

Bài 2.4: Các biến, hằng số trong C#

- ✓ Khái niệm, mục đích sử dụng
- ✓ Cú pháp khai báo biến, hằng
- ✓ Quy tắc đặt tên biến
- ✓ Giá trị mặc định của kiểu
- ✓ Kiểu ngầm định với var
- ✓ Ví dụ minh họa

Khái niệm, mục đích sử dụng

- ✓ Biến là tên đặt cho vùng nhớ dùng để lưu trữ dữ liệu.
- ✓ Nói cách khác, tên biến chính là nhãn ta gán cho một vùng nhớ nào đó.
- ✓ Mỗi biến phải có một cái tên duy nhất, không trùng nhau.
- ✓ Mục đích sử dụng của biến: lưu trữ dữ liệu phục vụ quá trình hoạt động của chương trình.
- ✓ Biến chỉ tồn tại trong thời gian ngắn, khi chương trình kết thúc thì biến sẽ bị hủy đi.
- ✓ Bộ nhớ được dùng để cấp phát cho các biến là bộ nhớ RAM.
- ✓ Giá trị lưu trong 1 biến có thể thay đổi tùy ý trong khả năng lưu trữ của kiểu biến.
- ✓ Hằng số là các biến đặc biệt chỉ gán giá trị được duy nhất 1 lần.
- ✓ Hằng số thường dùng để lưu các giá trị cố định không thay đổi trong suốt chương trình như số e, π , G,...

Cú pháp khai báo biến

- ✓ Cú pháp: **type variableName**;
- ✓ Khai báo nhiều biến cùng kiểu: **type variable1, variable2**;
- ✓ Cú pháp khởi tạo: **type variableName = value**;
- ✓ Trong đó:
 - ✓ Type là kiểu của biến, có thể là bất kì kiểu dữ liệu hợp lệ nào trong C#.
 - ✓ VariableName, variable1, variable2 là tên biến do bạn tự đặt.
 - ✓ Phần value là giá trị tương ứng cần gán cho biến.
 - ✓ Nếu khai báo nhiều biến cùng kiểu, phân tách các biến bằng dấu phẩy.
 - ✓ Ta cũng có thể khai báo và khởi tạo cho nhiều biến cùng kiểu.
 - ✓ Kết thúc khai báo biến hoặc khởi tạo biến bắt buộc phải có dấu chấm phẩy.

Quy tắc đặt tên biến

- ✓ Tên biến thường là danh từ, cụm danh từ mô tả ý nghĩa sử dụng của biến.
- ✓ Tên biến phân biệt chữ hoa chữ thường.
- ✓ Tên biến bắt đầu bằng chữ cái thường hoặc _ sau đó là chữ cái, số, _.
- ✓ Tên biến có nhiều từ thì viết liền và viết hoa chữ cái kể từ từ thứ 2.
- ✓ Tên biến không đặt trùng với tên các keyword của C#.
- ✓ Tên biến không được chứa các kí tự khác ngoài chữ cái, chữ số, dấu _.

```
float gpa; // khai báo biến gpa kiểu float
int age = 20; // khởi tạo biến age = 20
string address, fullName, major; // khai báo nhiều biến kiểu string
int x = 1, y = 2, z = 3; // khởi tạo cho nhiều biến
```

Hằng số

- ✓ Cú pháp: **const type NAME = value;**
- ✓ Trong đó:
 - ✓ Khai báo hằng số bắt buộc phải có const ở đầu.
 - ✓ Type là kiểu của hằng số.
 - ✓ NAME là tên hằng số, tên luôn viết hoa hoàn toàn và nếu nhiều từ thì phân tách bằng _.
 - ✓ Khi khai báo hằng số phải gán luôn giá trị value cho hằng số.
 - ✓ Kết thúc khai báo luôn có dấu chấm phẩy.
- ✓ Ví dụ:

```
const float PI = 3.141592f;  
const int MAX_AGE = 150; // tuổi tối đa của người  
const string MALE = "MALE"; // giới tính nam
```

Giá trị mặc định của kiểu

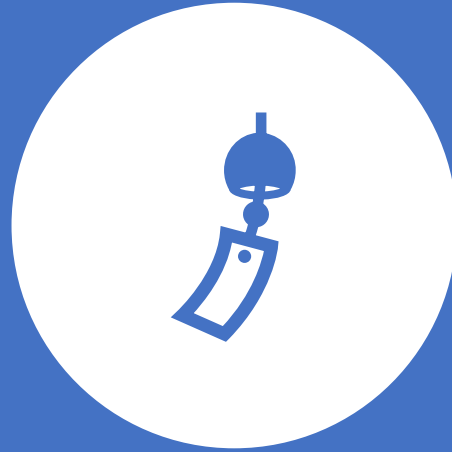
- ✓ Ta có thể gán giá trị mặc định cho biến khi khởi tạo nó bằng keyword default hoặc new().
- ✓ Các kiểu số nhận giá trị 0;
- ✓ Các kiểu tham chiếu là null;
- ✓ Kiểu bool là false;
- ✓ Kiểu char là '\0';

```
// gán giá trị mặc định của kiểu cho biến khi chưa sử dụng
int width = default; // width = 0
int height = default;
string email = default; // email = null
// hoặc dùng new type(), viết tắt là new()
float radius = new(); // radius = 0
float latitude = new float(); // latitude = 0
```

Kiểu ngầm định trong C#

- ✓ Trong C# ta có thể sử dụng tên kiểu tường minh như các ví dụ trước.
- ✓ Hoặc sử dụng kiểu ngầm định với keyword **var**.
- ✓ Cú pháp: **var** **variableName** = **value**;
- ✓ Trong đó bắt buộc phải khởi tạo giá trị cho biến khi khai báo với var.
- ✓ Mỗi lần chỉ khai báo một biến với var.
- ✓ Kiểu của biến sẽ được xác định qua giá trị ta gán cho biến.
- ✓ Phù hợp trong trường hợp chưa biết trước kiểu dữ liệu hoặc khai báo biến cục bộ.

```
var amount = 100; // amount kiểu int
var petName = "Meow"; // petName kiểu string
var brand = "Apple"; // brand kiểu string
var price = 5.25; // price kiểu double
var area = 3.52f; // area kiểu float
```



Nội dung tiếp theo

Nhập xuất dữ liệu C#