

# Chương 1

## Giới thiệu chung

Tại sao lại cần đến các quy ước chuẩn chung khi viết code?

Thực tế, 80% thời gian sử dụng của các phần mềm cần được bảo trì. Nhưng không phải lần bảo trì nào cũng được thực hiện bởi chính người đã viết ra phần mềm đó, hay có nghĩa là các lập trình viên sẽ phải thao tác trên mã nguồn của người khác. Do đó, cần có một quy chuẩn viết code chung để có thể tăng khả năng dễ đọc của code, cho phép người lập trình hiểu code nhanh và kĩ càng hơn. Ngoài ra, khi viết code theo quy ước chung cũng giống như mặc đồng phục cho sản phẩm của mình, thể hiện sự "gọn gàng, ngăn nắp", tăng độ tin cậy cho sản phẩm.

## Chương 2

# Các quy ước chuẩn khi code trong ngôn ngữ java

### 2.1 Cấu trúc của file mã nguồn java

Một file mã nguồn Java sẽ cấu trúc theo thứ tự như sau:

- Dòng comment giới thiệu ngắn gọn về phiên bản, bản quyền và chức năng của file mã nguồn.
- Khai báo package rồi import statement.
- Viết class/interface documentation comment (`/** ... */`)
- Khai báo các class và interface.
- Khai báo biến toàn cục, theo thứ tự từ public, protected đến private.
- Khai báo hàm khởi tạo.
- Khai báo các phương thức.

### 2.2 Khoảng cách thụt đầu dòng và bố cục dòng code

- Khoảng cách thụt đầu dòng  
Một lần thụt vào để biểu thị quan hệ giữa các lệnh có độ dài là 4 space, thường thì điều này đã được mặc định sẵn trong editor.
- Độ dài dòng code  
Không nên viết dòng code chứa quá 80 kí tự, do một số terminal không hỗ trợ các trường hợp đó.
- Xuống dòng khi đoạn code không phù hợp với một dòng code hiện tại
  - Vị trí ngắt sau dấu phẩy và trước toán tử.
  - Nên ngắt ở các toán hạng có thứ tự thực hiện thấp hơn.

- Khi ngắt dòng nên sắp xếp các tham số có cùng chức năng thẳng hàng nhau.
- Dưới đây là một số ví dụ:
 

```
function(longExpression1, longExpression2, longExpression3,
        longExpression4, longExpression5);
```

## 2.3 Comment

### 2.3.1 Block Comment

- Thường được dùng để mô tả file, thuật toán, cấu trúc dữ liệu, hàm.
- Thường được đặt trước hàm, file, hoặc nếu trong hàm thì được đặt cùng số dòng thụt vào với đoạn code nó mô tả.
- Đây là một ví dụ của block comment
 

```
/*
 * Đây là một ví dụ về block comment
 */
```

### 2.3.2 Single-Line Comment

- Nếu comment chỉ trên một dòng thì sẽ dùng Single-Line Comment, có cùng số dòng thụt vào với đoạn code.
- Sau đây là ví dụ của một Single-Line Comment

```
if (condition) {
    /* Write Comment in here */
}
```

### 2.3.3 Trailing Comment

- Là một đoạn comment gắn cùng dòng với code
- Trong một khối code thì các Trailing Comment sẽ được thụt ra sao cho thẳng hàng với nhau.

### 2.3.4 End-Of-Line Comment

- Trên cùng một dòng code, sau `//` sẽ là comment, chương trình dịch sẽ bỏ qua nó
- Thường được dùng để comment từng phần của dòng code.

### 2.3.5 Documentation Comment

- Doc-Comment được dùng để mô tả về class, interface, công cụ khởi tạo, phương thức và các trường. Mỗi giao diện chương trình sẽ có một doc-comment.
- Doc-comment thường được đặt trước khai báo, ví dụ:

```

/**
 * Ví dụ về doc-comment.
 */
class Example{
    ...
}

```

- Doc Comment có thể phân tách thành file HTML bằng các công cụ của java.

## 2.4 Declarations

- Nên sử dụng một khai báo trên một dòng code.
- Không khai báo khác kiểu trên cùng một dòng.
- Các biến địa phương nên được khai báo trên đầu khối code sử dụng chúng.

```

void MyMethod() {
    int int1;           // beginning of method block

    if (condition) {
        int int2;       // beginning of "if" block
        ...
    }
}

```

## 2.5 Các quy ước đặt tên

- Tên của class sẽ là danh từ, với các chữ cái đầu của từng từ được viết hoa.
- Cách đặt tên của interface giống với class.
- Tên phương thức là động từ, với chữ cái của từ đầu tiên được viết thường, còn chữ cái đầu tiên của các từ tiếp theo được viết hoa.
- Cách đặt tên biến cũng giống như phương thức, tránh các trường hợp tên biến quá dài hoặc chỉ có một ký tự, trừ các trường hợp là biến đếm tạm thời.
- Cách đặt tên các hằng số là toàn bộ bởi các chữ cái in hoa, các từ được nối với nhau bởi "\_".

## Chương 3

# Các quy ước chuẩn khi code trong ngôn ngữ Python

Về các quy tắc cơ bản thì trong Python cũng quy ước giống trong Java. Do Python 2 và Python 3 sẽ có các định dạng chuẩn khác nhau nên dưới đây chỉ nói những phần chung nhất của cả hai.

### 3.1 Code Lay-Out

#### 3.1.1 Indentation

- Sử dụng 4 khoảng trắng mỗi mức thụt vào đầu dòng.
- Các dòng nằm trong cùng khối code thì được xếp thẳng hàng.
- Khi ngắt dòng các dòng code thì các đoạn code hay các thành phần code có cùng chức năng sẽ được xếp thẳng hàng.
- Trong if-else statement, nếu điều kiện viết trong một dòng không đủ thì có thể ngắt dòng, thêm dấu ngoặc đơn vào điều kiện, dữ các tham chiếu thẳng hàng với nhau.
- Khi có nhiều dòng trong ngoặc đơn, ngoặc móc, thì dấu đóng ngoặc sẽ nằm ở dòng khác sau dòng cuối cùng của dãy code trong ngoặc. Ví dụ:

```
my_list = [  
    1, 2, 3,  
    4, 5, 6,  
]
```

#### 3.1.2 Độ dài dòng code

Một dòng code nói chung sẽ không có đến 80 kí tự trên một dòng, trong các block code thì độ dài không quá 72 kí tự.

### 3.1.3 Import

- Một dòng chỉ có một Import
- Import đứng ở đầu của file, chỉ sau comment hoặc docstring.

### 3.1.4 Naming Conventions

- Python không có cú pháp chặt chẽ cho các đối tượng private như trong java. Nhưng quy ước chung là, bắt đầu tên với kí tự "\_" nhằm chỉ thị rằng đây là thành phần private, không nên truy cập đến các thành phần này từ bên ngoài.
- Dùng kí tự "\_" ở cuối của tên để tránh sự trùng lặp với các tên, từ khoá,... đã sẵn có.
- Tên của Class sẽ được viết hoa chữ cái đầu tiên của các từ.  
VD: ClassName
- Tên của các Exception giống như cách đặt với tên của class, thường thêm hậu tố Error để phân biệt.
- Tên của hàm sẽ là các chữ cái viết thường, các từ được cách nhau bởi "\_", mục đích tăng khả năng đọc.
- Quy tắc đặt tên biến cũng giống như tên hàm.
- Tên các hằng số được viết hoa tất cả chữ cái, các từ được cách nhau bởi kí tự "\_"