



Chương 3

Lớp và đối tượng

Nội dung

- ❖ Khái niệm đối tượng
- ❖ Các mức truy xuất
- ❖ Các thành phần của lớp



QUẢN LÝ TUYỂN DỤNG



QUẢN LÝ NHÂN VIÊN



ĐÁNH GIÁ NHÂN SỰ



QUẢN LÝ TIỀN LƯƠNG

Bài toán quản lý nhân viên

NHÂN VIÊN HÀNH CHÍNH



Thông tin/đặc điểm nhân viên



Thông tin nhân viên



Lý lịch

- ❖ Họ và tên
- ❖ Ngày tháng năm sinh
- ❖ Tốt nghiệp đại học

Thông tin trong công ty

- ❖ Bộ phận làm việc
- ❖ Chức vụ làm việc
- ❖ Mức lương hiện tại.

Lương

Hệ số lương

Ngày vào công ty.

....

.....



Thông tin về lương



Mã	A0213001
Tên	HOÀNG VĂN TOÀN
Ngày vào	1/3/1992
Bộ phận	PP [Điều hành sản xuất/Production control]
Nơi làm	08 [Hồ Chí Minh]
Chức vụ	610 [Trưởng ca]



Mức lương

Loại giá trị	Thời gian	...
Bậc lương	0	...
Lương cơ bản	3,284,800	VND
Ngày hiệu lực	21/06/2005	...
Trả loại tiền	VND	Vietnamese Đồng
% lương thêm thuế thu nhập cá nhân		0
Trả qua N.H	<input checked="" type="checkbox"/>	101
Số T.K N.Hàng	A004511-2	
Khoảng cách (Km)	10	Loại xe
		Motobike

Thuế

Loại	VN1	Vietnamese Tax Type 1 (Normal)
G.trị cố định	0	

☐ Lương ròng (Net) ☒ Lương gộp

☒ Ứng lương 1,500,000

☒ Lương tháng ☒ Lương tháng 13

Nhóm lương GRP1

☒ Phụ cấp ... ☒ Thưởng khác ...

☒ Làm ngoài giờ ☒ Làm ca đêm

☒ Nghỉ trừ lương ☒ BHYT

☒ Khoản trừ khác ... ☒ BHXH

☐ Lương người NN?



Lưu



Thông tin thêm



Đóng

Công thức tính lương

Công thức tính mức tiền của hệ số chênh lệch bảo lưu (nếu có):

$$\left| \begin{array}{l} \text{Mức tiền của hệ số chênh lệch bảo lưu} \\ \text{thực hiện từ ngày 01 tháng 7 năm 2018} \end{array} \right| = \left| \begin{array}{l} \text{Mức lương cơ sở 1.390.000} \\ \text{đồng/tháng} \end{array} \right| \times \left| \begin{array}{l} \text{Hệ số chênh lệch bảo lưu hiện} \\ \text{hưởng (nếu có)} \end{array} \right|$$

Đối với đại biểu Hội đồng nhân dân các cấp, căn cứ vào hệ số hoạt động phí đối với đại biểu Hội đồng nhân dân các cấp theo quy định của pháp luật hiện hành để tính mức hoạt động phí theo công thức sau:

$$\left| \begin{array}{l} \text{Mức hoạt động phí thực hiện từ ngày 01} \\ \text{tháng 7 năm 2018} \end{array} \right| = \left| \begin{array}{l} \text{Mức lương cơ sở 1.390.000} \\ \text{đồng/tháng} \end{array} \right| \times \left| \begin{array}{l} \text{Hệ số hoạt động phí theo} \\ \text{quy định} \end{array} \right|$$

Cách quản lý

- ❖ Nhiều tập hợp của nhiều thành phần ?
- ❖ 1 tập hợp của nhiều thành phần???
- ❖ 1 tập hợp của **một loại dữ liệu** bao gồm nhiều thành phần ??



Đối tượng

Đối tượng

- ❖ Khái niệm **đối tượng** (object) trong lập trình hướng đối tượng giống như một đối tượng cụ thể trong thế giới thực.
- ❖ Mỗi đối tượng có các thuộc tính và các hành vi riêng.
 - **Thuộc tính** (attribute) mô tả đặc điểm của đối tượng.
 - Hành vi là phương thức hoạt động của đối tượng, gọi tắt là **phương thức** (method).

Các loại đối tượng

- ❖ Đối tượng giao diện
- ❖ Đối tượng dữ liệu
- ❖ Đối tượng xây dựng sẵn
- ❖ Đối tượng tự xây dựng
- ❖ Loại khác như dự án/module

Đối tượng trên giao diện

- ❖ Ví dụ nút lệnh
- ❖ Button
- ❖ Thuộc tính
 - Width
 - Height
 - Padx/pady
 - Color(bg/highlightcolor...)
 - Font
 - ...
- ❖ Phương thức
 - Place
 - Pack
 - ...command = tenham



Đối tượng dữ liệu

❖ Như mô tả thông tin cá nhân/ hàng hóa
VVV

Đối tượng

❖ Ví dụ: Phân số

❖ Thuộc tính

- Tử số
- Mẫu số

❖ Phương thức

- Cộng, trừ, nhân, chia
- Tối giản
- Nghịch đảo

Fraction

Thuộc tính

- Numerator
- Denominator

Phương thức

- Fraction() khởi tạo
- Add,sub,mul,div
- Compact

Đối tượng

❖ Ví dụ: xe hơi

- Màu trắng
- 4 cửa
- 4 bánh
- Hiệu Toyota
- Chạy tới
- Chạy lui
- Xe dừng
- ...

Đối tượng

❖ Đối tượng:

❖ XeHoi

❖ Hiệu xe

❖ Màu xe

❖ Số bánh xe

❖ Số cửa

❖ Chạy tới

❖ Chạy lui

❖ Dừng xe

❖ Tên đối tượng

❖ Thuộc tính

❖ Phương thức

Đối tượng

- ❖ Các đối tượng có các thuộc tính và phương thức giống nhau được gom nhóm thành một lớp để phân biệt với các đối tượng khác và dễ quản lý.
- ⇒ Một **lớp** (class) là sự phân loại của các đối tượng hay là kiểu (type) của đối tượng.
- ❖ Ví dụ:
 - Các chiếc xe Toyota, Honda, Porsche thuộc lớp xe hơi.
 - Các con chó giữ nhà, chó săn, chó kiểng thuộc lớp chó.

Khai báo đối tượng

```
class TenLop:
    #khai báo biến thuộc tính
    thuoctinh_1
    thuoctinh_2
    thuoctinh_n

    #khai báo phương thức
    def PhuongThuc1():
    def PhuongThuc2():
```


Khởi tạo đối tượng

❖ Dùng hàm khởi tạo

```
def __init__(self, [danh sach thuoc tinh])
```

❖ Sử dụng

```
tenBien = TenLop([danh sach giá trị])
```

Khởi tạo đối tượng Ví dụ

❖ Class Cnhanvien:

- `def __init__(self, ms, hoten, ngay)`
 - `self.id=ms`
 - `self.name=hoten`
 - `self.date=ngay`

`nv=NhanVien('12', 'Truong van A', '22/2/2002')`

Biến thuộc tính

- ❖ Lưu trữ thuộc tính của đối tượng
- ❖ Sử dụng cách khai báo biến thông thường
- ❖ Thêm mức truy xuất
- ❖ Ví dụ
 - tuSo
 - __mauSo
 - _soBanhXe
- ❖ Truy xuất biến lớp:
 - tenDoiTuong.bienLop

Phương thức (method)

- ❖ Các hàm bên trong lớp
- ❖ Mô tả hoạt động của đối tượng
- ❖ Truy xuất phương thức
 - `tenBien.TenPhuongThuc();`
 - Ví dụ:
 - `a = PhanSo();`
 - `a.ToiGian();`
 - `b = a.NhanMotSo(2);`

Hàm khởi tạo (constructor)

- ❖ Khởi tạo giá trị ban đầu cho đối tượng khi khai báo
- ❖ Không khai báo kiểu trả về
- ❖ Khai báo:
 - `__init__(self,.....)`

Constructor

- ❖ Nếu không khai báo constructor thì trình biên dịch sẽ tự thêm constructor mặc định
- Mỗi class **chỉ có 1** constructor

Constructor

```
class phanso:  
    tuso=0  
    mauso=1  
  
    def __init__(self,ts,ms):  
        self.tuso = ts  
        self.mauso = ms
```

Constructor

- ❖ Khởi tạo đối tượng phân số có tử số = 0, mẫu số = 1:
`b = PhanSo(0, 1);`
- ❖ Khởi tạo đối tượng phân số có tử số = 1, mẫu số = 2
 - `PhanSo c = PhanSo(1, 2)`

Tham chiếu self

- ❖ Tham chiếu đến đối tượng hiện hành
- ❖ Trong trường hợp đặt tên tham số trùng với tên biến thuộc tính, sử dụng **self** để chỉ biến thuộc tính

```
class PhanSo
{
    tuSo
    mauSo

    public PhanSo(self, int tuSo, int mauSo)
    {
        self.tuSo = tuSo;
        self.mauSo = mauSo;
    }
}
```

Từ khóa self

- ❖ Từ khóa self cho biết đây là thuộc tính, phương thức thuộc đối tượng (có thể sử dụng từ khóa khác thay thế miễn để ở đầu danh sách tham số)

```
class MyClass:
```

```
    a = 10
```

```
    def func(self):
```

```
        ```noi dung ham
```

```
ob = MyClass()
```

```
ob.func()
```





❖ class Person:

❖ def \_\_init\_\_(mypara, name, age):

❖     mypara.name = name

❖     mypara.age = age

❖ def myfunc(abc):

❖     print("Hello my name is " + abc.name)

❖ p1 = Person("John", 36)

❖ p1.myfunc()

# Từ khóa pass

- ❖ Khi viết phương thức trong lớp nếu không có nội dung thì có thể dùng từ khóa **pass** để không bị báo lỗi

# Truyền đối tượng vào phương thức

- ❖ Đối tượng là kiểu tham chiếu
- ❖ Nếu có câu lệnh tác động đến giá trị các biến lớp của đối tượng trong phương thức  
→ đối tượng bị thay đổi sau khi hoàn tất phương thức.

# Xóa bỏ thuộc tính và đối tượng

❖ Thuộc tính của đối tượng có thể bị xóa bằng lệnh *del*

*c1 = phanso(2,3)*

❖ Xóa thuộc tính

*del c1.tuso*

❖ Xóa đối tượng

*del c1*

# Mức truy xuất

## ❖ public

- Có thể truy xuất ở mọi nơi khác
- `tenbien`

## ❖ protected

- Chỉ có thể truy xuất ở trong class hoặc class kế thừa
- `_tenbien` (1 gạch dưới)
- Lưu ý trong python mức này không rõ ràng

## ❖ private

- Chỉ có thể truy xuất ở trong class
- `__tenbien` (2 gạch dưới)

# static

- ❖ Dùng cho các thành phần lớp không phụ thuộc vào đối tượng cụ thể
- ❖ Có thể truy cập trong bản thân lớp
- ❖ Được truy cập trực tiếp từ tên lớp
- ❖ Phương thức static chỉ truy cập được những thành viên static của lớp
- ❖ Cách viết là không có từ khóa **self**
- ❖ Truy cập:
  - TenLop.TenBien
  - TenLop.TenPhuongThuc()

# Nhiều hàm Constructor ?

- ❖ Trong python chỉ có một constructor
- ❖ Có thể sử dụng giá trị mặc định để tạo nhiều constructor
- ❖ `def __init__(self,ts=0,ms=1):`
- ❖ `self.tuso = ts`
- ❖ `self.mauso = ms`
- ❖ `a=phanso()`
- ❖ `b=phanso(2,3)`

```
class phanso:
 tuso = 0
 mauso = 1

 def __init__(self, ts=3, ms=4):
 self.tuso = ts
 self.mauso = ms
 def inra(self):
 return ("{} / {}".format(self.tuso, self.mauso))

a=phanso()
b=phanso(11,2)
print(a.inra())
print(b.inra())
```



# Hàm thuộc tính

- ❖ Trong python không có hàm thuộc tính, mà hàm thuộc tính được viết lại nếu cần