

CODING CONVENTION

Tham khảo: [Google C++ Style Guide](#)

I – Quy tắc đặt tên

1. Đặt tên hàm:

Đặt tên hàm bằng tiếng Anh theo dạng lowerCamelCase, đứng đầu là động từ. Ví dụ:

- Hàm kiểm tra số nguyên tố: `isPrime(int x)`
- Hàm đếm số nguyên tố: `countPrimes(int a[])`

2. Đặt tên biến:

Đặt tên biến bằng tiếng Anh theo dạng under_score. Tên biến là một danh từ có ý nghĩa, hạn chế các biến có ý nghĩa chung chung như a, b, c, ... Riêng biến chạy trong vòng lặp có thể đặt là i, j, k, ... Ví dụ:

- Tên biến đếm số lượng phần tử phân biệt: `num_distinct / num_unique / ...`

3. Đặt tên lớp:

Đặt tên lớp bằng tiếng Anh theo dạng UpperCamelCase. Tên lớp là một danh từ có ý nghĩa. Ví dụ:

Tên lớp tài khoản tiết kiệm: `SavingAccount`

4. Đặt tên cho các thuộc tính trong lớp:

Đặt tên cho các thuộc tính trong lớp tương tự như các biến, tuy nhiên thêm dấu _ ở cuối. Ví dụ:

Lớp tài khoản tiết kiệm:

```
class SavingAccount {  
    private:  
        string    user_id_;  
        double    balance_;  
};
```

5. Đặt tên hằng số:

Đặt tên hằng số bằng tiếng Anh theo dạng UPPER_CASE. Tên hằng số là một danh từ có ý nghĩa. Ví dụ:

- Hằng số pi: `PI`
- Hằng số xác định lỗi chia 0: `ERROR_DIVIDE_0`

6. Đặt tên header.

Đặt tên header bằng tiếng Anh theo dạng UPPER_CASE_H_ (kết thúc là _H_) và comment ở phần `#endif`. Ví dụ:

```
#ifndef IT_FACULTY_H_
#define IT_FACULTY_H_
...
#endif    // IT_FACULTY_H_
```

II – Sử dụng khoảng trắng

Khoảng trắng được sử dụng:

- Giữa hai toán tử của một phép toán. Ví dụ: `a + b` , `a ^ b` , `a | b` , `a >>= b`
- Đằng sau các từ khóa như `if`, `while`, `switch`.... Ví dụ:

```
if (a == b)
{
    ...
}
```

```
for (int i = 0; i < 10; i++)
{
    ...
}
```

- Đối với con trỏ, ta khai báo với định dạng như sau: `int *a`
- Đối với tham chiếu, ta khai báo với định dạng: `vector<int>& my_vector`

III – Sử dụng inline function

Chỉ sử dụng inline function với những hàm nhỏ, khoảng từ 10 dòng code đổ xuống.

IV – Comment

Comment được viết bằng tiếng Việt có dấu và được đặt ở đầu hàm thể hiện ý tưởng cài đặt và nhiệm vụ của hàm đó. Ví dụ:

```
bool isPrime(int x)
{
    /* Hàm kiểm tra một số x có phải là số nguyên tố hay không.
    TH đặc biệt: x = 2.
    Các TH còn lại: Duyệt qua tất cả các số nguyên từ 2 đến căn x để tìm
    ước số khác 1 của x. Nếu tồn tại nghĩa là x không phải số nguyên tố.
    */
    ...
}
```

V – Một số lưu ý khác

- Sử dụng `cin`, `cout` để nhập xuất.
- Ký tự kết thúc sử dụng `endl` (không sử dụng `'\n'`).
- Sử dụng `string` cho dữ liệu kiểu chuỗi.
- Sử dụng toán tử 3 ngôi (ternary operator) đối với những phép toán hay phép in ra màn hình yêu cầu câu lệnh `if... else...` đơn giản. Ví dụ:
 - Nếu số `x` là nguyên tố thì in “yes”, ngược lại in “no”. Ta viết câu lệnh in ra màn hình như sau: `cout << (isPrime(x) ? "yes" : "no");`
 - Nếu `x` là số chẵn thì ta cộng thêm 3, số lẻ thì cộng 1 vào biến kết quả. Ta có câu lệnh: `result += (x % 2 == 0 ? 3 : 1);`