tạo lớp
 - ▶ Khai báo các thuộc tính của lớp
 - ▶ Các phương thức của lớp
 - ▶ Minh họa xây dựng lớp phân số
3. Sử dụng class
4. Kế thừa

Nội dung

1. Tổng quan về lập trình hướng đối tượng
 - ▶ Tóm tắt sự tiến hóa của các mô hình khái niệm
 - ▶ Khái niệm
 - ▶ Đối tượng
 - ▶ Lớp đối tượng
2. Xây dựng lớp đối tượng (class)
 - ▶ Tạo lớp
 - ▶ Khai báo các thuộc tính của lớp
 - ▶ Các phương thức của lớp
 - ▶ Minh họa xây dựng lớp phân số
3. Sử dụng class
4. Kế thừa

Các phương thức của lớp

- ▶ Là chức năng mà đối tượng có thể thực hiện.

```
<?php  
function <tên phương thức> (các tham số truyền vào  
nếu có)  
{  
    // khối lệnh  
}  
?>
```

Các phương thức của lớp

► Chú ý:

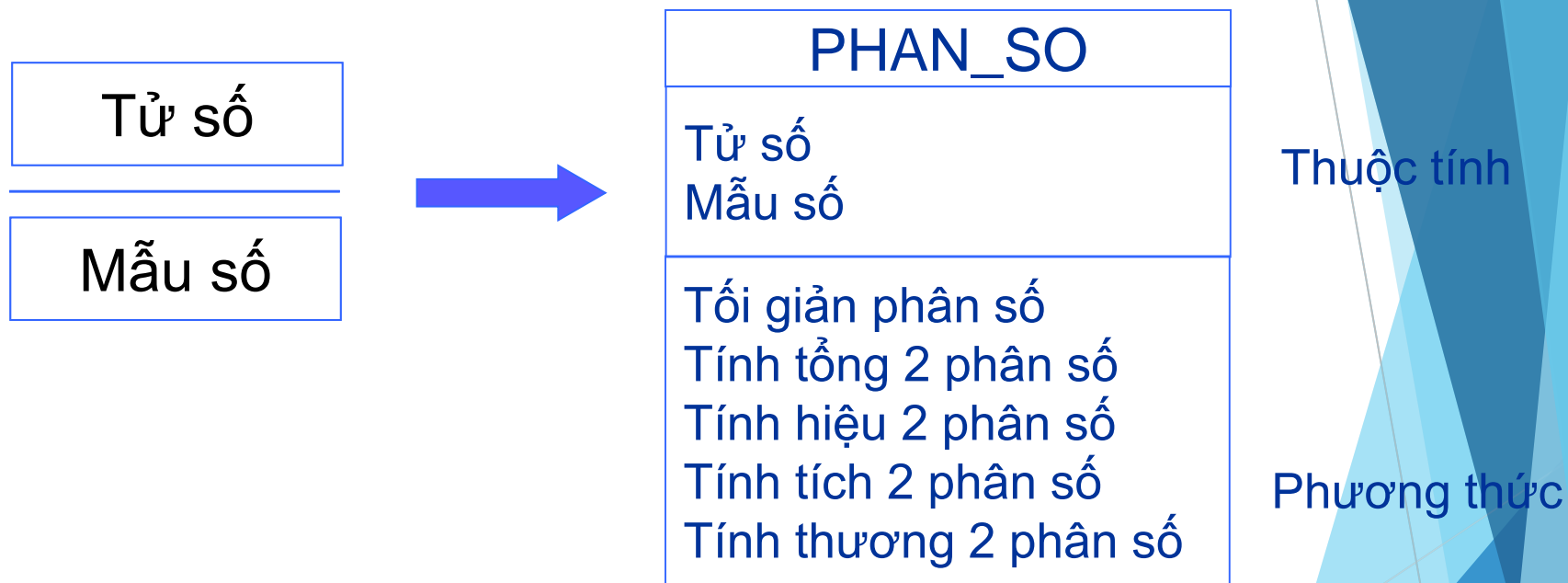
- Trong các phương thức của lớp, có thể truy cập các thuộc tính hay phương thức thông qua con trỏ **\$this**
- Con trỏ **\$this** được dùng để chỉ lớp hiện tại đang làm việc.
- Ví dụ: nếu một thuộc tính của lớp phân số có tên là \$tu_so thì có thể tham chiếu đến nó như sau:

\$this->tu_so

Nội dung

1. Tổng quan về lập trình hướng đối tượng
 - ▶ Tóm tắt sự tiến hóa của các mô hình khái niệm
 - ▶ Khái niệm
 - ▶ Đối tượng
 - ▶ Lớp đối tượng
2. Xây dựng lớp đối tượng (class)
 - ▶ Tạo lớp
 - ▶ Khai báo các thuộc tính của lớp
 - ▶ Các phương thức của lớp
 - ▶ Minh họa xây dựng lớp phân số
3. Sử dụng class
4. Kế thừa

Minh họa xây dựng lớp phân số



Minh họa xây dựng lớp phân số

```
<?php
class PHAN_SO
{
    // khai báo các thuộc tính
    private $tu_so;
    private $mau_so;
    // gán và lấy giá trị của các thuộc tính
    // xây dựng các phương thức cần thiết cho lớp phân số
}
?>
```

Minh họa xây dựng lớp phân số

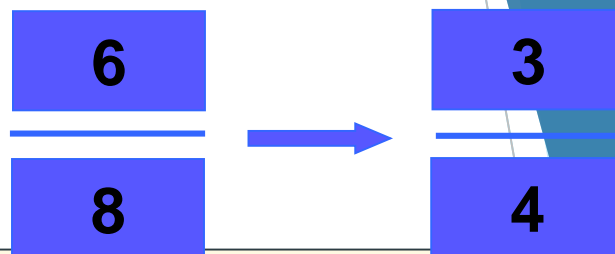
```
<?php
class PHAN_SO
{
    ...
    // xây dựng các phương thức cần thiết cho lớp phân số
    function khai_tao_phan_so($ptu_so,$pmau_so)
    {
        $this->tu_so = $ptu_so;
        $this->mau_so = $pmau_so;
    }
    ...
}
?>
```

Minh họa xây dựng lớp phân

```
<?php
class PHAN_SO
{
    ...
    function USCLN($a, $b)
    {
        $so_nho = ($a<$b)? $a:$b;
        for($i=$so_nho; $i>0;$i--)
            if( ($a%$i)==0 && ($b%$i)==0 )
            {
                return $i;
                break;
            }
    }
    ...
}
?>
```

Minh họa xây dựng lớp phân số

► **Số** Tối giản phân số



```
<?php
class PHAN_SO
{
    ...
    function Toi_gian_phan_so()
    {
        $uscln = USCLN($this->tu_so, $this->mau_so);
        $this->tu_so = $this->tu_so / $uscln;
        $this->mau_so = $this->mau_so / $uscln;
        return PHAN_SO;
    }
    ...
}
?>
```


Minh họa xây dựng lớp phân số

► **Số** Tính tổng hai phân số

$$\frac{a}{b} + \frac{c}{d} = \frac{a*d + c*b}{b*d}$$

Phân số thứ nhất

Phân số thứ hai

Phân số tổng

Minh họa xây dựng lớp phân

```
<?php
class PHAN_SO
{
    ...
    function tong($ptu_so, $pmau_so)
    {
        $phan_so = new PHAN_SO();
        $phan_so->khoi_tao($ptu_so,$pmau_so);
        $phan_so->tu_so = ($this->tu_so * $phan_so -> mau_so)
            + ($phan_so->tu_so * $this->mau_so);
        $phan_so->mau_so = $this->mau_so * $phan_so -> mau_so;
        $phan_so->Toi_gian_phan_so();
        return $phan_so;
    }
    ... (tiếp tục thực hiện các phương thức khác của phân số)
}
?>
```

Nội dung

1. Tổng quan về lập trình hướng đối tượng
 - ▶ Tóm tắt sự tiến hóa của các mô hình khái niệm
 - ▶ Khái niệm
 - ▶ Đối tượng
 - ▶ Lớp đối tượng
2. Xây dựng lớp đối tượng (class)
 - ▶ Tạo lớp
 - ▶ Khai báo các thuộc tính của lớp
 - ▶ Các phương thức của lớp
 - ▶ Minh họa xây dựng lớp phân số
3. Sử dụng class
4. Kế thừa

Sử dụng class (lớp)

► Khởi tạo đối tượng

- Dùng từ khóa **new** tiếp đó là **tên lớp**
- Cú pháp: **\$<tên biến đối tượng> = new**

- Ví dụ:

```
<?php
    $phan_so = new PHAN_SO();
?>
```

Sử dụng class (lớp)

- ▶ Gán giá trị cho các thuộc tính của lớp

Cú pháp: `$<tên biến đối tượng>-><tên`

```
<?php
    $phan_so->tu_so = 2;
    $phan_so->mau_so = 4;
?>
```

Sử dụng class (lớp)

- ▶ Gọi sử dụng các phương thức của lớp

- ▶ Phương thức không có giá trị trả về

- ▶ Gọi tên của phương thức với các giá trị truyền vào cho tham số nếu có

- ▶ Cú pháp: `$<tên biến đối tượng>-><tên`

```
<?php  
$phan_so->khoi_tao_phan_so($ptu_so,$pmau_so); ?>
```

- ▶ Ví dụ: gọi hàm khởi tạo phân số

Sử dụng class (lớp)

► Gọi sử dụng các phương thức của lớp

► Phương thức có giá trị trả về

- Sử dụng biến để nhận giá trị trả về của phương thức
- Cú pháp: `$<tên biến nhận giá trị> = $<tên biến đối tượng>-> <tên phương thức>`(các giá trị truyền vào nếu có);
- Ví dụ: gọi hàm tính tổng hai phân số

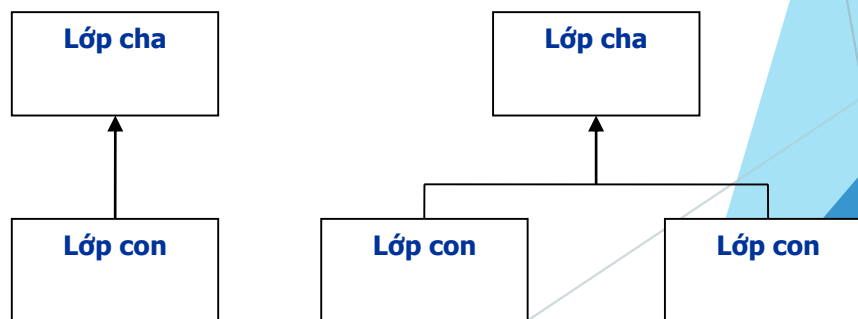
```
<?php
    $phan_so_2 = New PHAN_SO();
    $phan_so_2 = $phan_so->tong($a,$b);
?>
```

Nội dung

1. Tổng quan về lập trình hướng đối tượng
 - ▶ Tóm tắt sự tiến hóa của các mô hình khái niệm
 - ▶ Khái niệm
 - ▶ Đối tượng
 - ▶ Lớp đối tượng
2. Xây dựng lớp đối tượng (class)
 - ▶ Tạo lớp
 - ▶ Khai báo các thuộc tính của lớp
 - ▶ Các phương thức của lớp
 - ▶ Minh họa xây dựng lớp phân số
3. Sử dụng class
4. Kế thừa

Khái niệm

- ▶ Kế thừa: là khả năng xây dựng một lớp mới dựa trên các định nghĩa của một lớp đã có.
- ▶ Lớp đã có gọi là lớp cha, lớp mới phát sinh gọi là lớp con (subclass) và kế thừa tất cả các thành phần của lớp cha, có thể mở rộng tính năng các thành phần kế thừa cũng như bổ sung thêm các thành phần mới.



Kế thừa

► Thiết lập kế thừa

- Dùng từ khóa **extends** để khai báo **lớp**

```
class <tên lớp con> extends <tên lớp cha>
{
    ...
}
```

Kế thừa

► Chồng hàm:

- Là hai phương thức có cùng tên được khai báo trong hai lớp “lớp cha” và “lớp con”. Hai phương thức này có nội dung khác nhau.
- Khi khởi tạo “lớp con”, phương thức được gọi có cùng tên này được hiểu là phương thức khai báo trong “lớp con”.
- Khi muốn truy cập đến các phương thức đã bị nạp chồng (phương thức khai báo trong “lớp cha”) thì tham chiếu chúng với từ khóa **parent** và toán tử tham chiếu **::**, tiếp theo là phương thức hay thành viên cần tham chiếu.
- Cú pháp: **parent::tên_phương_thức**

Kế thừa

► Ví dụ về sử dụng parent::

```
class A
{
    function A()
    {
        print "Khởi tạo A<br>" ;
    }
}

class B extends A
{
    function B()
    {
        parent::A () ;
        print "Khởi tạo B<br>" ;
    }
}
```

```
class C extends B
{
    function C()
    {
        parent::B ();
        print "Khởi tạo C<br>" ;
    }
}

$o = new C;
```

Kết quả:

Khởi tạo A
Khởi tạo B
Khởi tạo C

Kế thừa

► Chồng hàm

- Ví dụ: phương thức tính lương cho nhân viên và phương thức tính lương cho nhân viên sản xuất (kế thừa lớp nhân viên) được gọi là “chồng hàm”

Lớp nhân viên

```
function tinh_luong()  
{  
    return $this->luong_can_ban *  
    $this->he_so_luong;  
}
```

Lớp nhân viên sản xuất

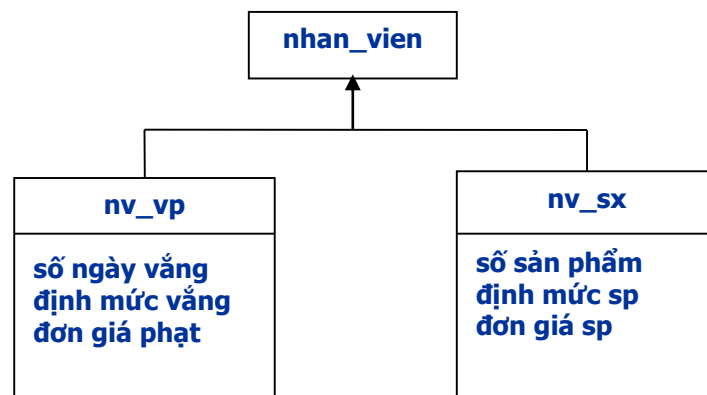
```
function tinh_luong()  
{  
    return $this->so_sp * $don_gia_sp  
    + $this->tien_thuong();  
}
```

Kế thừa

- ▶ Ví dụ kế thừa
 - ▶ Xây dựng lớp nhân viên gồm có các thuộc tính: họ tên, giới tính, ngày sinh, ngày vào làm, hệ số lương, số con, lương căn bản và các phương thức: tính lương, trợ cấp, tiền thưởng.
 - ▶ Xây dựng lớp nhân viên văn phòng kế thừa từ lớp nhân viên có thêm các thuộc tính riêng: số ngày vắng, định mức vắng, đơn giá phạt và các phương thức tính tiền phạt, trợ cấp, tiền lương.
 - ▶ Xây dựng lớp nhân viên sản xuất kế thừa từ lớp nhân viên có thêm các thuộc tính riêng: số sản phẩm, định mức sản phẩm, đơn giá sản phẩm và các phương thức tính tiền thưởng, trợ cấp, tiền lương.

Kế thừa

► Ví dụ kế thừa



- Tính tiền phạt nếu vắng quá số ngày quy định
- Tính trợ cấp căn cứ theo số con và giới tính
- Tính tiền lương theo lương cơ bản - tiền phạt

- Tính tiền thưởng căn cứ vào số sản phẩm và định mức sản phẩm
- Tính trợ cấp căn cứ vào số con và tăng ca
- Tính tiền lương theo (số sản phẩm * đơn giá sản phẩm) + tiền thưởng

Kế thừa

► Ví dụ kế thừa: các thuộc tính của lớp

```
class nhan_vien
{
    protected $ho_ten;
    protected $gioi_tinh;
    protected $ngay_sinh;
    protected $ngay_vao_lam;
    protected $he_so_luong;
    protected $so_con;
    protected $luong_cb = 450000;
    ...
}
```

nhan_vien
\$ho_ten
\$gioi_tinh
\$ngay_sinh
\$ngay_vao_lam
\$he_so_luong
\$so_con
\$luong_cb
tro_cap()
tien_thuong()
tien_luong()

Kế thừa

- ▶ Ví dụ kế thừa: hàm khởi tạo thông tin nhân viên

```
class nhan_vien
{
    ...
    function khai_tao_thong_tin_chung($ho_ten, $gioi_tinh,
    $ngay_sinh, $ngay_vao_lam, $he_so_luong, $so_con)
    {
        $this->ho_ten = $ho_ten;
        $this->gioi_tinh = $gioi_tinh;
        $this->ngay_sinh = $ngay_sinh;
        $this->ngay_vao_lam = $ngay_vao_lam;
        $this->he_so_luong = $he_so_luong;
        $this->so_con = $so_con;
    }
    ...
}
```

Kế thừa

- ▶ Ví dụ kế thừa: tính trợ cấp, tiền thưởng, tiền lương

```
class nhan_vien
{
    ...
    function tro_cap()
    {
        return $this->so_con * 100000;
    }
    function tien_thuong()
    {
        return (date("Y") - date("Y", $this -> ngay_vao_lam)) * 500000;
    }
    ...
}
```

Kế thừa

- ▶ Ví dụ kế thừa: tính trợ cấp, tiền thưởng, tiền lương

```
class nhan_vien
{
    ...
    function tien_luong()
    {
        return $this->luong_cb * $this->he_so_luong;
    }
    ...
}
```

Kế thừa

- ▶ Ví dụ kế thừa: các thuộc tính riêng của lớp nhân viên văn phòng

```
class nv_vp extends nhan_vien
{
    private $so_ngay_vang;
    private $dm_vang = 2;
    private $don_gia_phat = 25000;
    ...
}
```

nv_vp
\$so_ngay_vang
\$dm_vang
\$don_gia_phat
tro_cap()
tien_phat()
tien_luong()

Kế thừa

- ▶ Ví dụ kế thừa: tính tiền phạt, trợ cấp, tiền lương

```
class nv_vp
{
    ...
    function tien_phat()
    {
        if ($this->so_ngay_vang > $this->dm_vang)
            return ($this->so_ngay_vang - $this->
dm_vang)* $this->don_gia_phat;
        else
            return 0;
    }
    ...
}
```

Kế thừa

- ▶ Ví dụ kế thừa: tính tiền phạt, trợ cấp, tiền lương

```
class nv_vp
{
    ...
    function tro_cap()
    {
        if($this->gioi_tinh ==1)
            return $this->so_con * 100000 *1.2;
        else
            return $this->so_con * 100000;
    }
    ...
}
```

Kế thừa

- ▶ Ví dụ kế thừa: tính tiền phạt, trợ cấp, tiền lương

```
class nv_vp
{
    ...
    function tien_luong()
    {
        return ($this->luong_cb * $this->he_so_luong)
        - $this->tien_phat();
    }
    ...
}
```

Kế thừa

- ▶ Ví dụ kế thừa: các thuộc tính riêng của lớp nhân viên sản xuất

```
class nv_sx extends nhan_vien
{
    private $so_sp;
    private $dm_sp = 100;
    private $don_gia_sp = 12000;
    ...
}
```

nv_sx
\$so_sp \$dm_sp \$don_gia_sp
tro_cap() tien_thuong() tien_luong()

Kế thừa

- ▶ Ví dụ kế thừa: tính tiền thưởng, trợ cấp, tiền lương

```
class nv_sx
{
    ...
    function tien_thuong()
    {
        if($this->so_sp > $this->dm_sp)
            return (($this->so_sp - $this->dm_sp) *
                $this->don_gia_sp * 0.05);
        else
            return 0;
    }
    ...
}
```

Kế thừa

- ▶ Ví dụ kế thừa: tính tiền thưởng, trợ cấp, tiền lương

```
class nv_sx
{
    ...
    function tro_cap($tang_ca)
    {
        if($tang_ca==1)
            return (($this->so_con * 100000) +200000);
        else
            return ($this->so_con * 100000);
    }
    ...
}
```

Kế thừa

- ▶ Ví dụ kế thừa: tính tiền thưởng, trợ cấp, tiền lương

```
class nv_sx
{
    ...
    function tien_luong()
    {
        return (($this->so_sp * $this->don_gia_sp)
            + $this->tien_thuong());
    }
    ...
}
```

Kế thừa

► Ví dụ kế thừa:

QUẢN LÝ NHÂN VIÊN			
Họ và tên:	<input type="text" value="Nguyễn Thiên Trang"/>	Số con:	<input type="text" value="2"/>
Ngày sinh:	<input type="text" value="10/10/1978"/>	Ngày vào làm:	<input type="text" value="20/11/2005"/>
Giới tính:	<input type="radio"/> Nam <input checked="" type="radio"/> Nữ	Hệ số lương:	<input type="text" value="2.0"/>
Loại nhân viên:	<input checked="" type="radio"/> Văn phòng <input type="radio"/> Sản xuất		
	Số ngày vắng : <input type="text" value="3"/>	Số sản phẩm: <input type="text"/>	Tăng ca: <input type="radio"/> Có <input type="radio"/> Không
Tiền lương:	<input type="text" value="875.000 VNĐ"/>	Trợ cấp:	<input type="text" value="240.000 VNĐ"/>
Thực lĩnh:		<input type="text" value="1.115.000 VNĐ"/>	
<input type="button" value="Tính lương"/>			

Thảo luận

