**CODING CONVENTION**

**FOR JOBRECSYS PROJECT**

**MỤC LỤC**

[**1.** **INDENTATION (THỤT ĐẦU DÒNG)** 1](#_Toc455254312)

[**1.1.** **Độ dài một dòng** 1](#_Toc455254313)

[**1.2.** **Ngắt dòng** 1](#_Toc455254314)

[**2.** **COMMENTS (CHÚ THÍCH)** 3](#_Toc455254315)

[**2.1.** **Block comments** 3](#_Toc455254316)

[**2.2.** **Single-line comments** 3](#_Toc455254317)

[**2.3.** **Trailing comments** 3](#_Toc455254318)

[**2.4.** **End-of-line comments** 3](#_Toc455254319)

[**3.** **DECLARATIONS (KHAI BÁO)** 4](#_Toc455254320)

[3.1. **Số khai báo trên 1 dòng** 4](#_Toc455254321)

[3.2. **Vị trí khai báo** 4](#_Toc455254322)

[3.3. **Khởi tạo** 5](#_Toc455254323)

[3.4. **Khai báo class và interface** 5](#_Toc455254324)

[**4.** **STATEMENTS (CÂU LỆNH)** 5](#_Toc455254325)

[**4.1.** **Lệnh đơn giản** 5](#_Toc455254326)

[**4.2.** **Lệnh ghép** 6](#_Toc455254327)

[**4.3.** **if, if-else, if-else-if-else** 6](#_Toc455254328)

[**4.4.** **for** 6](#_Toc455254329)

[**4.5.** **while** 7](#_Toc455254330)

[**4.6.** **do-while** 7](#_Toc455254331)

[**4.7.** **switch** 7](#_Toc455254332)

[**4.8.** **try-catch** 7](#_Toc455254333)

[**5.** **WHITE SPACE (KHOẢNG TRẮNG)** 8](#_Toc455254334)

[**5.1.** **Dòng trắng** 8](#_Toc455254335)

[**5.2.** **Khoảng trắng** 8](#_Toc455254336)

[**6.** **NAMING CONVENTIONS (QUY TẮC ĐẶT TÊN)** 8](#_Toc455254337)

# **INDENTATION (THỤT ĐẦU DÒNG)**

## **Độ dài một dòng**

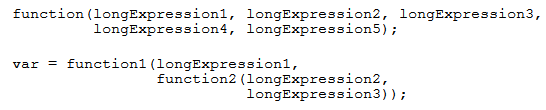
Độ dài 1 dòng không quá 80 ký tự

## **Ngắt dòng**

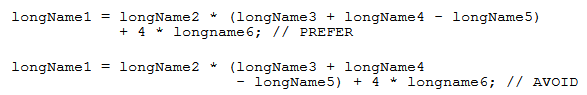
Khi ngắt dòng theo các nguyên tắc sau:

* Ngắt dòng sau dấu phẩy
* Ngắt dòng trước toán tử
* Căn lề dòng mới được ngắt bằng với ký tự đầu của biểu thức được ngắt
* Ví dụ:

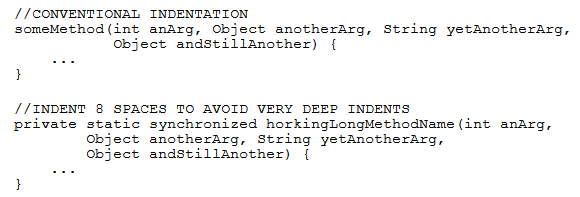
Cách thụt dòng khi ngắt dòng



Ngắt dòng không được ngắt giữa chừng biểu thức



Trường hợp đặc biệt, khi phần tên khai báo quá dài để tránh việc thụt dòng quá sâu, ta chỉ cần thụt vào 8 khoảng trắng.



# **COMMENTS (CHÚ THÍCH)**

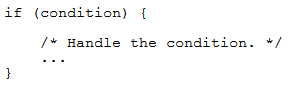
## **Block comments**

Block comment sử dụng để mô tả cho files, methods, data structures and algorithms. Block comment được đặt ở đầu mỗi file, method hoặc có thể bên trong method tuy nhiên đặt nó ngay trước phần cần chú thích (mô tả). Trước block comment là một dòng trống để tách biệt phần comment với đoạn code ở trên. Block comment nằm trong khối /\* … \*/

Ví dụ: 

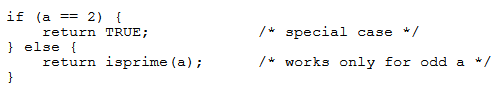
## **Single-line comments**

Tương tự với block comment tuy nhiên chỉ comment trên 1 dòng duy nhất.

Ví dụ: 

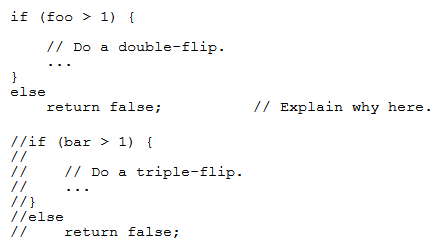
## **Trailing comments**

Loại comment rất ngắn gọn được viết ngay sau và trên cùng 1 dòng code để mô tả cho dòng code này. Nhưng phải thụt dòng 2 ký tự tab để tách bạch với phần code. Nếu có nhiều hơn 1 comment trong 1 khối code thì cần căn lề chúng bằng nhau.

Ví dụ: 

## **End-of-line comments**

Loại comment này bắt đầu với dấu // và comment từ ký tự comment đến hết dòng. Không nên sử dụng dấu // để comment cho một khối text trên nhiều dòng liên tục, tuy nhiên có thể dùng để comment cho một đoạn code trên nhiều dòng liên tục.

Ví dụ: 

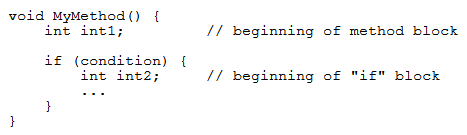
# **DECLARATIONS (KHAI BÁO)**

## **Số khai báo trên 1 dòng**

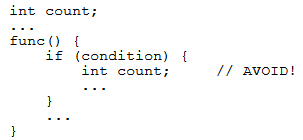
Mỗi khai báo nằm trên một dòng riêng biệt

## **Vị trí khai báo**

Chỉ khai báo tại đầu mỗi block (block là đoàn code nằm trong dấu ngoặc nhọn {}). Không được phép đợi đến khi dùng mới khai báo, nó có thể khiến dễ nhầm lẫn về phạm vi của khai báo.

Ví dụ: 

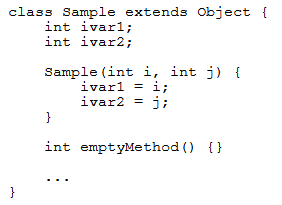
Trường hợp ngoại lệ duy nhất có thể khai báo ngay khi dùng là trong vòng lặp for: 

Tránh việc khai báo trùng tên ở bên trong 1 block, ví dụ: 

## **Khởi tạo**

Khởi tạo local variable (biến cục bộ) tại vị trí khai báo. Trường hợp duy nhất không khởi tạo tại vị trí khai báo là giá trị khởi tạo phụ thuộc vào một vài phép tính toán nào đó.

## **Khai báo class và interface**

* Không có khoảng trắng giữa tên class/interface với dấu ngoặc bắt đầu danh sách tham số.
* Dấu mở ngoặc nhọn “{” đặt ở cuối dòng của dòng lệnh khai báo
* Dấu đóng ngoặc nhọn “}” đặt ở một dòng riêng, căn lề bằng với dòng lệnh khai báo
* Khai báo phương thức được tách biệt bởi 1 dòng trắng
* Ví dụ: 

# **STATEMENTS (CÂU LỆNH)**

## **Lệnh đơn giản**

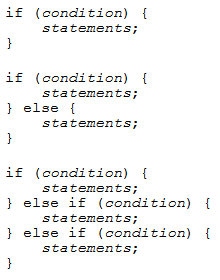
Mỗi dòng chứa tối đa một câu lệnh

## **Lệnh ghép**

* Lệnh ghép là câu lệnh mà chứa một danh sách các lệnh nằm trong cặp ngoặc nhọn “{statement}”.
* Đặt dấu ngoặc nhọn mở “{” ở cuối dòng của câu lệnh ghép
* Đặt dấu ngoặc nhọn đóng “}” ở một dòng mới, và căn lề ngang với câu lệnh ghép.
* Dấu ngoặc nhọn dùng để bao bọc một khối các lệnh hoặc thậm chí là một lệnh khi chúng là một phần của cấu trúc điểu khiển như if-else hoặc for.

## **if, if-else, if-else-if-else**

Cấu trúc if-else nên có cấu trúc như sau:



Lưu ý luôn sử dụng cặp dấu ngoặc nhọn với lệnh if-else. Tránh trường hợp sau: 

## **for**

Cấu trúc lệnh for như sau: 

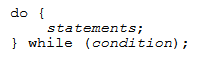
## **while**

Cấu trúc lệnh while như sau:



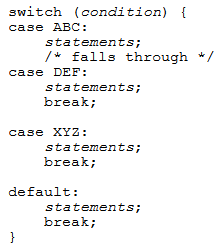
## **do-while**

Cấu trúc lệnh do-while như sau:



## **switch**

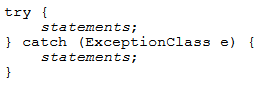
Cấu trúc lệnh switch như sau:



* Lưu ý: mỗi lệnh case mà không có break thì phải đưa vào một comment tại vị trí lệnh break bị bỏ qua để rõ ràng (ví dụ ở trên là /\*falls through\*/)
* Lệnh switch bắt buộc phải có default.

## **try-catch**

Lệnh try-catch có cấu trúc như sau:



# **WHITE SPACE (KHOẢNG TRẮNG)**

## **Dòng trắng**

Dòng trắng được thêm vào để code dễ đọc hơn.

Nên chèn một dòng trắng trong những trường hợp sau:

* Giữa các phương thức
* Giữa biến cục bộ trong phương thức và các lệnh của phương thức
* Trước một block comment hoặc single comment
* Giữa các phần logic khác nhau bên trong 1 phương thức

## **Khoảng trắng**

Khoảng trắng nên được sử dụng trong các trường hợp sau:

* Một từ khóa theo sau bởi 1 dấu ngoặc 
* Sau dấu phẩy trong một danh sách tham số
* Các biểu thức trong cấu trúc for 
* Sau các kiểu ép kiểu 

# **NAMING CONVENTIONS (QUY TẮC ĐẶT TÊN)**

Quy ước đặt tên phải ngắn gọn, rõ nghĩa

|  |  |  |
| --- | --- | --- |
| Kiểu áp dụng | Quy tắc | Ví dụ |
| Classes | Tên class là danh từ, viết hoa mỗi kí tự đầu mỗi từ | class Algorithm  class DataAnalytics |
| Interfaces | Tên interface tương tự với tên class nhưng thêm chữ cái “I” vào đầu tên | interface IAlgorithm  interface IDataPreparer |
| Methods | Tên phương thức là động từ, chữ cái đầu tiên viết thường, chữ cái đầu mỗi từ tiếp theo phải viết hoa. | run();  runHybrid();  getRecommendedResult(); |
| Variables | Hạn chế sử dụng các tên biến có một ký tự, ngoại trừ các biến tạm sẽ bỏ đi ngay sau đó. | int count;  String path;  float arrayLenght; |
| Constants | Tên hằng được viết in tất cả các ký tự, các từ cách nhau bằng dấu gạch dưới (\_) | int MAX\_WIDTH;  int RECOMMENDED\_ITEM; |