A: Đặt tên

* Tên biến thường,biến thành viên có ký tự đầu tiên viết thường, từng từ tiếp theo có ký tự đầu viết hoa. VD: typeName, tempData
* Tất cả các phần: tên lớp, tên thuộc tính, tên phương thức, tên không gian, sự kiện, biến tĩnh, enum… đều viết hoa từng ký tự đầu
* Tên phương thức, sự kiện hàm bắt đầu bằng một động từ
* Tên Interface bắt đầu bằng chữ I
* Tên không gian, lớp, enum, thuộc tính là một cụm danh từ
* Không dùng ký tự \_ khi đặt tên biến
* Không dùng từ “enum” trong đặt tên của enum

B: Viết comment

* Không cần comment những đoạn mã đã rõ nghĩa
* VD: *// This is a for loop that prints the 1 million times*
* for (int i = 0; i < 1000000; i++) Console.WriteLine(i);
* Bản thân việc đặt tên hàm đã phải đủ nghĩa cho chức năng hoạt động chính của hàm đó, chỉ cần comment để lý giải các trường hợp hoạt động của hàm mà thôi.
* Viết comment trong trường hợp lý giải từ đầu phương thức hoạt động, các trường hợp đặc biệt cần xử lý của hàm, những đoạn chưa chắc chắn cần note lại với ký hiệu quy ước đặc biệt để tìm và sửa sau.

C: Tổ chức hàm

* Hàm cần thực hiện duy nhất một tính năng
* Một hàm chỉ nên có từ 20-25 dòng code
* Không nên truyền quá nhiều biến vào một hàm, mà hãy đóng gói các biến cần truyền vào một biến cấu trúc hoặc biến đối tượng

void HàmA() 🡪 void HàmB() 🡪 void HàmA()  
{ { {

//Line 1 HàmB()

//Line 2 //Line 1,2,3 HàmC()

//Line 3 } }

//Line 4 🡪 void HàmC()

//Line 5 {

//Line 6 //Line 4,5,6,7,8

//Line 7 }

//Line 8

}

1. Tổ chức mã nguồn

* Tránh những logic phức tạp mà có thể đóng gói những logic đó vào các hàm trả về bool

VD: if(product.Price>500 && !product.IsDeleted && !product.IsFeatured && product.IsExported)

* Tránh và cần xóa những biến được khai báo nhưng không sử dụng đến
* Tên biến cần đặt tên là danh từ có nghĩa phục vụ cho việc xử dụng
* Nên đặt tiền tố trước tên biến theo quy ước: i cho integer, d cho double, dt cho datetime...
* Nên gán giá trị thực trả về của hàm vào một biến trước khi sử dụng vào việc tiếp theo.
* Nếu trong vòng lặp, có một vài trường hợp cần kiểm tra đặc biệt thì hãy để các trường hợp đó vào hàm con
* Tránh các vòng lặp lồng trong cùng một hàm, có thể chia nhỏ những vòng lặp con vào hàm nhỏ hơn
* Suy nghĩ kỹ để sử dụng kiểu dữ liệu phù hợp để lưu trữ hay xử lý trong hàm: biến, đối tượng, mảng, danh sách, từ điển, ngăn xếp, hàng đợi, cây...
* Tránh khai báo, khởi tạo lại đối tượng trong vòng lặp nếu không cần thiết.
* Tránh gọi lại hàm có xử lý lớn trong vòng lặp nếu không cần thiết.
* Tránh thay đổi kích thước của các cấu trúc danh sách ngay trong vòng lặp duyệt nó.