

Quy ước chuẩn code với ngôn ngữ CSharp

Ngày 14 tháng 5 năm 2019

Mục lục

1	Quy ước đặt tên	1
1.1	Quy ước chung	1
1.2	Quy tắc đặt tên và cú pháp	2
2	Coding Style	2
2.1	Formating	2
2.2	Code Commenting	3

1 Quy ước đặt tên

Trong C#, có 3 kiểu đặt tên là thông dụng:

1. camelCase: Tên với chữ cái đầu tiên của từ được viết thường, các từ còn lại được viết hoa.

```
1      int exampleValue;  
2
```

2. PascalCase: Tên với chữ cái đầu tiên được viết hoa, các chữ cái đầu tiên của các từ tiếp theo cũng được viết hoa, còn lại viết thường.

```
1      void GetValue(){  
2          ...  
3      }  
4
```

3. _UnderScore: Xuất hiện với tiền tố "_" kết hợp với kiểu tên Camel hoặc Pascal.

```
1      private int _indentity;  
2
```

1.1 Quy ước chung

1. Luôn dùng CamelCase hoặc PascalCase để đặt tên.
2. Không đặt tất cả các chữ cái của tên viết hoa. Trong trường hợp có duy nhất một từ hoặc một ký tự thì có thể chấp nhận.
3. Không tạo các khai báo cùng kiểu(namespace, class, property, method, field, parameter) và các quyền truy cập(public, private, protected, internal) mà chỉ cần viết hoa các từ là có thể thay đổi.
4. Không bắt đầu các tên với ký tự số.
5. Không sử dụng các hậu tố là các số để định danh.
6. Chọn tên có nghĩa và cụ thể.
7. Biến và thuộc tính nên mô tả về thực thể.
8. Không dùng ký hiệu Hungarian.

9. Không đặt tên quá 5 từ.
10. Không sử dụng các từ vô nghĩa hoặc không cần thiết để đặt tên.
11. Trong tên của thuộc tính thì không sử dụng tên của class.
12. Cố gắng sử dụng tiền tố "Can", "Is", "Has" với các biến boolean.

1.2 Quy tắc đặt tên và cú pháp

Đối tượng	Quy tắc đặt tên
Project File	Đặt theo quy tắc Pascal Phù hợp với tên Namespace gốc
Source File	Đặt tên theo quy tắc Pascal Không bao gồm nhiều hơn 1 Class, Enum (globe) mỗi file.
Namespace	Đặt tên theo quy tắc Pascal Phù hợp với tên Project
Class và Struct	Đặt tên theo quy tắc Pascal Sử dụng danh từ hoặc cụm danh từ để đặt tên.
Interface	Đặt tên theo quy tắc Pascal Luôn sử dụng ký tự "I" trước tên Interface
Phương thức	Đặt tên theo quy tắc Pascal Sử dụng động từ hoặc động-trạng từ để đặt tên
Đặc tính	Đặt tên theo quy tắc Pascal Không sử dụng tiền tố set hoặc get trước tên đặc tính
Trường Private	Đặt tên theo quy tắc Camel Sử dụng ký tự "_" là tiền tố của tên
Trường non-private	Đặt tên theo quy tắc Pascal Không nên sử dụng trường non-private, nên dùng đặc tính thay thế
Hằng và trường tĩnh	Đặt tên giống như trường với các quyền truy cập tương ứng
Biến	Đặt tên theo quy tắc Camel Không nên đặt tên biến chỉ với đơn ký tự trừ trường hợp vòng for Không nên đặt tên liệt kê theo kiểu text1, text2, text3, ...
Tham số	Đặt tên theo quy tắc Camel

2 Coding Style

2.1 Formating

1. Không khai báo quá 1 namespace trong 1 file.
2. Không nên đặt nhiều class trong một file đơn lẻ.
3. Nên để các ngoặc móc (và) ở một dòng riêng.
4. Với các khối lệnh của câu lệnh điều kiện thì nên cho vào trong ngoặc móc.
5. Tab và thụt dòng mỗi mức là 4 đơn vị.
6. Khai báo các biến nên đọc lập, không nên cùng một câu lệnh.
7. Sử dụng khai báo **using namespace** ở đầu file, ưu tiên các **namespace** của ngôn ngữ hỗ trợ trước của người dùng.
8. Nội bộ bên trong class sẽ có bố cục như sau:
 - Các biến thành viên.
 - Các công cụ khởi tạo và công cụ kết thúc.
 - Các class, struct, enum.
 - Đặc tính

- Phương thức.
9. Quyền truy cập được khai báo theo thứ tự ưu tiên sau:
- Public
 - Protected
 - Internal
 - Private

2.2 Code Commenting

1. Mọi comment phải được viết cùng bởi một ngôn ngữ.
2. Nên sử dụng `//` hoặc `///`.
3. Sử dụng comment trong dòng code để giải thích giả thiết, thuật toán.
4. Không sử dụng comment trong code để giải thích rõ ràng code, code tốt thì tự bản thân nó đã thể hiện rõ ràng.
5. Nên sử dụng các tag-keyword để có thể lọc comment.

```
1 // TODO: Place DataBase Code Here!
2 // ERRFORMAT: Tackle in case wrong format input.
3
```

6. Luôn sử dụng CSharp comment-block (`///`) cho các khai báo `public`, `protected` và `internal`.
7. Luôn bao gồm các tag `<summary>` comments và các tag `<param>`, `<return>`, `exception` ở các vị trí phù hợp.