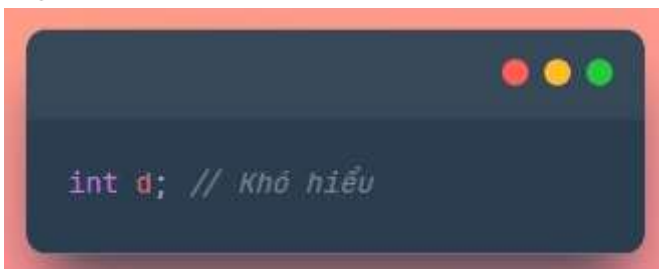


🕒 Cập nhật tháng 8 năm 2024

[Bài đọc] Các kỹ thuật Clean code

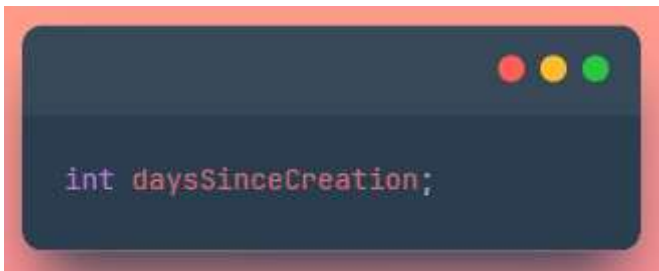
1. Đặt Tên Biến, Hàm, Lớp Có Ý Nghĩa

- Tên biến, hàm, lớp phải mô tả đúng mục đích và tránh viết tắt khó hiểu
- Không nên:



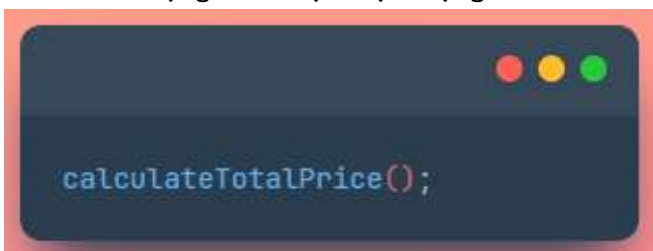
```
int d; // Khó hiểu
```

- Nên:



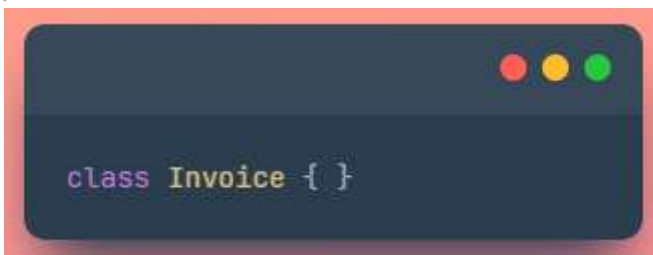
```
int daysSinceCreation;
```

- Hàm nên là động từ hoặc cụm động từ:



```
calculateTotalPrice();
```

- Lớp nên là danh từ:



```
class Invoice { }
```

2. Hàm Ngắn Gọn Và Làm Một Việc

- Một hàm nên chỉ thực hiện một nhiệm vụ duy nhất
- Độ dài lý tưởng: 5-20 dòng
- Nếu một hàm thực hiện nhiều bước, hãy tách chúng thành các hàm con rõ ràng
- Ví dụ xấu:

```
void processOrder() {  
    validateOrder();  
    saveOrder();  
    sendConfirmationEmail();  
    generateInvoice();  
}
```

- Cải thiện:

```
void processOrder() {  
    validateOrder();  
    persistOrder();  
    notifyCustomer();  
    createInvoice();  
}
```

3. Tránh Magic Number (Số Ma)

- Magic number là những con số xuất hiện trong code mà **không rõ ý nghĩa**
- Không nên:

```
if (salary > 5000) { ... }
```

- Nên:

```
final int HIGH_SALARY_THRESHOLD = 5000;  
if (salary > HIGH_SALARY_THRESHOLD) { ... }
```

4. Tách Code Theo Nguyên Tắc SRP (Single Responsibility Principle)

- Một class chỉ nên có một lý do để thay đổi
- Ví dụ:
 - Class **InvoicePrinter** chỉ nên lo việc in hóa đơn
 - Class **InvoiceCalculator** chỉ tính toán

5. Tránh Nested Code (Lồng Quá Nhiều)

- Tránh if-else lồng nhau quá sâu bằng cách **return sớm** hoặc sử dụng **guard clause**

- Ví dụ:

```
if (order != null) {  
    if (order.isValid()) {  
        process(order);  
    }  
}
```

- Cải thiện:

```
if (order == null || !order.isValid()) return;  
process(order);
```

6. DRY - Don't Repeat Yourself (Đừng Lặp Lại)

- Tránh viết lặp code bằng cách trích xuất thành hàm chung hoặc class chung

7. Comment Đúng Chỗ, Đúng Mục Đích

- Chỉ comment khi cần giải thích ý đồ hoặc quy tắc nghiệp vụ phức tạp
- Code tự nó đã phải rõ ràng
- Comment không được thay thế cho việc đặt tên đúng

Tài nguyên đọc thêm: <https://www.geeksforgeeks.org/blogs/tips-to-write-clean-and-better-code>

Danh sách các bài học

