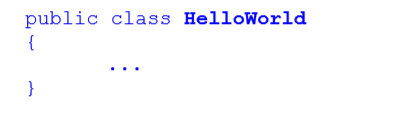
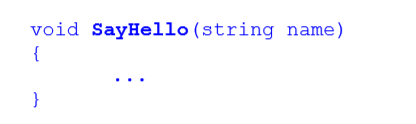
**TÀI LIỆU CHUẨN VIẾT MÃ**

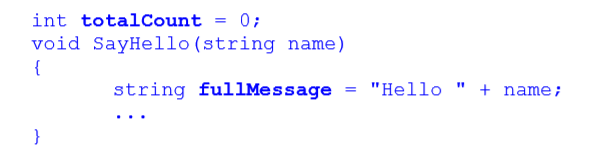
1. Sử dụng quy chuẩn “Pascal case” để đặt tên lớp. (Thêm tiền tố DTO\_ , DAL\_ , BLL\_ , GUI\_ đối với mô hình 3 lớp)



1. Sử dụng quy chuẩn “Pascal case” để đặt tên phương thức.



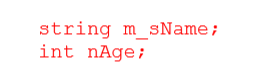
1. Sử dụng quy chuẩn “Camel case” để đặt tên biến và tham số của phương thức.



1. Sử dụng tiền tố “I” và quy chuẩn “Pascal case” khi đặt tên interface.



1. Sử dụng quy chuẩn Pascal case” để đặt tên file.
2. Không sử dụng kí hiệu Hungary để đặt tên biến.



1. Sử dụng các từ có ý nghĩa, mô tả để đặt tên biến. Không sử dụng từ viết tắt.



1. Không sử dụng các viết ký tự đơn như I, n, s … Sử dụng các tên như: index, temp. Ngoại lệ: nếu biến chỉ được làm bộ đếm cho phép lặp và không sử dụng ở bất kỳ nơi nào khác trong vòng lặp.
2. Không sử dụng dấu gạch dưới (\_) cho tên biến cục bộ.
3. Tất cả các biến thành viên phải được bắt đầu bằng dấu gạch dưới (\_) để có thể xác định chúng từ các biến cục bộ khác.
4. Không sử dụng tên biến trùng với từ khóa.
5. Các biến, thuộc tính, phương thức kiểu boolean thêm tiền tố “is” hoặc các tiền tố tương tự.



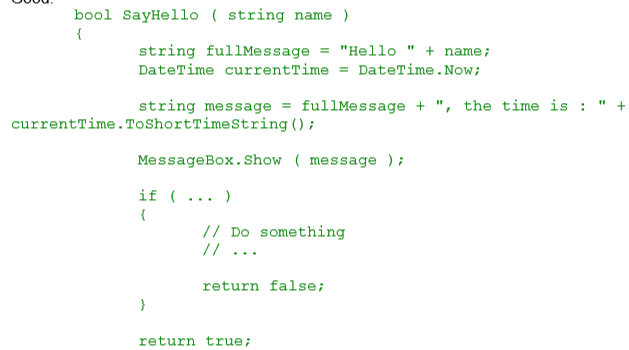
1. Tên Namespace phải theo mẫu chuẩn.
2. Sử dụng tiền tố thích hợp cho các thành phần UI để có thể xác định chúng từ phần còn lại cảu biến.

|  |  |
| --- | --- |
| **Control** | **Prefix** |
| Label | lbl |
| Textbox | txt |
| DataGrid | gtg |
| Button | btn |
| ImageButton | imb |
| HypeLink | hlk |
| DropDownList | ddl |
| ListBox | lst |
| DataList | dtl |
| Repeater | rep |
| Checkbox | chk |
| CheckboxList | cbl |
| RadioButton | rdo |
| RadioButtonList | rdl |
| Image | img |
| Panel | pnl |
| PlaceHolder | phd |
| Table | tbl |
| Validators | val |

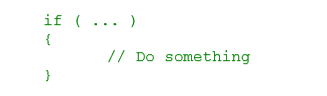
1. Tên file phải khớp với tên class.
2. Sử dụng TAB để thụt đầu dòng. Không sử dụng SPACES. Xác định kích thước TAB là 4.
3. Comment phải ở cùng cấp độ với mã. (cùng mức thụt lề)



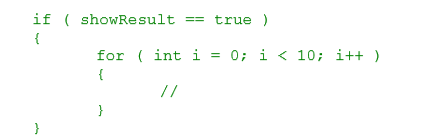
1. Sử dụng một dòng trống để phân tách các nhóm mã logic.



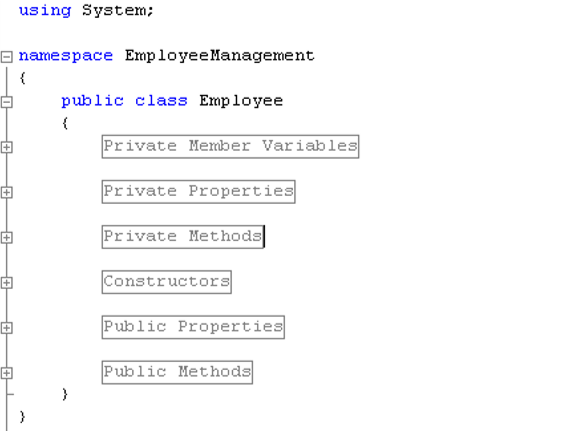
1. Cần có một và chỉ một dòng trống duy nhất giữa mỗi phương thức bên trong lớp.
2. Các dấu ngoặc nhọn phải nằm trên một dòng riêng biệt và không nằm trong cùng một dòng với “if” , “for”, …



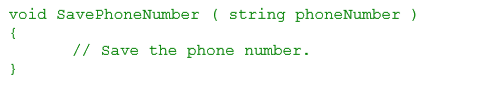
1. Sử dụng một khoảng trắng trước và sau mỗi toán tử và dấu ngoặc.



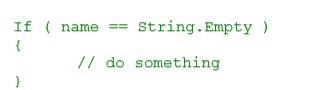
1. Sử dụng #region để nhóm các đoạn mã liên quan lại với nhau.



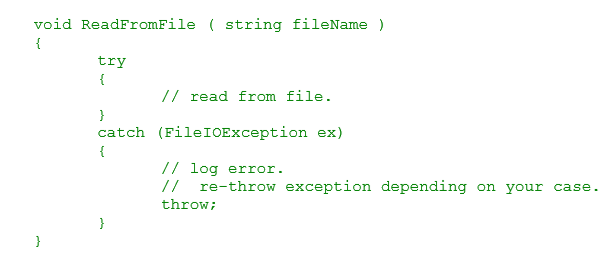
1. Giữ các biến thành viên, thuộc tính và phương thức private ở đầu tệp và public phía dưới.
2. Thánh viết phương thức quá dài ( >30 dòng).
3. Tên phương thức nên nói những gì nó làm. Không cần tài liệu giải thích những gì phương thức làm.



1. Một phương thức chỉ nên làm 1 việc.
2. Không dùng mã số cứng. Sử dụng hằng số và phai báo ở đầu tệp và sử dụng nó.
3. Không sử dụng chuỗi mã hóa cứng . Sử dụng các tập tài nguyên.
4. Sử dụng String.Empty thay cho “”.



1. Không lập trình nhấp vào nút để thực hiện cùng một hành động bạn đã viết trong nút bấm sự kiện. Thay vào đó, hãy gọi cùng một phương thức được gọi bởi sự kiện nhấn nút xử lý.
2. Không bao giờ mã hóa một đường dẫn hoặc tên ổ đĩa trong mã. Nhận đường dẫn ứng dụng theo chương trình và sử dụng đường dẫn tương đối.
3. Mở các kết nối cơ sở dữ liệu, ổ cắm, luồng tệp, v.v., hãy luôn đóng chúng trong khối cuối cùng. Điều này sẽ đảm bảo rằng ngay cả khi có ngoại lệ xảy ra sau khi mở kết nối, nó sẽ được đóng an toàn trong khối cuối cùng.



1. Khai báo các biến càng gần càng tốt đến nơi nó được sử dụng lần đầu tiên. Sử dụng một biến khai báo trên mỗi dòng.