Coding Convention

1. Quy tắc đặt tên

3 cú pháp đặt tên có thể áp dụng và tuân theo:

* Cú pháp CamelCase (cú pháp lạc đà): Quy ước ký tự đầu tiên viết thường, những ký tự đầu tiên của chữ cái tiếp theo viết hoa.
* Cú pháp Pascal Case: Chữ cái đầu tiên viết hoa.
* Cú pháp Snake: Các chữ cái đều viết thường và các từ được phân tách bởi dấu gạch dưới.
* Quy tắc đặt tên:
* Tên biến, tên lớp thường là cụm danh từ, danh từ hoặc tính từ. Khi đặt tên phải có ý nghĩa, không đặt theo kiểu viết tắt, không đặt tên chung chung, tên tối nghĩa.
* Ngoài ra một số trường hợp, tên biến cần phải thể hiện rõ kiểu dữ liệu của biến đó. Ví dụ: biến có kiểu List thì nên đặt tên là movieList, biến có kiểu Set thì nên đặt là movieSet, biến có kiểu Map thì nên đặt tên là movieMap, biến có kiểu Array thì nên đặt tên là studentArray,…
* Tên biến, hàm sẽ được đặt theo cú pháp lạc đà. Ví dụ: movieName, description,…
* Tên lớp đặt theo cú pháp Pascal. Ví dụ: User, Movie,…
* Tên Class nên có thêm những từ có hậu tố phía sau để thể hiện rõ hơn mục đích của Class đó, chẳng hạn như DataController, SecurityConfig,…
* Tên Package tuân theo quy tắc chung ở trên và phải viết thường. Ví dụ: com.example.demo.controller, com.example.demo.model,…
* Các hằng số đặt theo UPPER\_CASE.
* Sử dụng biến: Tránh tình trạng khai báo biến mà không sử dụng.
* Import thư viện sử dụng: Chỉ import thư viện sử dụng cần thiết. Không sử dụng import tất cả.

1. Quy tắt số lượng

* Các hàm không được quá 30 dòng.
* Các lớp không vượt quá 500 dòng.
* Mỗi hàm không được vượt quá 5 tham số.
* Hàm chỉ thực hiện một việc duy nhất.
* 1 Dòng chỉ chứa 1 biến.
* Mỗi dòng không được quá 80 ký tự và các câu lệnh được lồng vào nhau không quá 4 cấp.

1. Quy tắc xuống hàng

* Nếu một hàm có nhiều cấp lồng nhau, mỗi cấp nên xuống dòng.
* Các đoạn code bằng cấp nên ở cùng một cột với nhau, dòng xuống hàng nên bắt đầu cùng cấp với dòng phía trên.
* Xuống hàng trước các toán tử (ví dụ như +, -, \*, ?,…).

Ví dụ:

**public** SecurityFilterChain securityFilterChain(HttpSecurity http) **throws** Exception {

http

.authorizeHttpRequests((requests) -> requests

.antMatchers("/login","/register","/home", "/","/images/\*\*","/saveData","/css/\*\*").permitAll()

.anyRequest().authenticated()

)

.formLogin((form) -> form

.loginPage("/login")

.permitAll()

.defaultSuccessUrl("/login?success=true")

.failureUrl("/login?success=false")

)

.logout((logout) -> logout.permitAll());

**return** http.build();

}

1. Quy tắc comment

* Hạn chế dùng comment để giải thích code hoặc chú thích những sự thật hiển nhiên.
* Không comment khi đóng thẻ/ngoặc:
* Xoá code không dùng, đừng comment out đoạn code đó.
* Comment cảnh báo hậu quả.
* Comment làm rõ ý nghĩa của code (nếu logic phức tạp).

1. Quy tắc WhiteSpace
   1. White Space – thụt đầu dòng

* Xác định một đơn vị thụt đầu dòng cho toàn bộ mã nguồn của chương trình.
* 1 đơn vị thụt đầu dòng = 1 tab(\*).
* Hoặc, 1 đơn vị thụt đầu dòng = 5 khoảng trắng.
* Nên dùng tab thay cho khoảng trắng.
* Đỡ tốn công nhập quá nhiều lần khoảng trắng.
* Có thể tùy chỉnh một đơn vị tab ứng với bao nhiêu khoảng trắng tùy ý.
* Hai dòng code cách nhau một bậc thì sẽ cách nhau một đơn vị thụt đầu dòng.
  1. White Space – Dòng trống
* Những dòng code có quan hệ với nhau (cùng thực hiện một công việc) thì gom lại thành một block.
* Nghĩa là không có dòng trống giữa các đoạn code như trên.
* Hai block code thì cách nhau ít nhất một dòng trống.
* Đặt khoảng trắng sau dấu phẩy và dấu chấm phẩy.
* Đặt khoảng trắng xung quanh các toán tử.

1. Quy tắc ngoặc nhọn

Theo tiêu chuẩn Java: dấu “{” phải được đặt cùng dòng với các câu if, for, while,…