**Phong cách lập trình và chuẩn viết mã**

1. **Lập trình hướng đối tượng**

Nhóm đã chọn ngôn ngữ c# trên nền tảng .Net để viết mã, và ngôn ngữ này cũng hỗ trợ lập trình hướng đối tượng. Với hướng đối tượng, người dùng có thể dễ dàng quản lý code khi có sự thay đổi chương trình. Thay vì việc thay đổi các mã nguồn của đối tượng mà giờ đây lập trình viên chỉ cần thay đổi một số hàm thành phần. Ngoài ra lập trình hướng đối tượng còn có tính bảo mật cao và có khả năng tái sử dụng. Do đó sử dụng phương pháp lập trình hướng đối tượng giúp tăng năng suất và dễ bảo trì mã.

1. **Chuẩn viết mã**
   1. **Cách đặt tên và khai báo**

* Có 3 kiểu đặt tên thông dụng nhất:

+ Pascal Case: Chữ cái đầu tiên trong từ định danh và chữ cái đầu tiên của mỗi từ nối theo sau phải được viết hoa. Sử dụng Pascal Case để đặt tên cho một tên có từ 3 ký tự trở lên.

Vd: CodingConvention

+ Camel Case: Chữ cái đầu tiên trong từ định danh là chữ thường và chữ cái đầu tiên của mối từ nối theo sau phải được viết hoa.

Vd: codingConvention

+ Uppercase: Tất cả các ký tự trong từ định danh phải được viết hoa. Sử dụng quy tắc này đối với tên định danh có từ 2 ký tự trở xuống.

Vd: System.IO

* Tóm tắt cách sử dụng quy tắc:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Loại*** | ***Kiểu đặt tên*** | ***Ví dụ*** | ***Ghi chú*** |
| Tên class, enum | Pascal Case | CreateUser | Danh từ |
| Thuộc tính | Pascal Case | BackgroundColor | Danh từ |
| Phương thức | Pascal Case | GetValue() | Động từ |
| Sự kiện | Pascal Case | SelectedIndexChanged | Có hậu tố EventHandle |
| Giao diện (Interface) | Pascal Case | IButtonControl | Có tiền tố I |
| Tên biến | Carmel Case | firstName | Danh từ |
| Tham số | Carmel Case | displayTime | Danh từ |
| Hằng số | Uppercase | MAX\_VALUE | Có gạch chân giữa các từ |

* Tránh thêm tiền tố, hậu tố dư thừa.

Vd: ~~enum BorderEnum {…}~~ -> enum Border {…}

* Không thêm tên lớp chứa vào tên thuộc tính.

Vd: ~~Animal.WeightAnimal~~ -> Animal.Weight

* Tên biến, phương thức bool phải thể hiện được ý nghĩa nếu trả về true hoặc false. Nên sử dụng tiền tố “Is” “Can” “Has” trước tên biến, phương thức.

Vd: ~~bool checked = true;~~ -> bool isChecked = true;

* Không dùng các tên giống nhau.

Vd: int yourAge

~~int YourAge~~

* Không tạo 2 namespace cùng tên và chỉ khác nhau ở kiểu viết hoa/thường.

Vd: Namespace SunAsterisk

~~Namespace sunAsterisk~~

* Không nên xây dựng 1 phương thức với các tham số có cùng tên và chỉ khác nhau kiểu chữ:

Vd: ~~void MyFunction(string a, string A)~~

* Không xây dựng 1 kiểu với các tên property giống nhau và chỉ phân biệt ở kiểu chữ.

Vd: int Color {get, set}

~~int COLOR {get, set}~~

* Không đặt tên các phương thức có cùng tên và chỉ khác nhau ở kiểu chữ.

Vd: void Total ()

~~void total ()~~

* 1. **Câu lệnh**

Các câu lệnh khác nhau phải nằm trên các dòng khác nhau. Một dòng không nên nhiều quá 150 ký tự.

Đối với câu điều kiện thì phần điều kiện nên ở chung một dòng với if, nếu nhiều điều kiện quá có thể xuống dòng. Phần code thực hiện câu điều kiện nằm ở giữa 2 dấu “**{**“, “**}**”. Mỗi dấu “**{**“, “**}**” nằm ở 2 dòng riêng. Các câu điều kiện liên tiếp nhau cách nhau 1 dòng trống.

Text

Description automatically generated

* 1. **Hàm và tham số**

Mỗi hàm được phải được đặt tên bằng tiếng Anh và thể hiện được rõ chức năng của hàm và các từ trong hàm được cách nhau bằng “ \_ “.

Vd : int run\_sum() , không được viết thành runsum.

Không được đặt số và kí tự đặc biệt vào đầu tên của mỗi hàm

Vd : int 1run\_sum() ,int $#rum\_sum()

Không được đặt trùng tên với các phương thức có sẵn

Vd: char sort(), int sum() ,int max()

Mỗi hàm chỉ thực hiện một chức năng duy nhất.

* + Tham số của hàm

Tên của tham số phải thể hiện được ý nghĩa của nó, đặt tên phải liên mạch,không có khoảng trắng và các kí tự đặc biệt.

Vd: double phan\_so (int tuso, int mauso)

Không được đặt trùng tên với các phương thức có sẵn của ngôn ngữ

Khai báo tham sô phải ngắn gọn, thể hiện được mục đích của người viết.

Tham số phải khai báo ngang cung hàm với hàm, không được khai báo khác hàng.

Vd : int sum(int a, int b, int c)

không được viết thành

int sum(int a,

int b,

int c)

Khi thực hiện vòng for hoặc if else thì các phương thức trong nó phải TAB vào 1 lần .

Vd : for (…)

{

# Phương thức 1

If ()

{

# Phương thức 2

}

}

* 1. **Lớp**

Tên lớp là một danh từ hay cụm danh từ, được viết hoa chữ cái đầu tiên của mỗi từ. Vd: Flower, Monster, BieuMau, …

Hàm trong lớp làm tương tự các hàm khác.

Các thuộc tính của lớp được khai báo ở phần đầu của lớp. Cách đặt tên, cách khai báo tương tự các biến khác.

Text

Description automatically generated