**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

🙤🙧🟍🙥🙦

A picture containing text

Description automatically generated

**PHONG CÁCH LẬP TRÌNH**

**CHUẨN VIẾT MÃ**

|  |  |  |
| --- | --- | --- |
| Sinh viên thực hiện: | | |
| STT | Họ tên | MSSV |
| 1 | Đỗ Nguyễn Hoàng Huy | 19521603 |
| 2 | Lê Dương Khánh Việt | 19522515 |
| 3 | Hoàng Xuân Vũ | 19522531 |
| 4 | Lê Văn Nhân | 19521935 |

**TP. HỒ CHÍ MINH – 6/2021**

Contents

[**1.** **Giới thiệu** 2](#_Toc76233314)

[**1.1.** **Tại sao cần phải có quy ước lập trình** 2](#_Toc76233315)

[**1.2.** **Thành lập quy ước** 2](#_Toc76233316)

[**1.3.** **Phạm vi** 2](#_Toc76233317)

[**2.** **Tên file** 2](#_Toc76233318)

[**2.1.** **Hậu tố** 2](#_Toc76233319)

[**2.2.** **Các file khác** 2](#_Toc76233320)

[**3.** **Phong cách lập trình** 3](#_Toc76233321)

[**3.1.** **Quy ước lập trình chung** 3](#_Toc76233322)

[**3.2.** **Quy ước đặt tên** 3](#_Toc76233323)

[**3.3.** **Quy ước viết mã** 3](#_Toc76233324)

[**3.4.** **Quy ước câu lệnh** 4](#_Toc76233325)

[**3.4.1.** **Câu lệnh cơ bản** 4](#_Toc76233326)

[**3.4.2.** **Câu lệnh ghép** 4](#_Toc76233327)

[**3.4.3.** **Câu lệnh return** 5](#_Toc76233328)

[**3.4.4.** **Câu lệnh if – else** 5](#_Toc76233329)

[**3.4.5.** **Câu lệnh for** 5](#_Toc76233330)

[**3.4.6.** **Câu lệnh foreach** 5](#_Toc76233331)

[**3.4.7.** **Câu lệnh while/do-while** 6](#_Toc76233332)

[**3.4.8.** **Câu lệnh switch-case** 6](#_Toc76233333)

[**3.4.9.** **Câu lệnh try-catch-finally** 7](#_Toc76233334)

[**3.5.** **Quy ước khoảng trắng** 7](#_Toc76233335)

[**3.5.1.** **Dòng trống** 7](#_Toc76233336)

[**3.5.2.** **Khoảng trống** 8](#_Toc76233337)

[**4.** **Keyword** 8](#_Toc76233338)

[**5.** **Tham khỏa** 8](#_Toc76233338)

1. **Giới thiệu**
   1. **Tại sao cần phải có quy ước lập trình**

Việc quy ước lập trình rất quan trọng bởi vì nhiều lý do có thể kể đến như:

* Phần lớn phần mềm được bảo hành và cập nhật trong suốt quảng đời duy trì hoạt động của phần mềm đó bởi tác giả.
* Các quy ước giúp việc lập trình trở nên thống nhất, dễ nhìn, cãi thiện khả năng đọc một cách nhanh chóng và kĩ lưỡng.
* Dễ bảo trì và phát triển ứng dụng sẽ đơn giản hơn nhiều (tránh trường hợp xem code của người khác không hiểu )
  1. **Thành lập quy ước**

Tài liệu này được tạo ra dựa trên các nguyên tắc, phản ánh các tiêu chuẩn viết mã do nhóm chúng em tham khảo từ các nguồn khác nhau và lập ra, được cả nhóm tuân theo trong quá trình phát triển phần mêm.

* 1. **Phạm vi**

Tài liệu này chỉ áp dụng cho ngôn ngữ lập trình C# và thư viện .NET Framewrok.

1. **Tên file**
   1. **Hậu tố**

Phần mềm C# sử dụng các hậu tố sau:

|  |  |
| --- | --- |
| **Loại tập tin** | **Hậu tố** |
| C# source | .cs |
| Developer Files | .resx |
| Database | .sql |

* 1. **Các file khác**

Thường sử dụng các file khác, bao gồm:

|  |  |
| --- | --- |
| Tên file | Mục đích sử dụng |
| README | Tóm tắt nội dung đồ án |
| .gitignore | Loại bỏ các file không cần thiết phải cập nhật trên github |

1. **Phong cách lập trình**
   1. **Quy ước lập trình chung**

|  |  |
| --- | --- |
| **Loại thuộc tính** | **Quy ước** |
| Source File | Một namespace/file và một class/file. |
| Dấu ngoặc nhọn | Trên dòng mới. |
| Comments | Chỉ sử dụng **//** và **///** để comment trên một dòng, không sử dụng (/\*…\*/) để comment trên nhiều dòng hoặc tạo flowerbox. |
| Biến | Mỗi biến khai báo một lần. |

* 1. **Quy ước đặt tên**

Bao gồm các quy tắt sau:

* Việc đặt tên phải có ý nghĩa và cụ thể, tránh đặt lộn xộn, không mô tả được ý nghĩa của tên được dặt.
* Tên file chứa source code phải viết hoa chữ cái đầu của mỗi từ.
* Tên biến phải viết thường chữ cái đầu tiên và viết hoa các chữ cái đầu của từ tiếp theo.
  + VD:
    - idBook
    - staffName
    - dobReader
* Tên hàm phải viết hoa chữ cái đầu tiên của từ trong câu.
* Tránh việc đặt tên biến/ hàm quá 5 từ.
* Tránh việc đặt tên biến/ hàm trùng với từ khóa, thư viện sẳn có trong C#.
* Chỉ sử dụng ký hiệu Hungarian trong việc đặt tên các tool C#.
  + VD:
    - Combo box = cb
    - Datagridview = dgv
    - Date Time Picker = dtp
  1. **Quy ước viết mã**

**Đoạn mã:**

* Không được sử dụng nhiều hơn 1 namespace trên 1 file.
* Tránh thêm quá nhiều class trong 1 file.
* Luôn sử dụng dấu ngoặc nhọn ({và}) trên một dòng mới.
* Luôn sử dụng dấu ngoặc nhọn ({và}) trong câu điều kiện.
* Khoảng cách thụt lề là 1 Tab hoặc 4 khoảng trắng.
* Đóng gói các phương thức (Methods), các biến cố (Events) trong từng khu vực(Region) riêng trong mỗi file.

**Comment:**

* Comment được sử dụng để giải thích các giả định, các vấn đề đã biết và thông tin chi tiết các thuật toán phức tạp
* Chỉ sử dụng **//** và **///** để comment trên một dòng.
* Không sử dụng (/\*…\*/) để comment trên nhiều dòng hoặc tạo flowerbox.
  + VD:

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// Flowerbox**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

* Comment phải được để trên hàm, phương thức hoặc thuật toán để giải tích về các hàm, phương thức hoặc thuật toán đó.
* Không sử dụng comment để giải thích những đoạn mã cơ bản, rõ ràng.
* Chỉ được comment 1 lần/ 1 đoạn code.
  1. **Quy ước câu lệnh**
     1. **Câu lệnh cơ bản**

Mỗi dòng là một câu lệnh khác nhau, không được gộp chung nhiều câu lệnh trên một dòng.

* VD:
  + Đúng:
    - private int idBillBorrow;
    - private DateTime Borrowdate;
    - private int idReader;
  + Sai:
    - private int idReader, idBillBorrow;
    1. **Câu lệnh ghép**

Câu lệnh ghép là câu lệnh chứa danh sách các lệnh được đặt trong dấu ngoặc nhọn ({“câu lệnh”}), tuân theo các quy tắc sau:

* Các câu lệnh ghép phải thụt lề (khoảng cách được đề cập ở phần **3.3**) so với câu lệnh gốc.
* Dấu ngoặc nhọn được sử dụng bao quanh tất cả các câu lệnh, kể cả câu lệnh đơn lẻ, giúp dễ dàng thêm câu lệnh mà không vô tình tạo ra lỗi thiếu dấu ngoặc nhọn.
  + 1. **Câu lệnh return**

Câu lệnh return phải trả về kiểu dữ liệu trùng với kiểu khai báo của hàm.

* + 1. **Câu lệnh if – else**

Câu lệnh if-else nên được viết dưới dạng sau:

if (<điều kiện>)

{

<Câu lệnh>;

}

else

{

<Câu lệnh>;

}

* + 1. **Câu lệnh for**

Câu lệnh for nên được viết dưới dạng sau:

for (<khởi tạo>;<điều kiện lặp>;<bước lặp>)

{

<Câu lệnh>;

}

* + 1. **Câu lệnh foreach**

Câu lệnh foreach nên được viết dưới dạng sau:

foreach (<kiểu dữ liệu> <tên biến tạm> in <tên mảng hoặc tập hợp>)

{

<Câu lệnh>;

}

* + 1. **Câu lệnh while/do-while**

Câu lệnh while/do-while nên được viết dưới dạng sau:

while (điều kiện lặp)

{

<Câu lệnh>;

}

do {

<Câu lệnh>;

}while(<điều kiện lặp>);

* + 1. **Câu lệnh switch-case**

Câu lệnh switch-case nên được viết dưới dạng sau:

**s**witch(<biểu thức>)

{

case <giá trị 1>:

<câu lệnh 1>;

break;

case <giá trị 2>:

<câu lệnh 2>;

break;

…

case <giá trị n>:

<câu lệnh n>:

break;

}

* + 1. **Câu lệnh try-catch-finally**

Câu lệnh try-catch-finally nên được viết dưới dạng sau:

try

{

<Câu lệnh>;

}

catch (<tên ngoại lệ> e)

{

<Câu lệnh xử lý ngoại lệ>;

}

finally

{

<Câu lệnh>;

}

* 1. **Quy ước khoảng trắng**
     1. **Dòng trống**

Các dòng trống được dùng để cải thiện khả năng đọc bằng cách phân chia các thành phần.

Chỉ sử dụng 1 dòng trống, được dùng trong việc phân chia giữa:

* Các phương thức (Methods) và biến cố (Event)
* Các khu vực (Region)
* Các hàm.
* Các biến.
  + 1. **Khoảng trống**

Khoảng trống là những kí tự không nhìn thấy trong code. Và trong C# các ký tự được xem là khoảng trắng: ký tự báo dòng mới, kí tự space, tab.

Khoảng trống nên được sử dụng:

* Giữa một từ khóa theo sau là dấu ngoặc đơn.
  + VD:
    - while (true)

{

…

}

* Trước và sau toán tử.

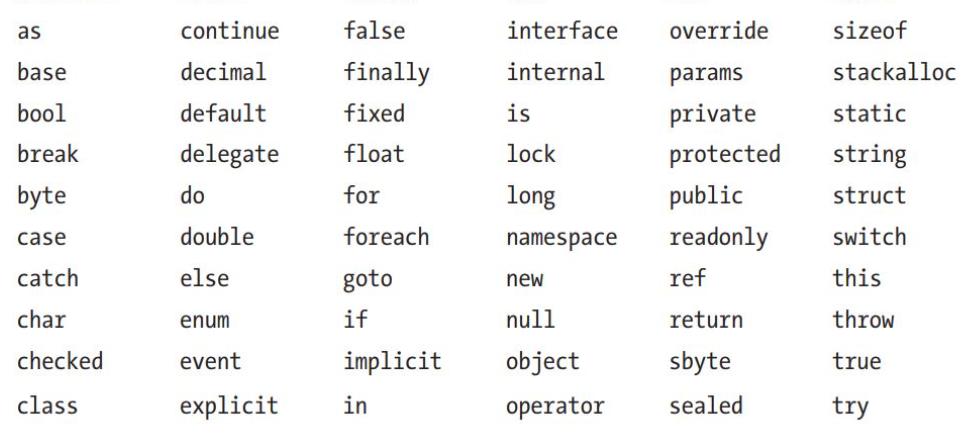
Compiler sẽ tự động bỏ qua các kí tự trắng thừa nên nó không có giá trị đối với compiler nhưng có vai trò quan trọng trong việc định dạng code, giúp code dễ hiểu hơn.

1. **Keyword (Từ khóa)**

Keyword là những từ được gán cho một ý nghĩa xác định và là nòng cốt trong cú pháp ngôn ngữ, không được sử dụng cho mục đích gì khác ngoài những thứ đã được ngôn ngữ qui định.

Từ khóa trong C# được chia làm 2 loại là: từ khóa theo ngữ cảnh và từ khóa dành riêng.

Dưới đây là ví dụ về từ khóa trong C#:



1. **Tham khảo**

“Code Conventions for the Java TM Programming Language”

[**https://www.oracle.com/java/technologies/javase/codeconventions-contents.html**](https://www.oracle.com/java/technologies/javase/codeconventions-contents.html)

“GNU Coding Standards”

[**https://www.gnu.org/prep/standards/standards.html**](https://www.gnu.org/prep/standards/standards.html)