Quy tắc viết code – Coding Conventions

Ver. 19.01

Tên app/project

- **Quy ước:** Danh từ, chữ hoa từng đầu từ, trừ tình huống đặc biệt về nhận dạng thương hiệu (iCloud). Không dùng khoảng trắng, dấu tiếng Việt, và các kí tự đặc biệt
- **Ví dụ:** `StudentManagement, HelloWorld, PetCareSystem, FacebookMessenger`

Tên package/thư mục chứa code

- **Quy ước:** Danh từ, chữ thường, cố gắng từ đơn, các thư mục con phân cách nhau bởi dấu chấm. Không dùng khoảng trắng, dấu tiếng Việt, và các kí tự đặc biệt. Theo lẽ thường, package gốc dùng phần đuôi của tên miền của tổ chức, ví dụ com, edu. gov. Thư mục/package mức con tùy thuộc vào quy tắc tổ chức các thành phần code của dự án
- **Ví dụ:** `com.apple.quicktime, util, data, edu.fpt.util, bmag.data`

Tên class/sự phân nhóm đối tượng – Tên interface

- **Quy ước:** Danh từ, chữ hoa từng đầu từ. Viết cả từ, tránh viết tắt
- **Ví dụ:** `Dog, Cat, File, String, Student, Person, FileInputStream, StringTokenizer, HinhTron, TamGiac`

Tên biến/tên vùng nhớ RAM, chứa data

- **Quy ước biến thường:** Danh từ, chữ hoa từng đầu từ, từ đầu tiên viết chữ thường (cú pháp con lạc đà, camel case notation). Nên đặt tên biến có ý nghĩa/theo mục đích sử dụng, tránh viết tắt trừ những biến tạm thời (biến trung gian t, tmp, biến đếm i, j, k)
- **Ví dụ:** `luongCoBan, dienTich, chuVi, basicSalary, yearOfBirth, yob, salary`
- **Quy ước hằng số (const, final):** Danh từ, chữ hoa tất cả các từ, gạch dưới _ dùng để kết nối các từ

- **Ví dụ:** `THUE_GIA_TRI_GIA_TANG, VAT, VALUE_ADDED_TAX, MAX_SPEED, MAX_ELEMENTS, PI`

Tên function/method/hàm, diễn tả hành động, xử lí

- **Quy ước:** Verb + Object, động từ kèm theo bổ ngữ, chữ hoa từng đầu từ, từ đầu tiên viết chữ thường (cú pháp con lạc đà, camel case notation)
- **Ví dụ:** `tinhLuong(), computeArea(), getSalary(), showInfo(), sort(), sortByName()`
- **Khuyến khích:** có ghi chú cho tên hàm (ý nghĩa của hàm, đầu vào, đầu ra…)

Indentation/gióng lề cho code

- **Quy ước:** Mặc định các đoạn code mức con sẽ thụt vào so với mức cha 4 khoảng trắng (tương đương 1 phím tab)
- **Ví dụ:**
```
if ((gpa >= 9) && (gpa <= 10))
        System.out.println("You are excellence");
```
- **Quy ước:** Lưu ý dấu cách trong các thành phần của biểu thức, phép gán, câu lệnh
- **Ví dụ:**
```
int a=10; //sai chuẩn
int a = 10; //đúng chuẩn
```

Kiểm thử nhập liệu - Tái sử dụng - Menu (Validation – Re-use - Interaction)

- **Tư duy tách hàm và re-use:** tránh lặp lại các khối lệnh có cùng mục đích sử dụng, thay vào đó viết một hàm để dùng lại trong các ngữ cảnh khác nhau
- **Ví dụ:** viết một hàm int `inputAnInteger(int lower, int upper)` dùng nhập một con số nguyên trong khoảng từ `lower` đến `upper`. Hàm này sẽ dùng lại ở bất cứ chỗ nào yêu cầu nhập một con số nguyên trong khoảng. Mọi việc nhập *"cà chớn"* hàm này *"cân"* hết sẵn rồi
- **Chặn dữ liệu nhập:** Mọi giá trị nhập vào từ bàn phím phải được chặn các lỗi không mong muốn do vô tình hay cố ý. Ví dụ yêu cầu nhập số nguyên, người dùng gõ *1a, 3.14, ahihi* đều bị *"chửi"* yêu cầu nhập lại.
- **Tránh mọi sự sụp đổ (crash) của chương trình:** kiểm tra kĩ các ngoại lệ (exception), tràn biên, sai định dạng, lỗi tập tin, vòng lặp vô tận, null pointer…

- **Tránh hard-code:** Các giá trị, con số có ý nghĩa riêng trong code, phải được định nghĩa thành hằng số
- Chương trình cho cơ chế hỏi đáp người dùng qua lựa chọn thực đơn, hay hỏi có muốn làm tiếp hay không?

Function Prototype và cấu trúc chương trình (trong C)

- **Các hàm phải viết prototype:** là dòng tiêu đề khai báo hàm, đặt dưới lệnh #include và #define
- **Cấu trúc file main.c** dạng như dưới đây:

```
#include statements
#define statements
type definitions
prototypes
the main function
the body of the functions
```

```c
#include <stdio.h>
#include <stdlib.h>
#define PI 3.14
const float VAT = 0.1;

void sayHello(char* message);  //prototype

int main()
{
    sayHello("Welcome to the C World!");  //invoke the function
    return 0;
}

//body of the function
void sayHello(char* message)
{
    printf("%s\n", message);
}
```