

Bài: Biến trong C#

Xem bài học trên website để ủng hộ Kteam: [Biến trong C#](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

Dẫn nhập

Ở bài [NHẬP XUẤT CƠ BẢN TRONG C#](#) chúng ta bắt gặp một câu lệnh khá lạ đó là `int a = 5`. Đây là lệnh khai báo biến và chúng ta sẽ cùng tìm hiểu chi tiết về biến trong bài học hôm nay – **BIẾN TRONG C#**.

Nội dung

Để đọc hiểu bài này tốt nhất các bạn nên có kiến thức cơ bản về các phần:

- [Cấu trúc lệnh của C# viết trên nền Console Application.](#)
- [Cấu trúc nhập xuất của C# trên nền Console Application.](#)

Trong bài học này, chúng ta sẽ cùng tìm hiểu các vấn đề:

- Biến là gì? Tại sao phải sử dụng biến?
- Khai báo và sử dụng biến.
- Quy tắc đặt tên biến.

Biến là gì? Tại sao phải sử dụng biến?

Trong toán học ta đã quá quen thuộc với thuật ngữ "**biến**". Nếu biến số trong toán học là một số có thể thay đổi thì trong lập trình biến cũng được định nghĩa tương tự:

- Là một giá trị dữ liệu có thể thay đổi được.
- Là tên gọi tham chiếu đến một vùng nhớ nào đó trong bộ nhớ.
- Là thành phần cốt lõi của một ngôn ngữ lập trình.

Tại sao phải sử dụng biến?

Lưu trữ dữ liệu và tái sử dụng: ví dụ hãy tưởng tượng nếu bạn yêu cầu người nhập vào tuổi của họ, nhưng bạn không lưu trữ lại thì đến khi bạn muốn sử dụng thì chẳng biết lấy đâu ra để sử dụng cả.

Thao tác với bộ nhớ một cách dễ dàng:

- Cấu trúc của bộ nhớ bao gồm nhiều ô nhớ liên tiếp nhau, mỗi ô nhớ có một địa chỉ riêng (địa chỉ ô nhớ thường mã hex (thập lục phân)).
- Khi muốn sử dụng ô nhớ nào (cấp phát, hủy, lấy giá trị, ...) bạn phải thông qua địa chỉ của chúng. Điều này làm cho việc lập trình trở nên khó khăn hơn.
- Thay vào đó bạn có thể khai báo một biến và cho nó tham chiếu đến ô nhớ bạn cần quản lý rồi khi sử dụng bạn sẽ dùng tên biến bạn đặt chứ không cần dùng địa chỉ của ô nhớ đó nữa. Rất tiện lợi phải không nào!

Khai báo và sử dụng biến

Cú pháp:

```
<Kiểu dữ liệu> <Tên biến>;
```

Trong đó:

- **<Kiểu dữ liệu>** có thể là:
 - Kiểu dữ liệu cơ bản.
 - Kiểu dữ liệu có cấu trúc, . . . (Sẽ được trình bày chi tiết trong bài [KIỂU DỮ LIỆU TRONG C#](#)).
- **<Tên biến>**
 - Là tên do người dùng đặt.
 - Phải tuân thủ theo quy tắc đặt tên (sẽ được trình bày ngay sau đây).

Sử dụng biến

Khai báo:

```
int BienKieuSoNguyen;  
  
float BienKieuSoThuc;  
  
string BienKieuChuoai;  
  
bool BienKieuLuanLy;  
  
char BienKieuKyTu;
```

Trong đó:

- Kiểu dữ liệu là: `int`, `float`, `string`, `bool`, `char`
- Tên biến là: `BienKieuSoNguyen`, `BienKieuSoThuc`,...

Sử dụng:

Để sử dụng biến ta cần phải gán giá trị cho nó trước. Có 2 cách gán giá trị:

- Khởi tạo giá trị lúc khai báo:

C#:

```
int BienKieuSoNguyen = 10;  
string BienKieuChuoai = "Kteam";
```

- Gán giá trị theo cú pháp:

```
<Tên biến> = <Giá trị>;
```

C#:

```
BienKieuSoNguyen = 9;  
BienKieuKyTu = 'K';
```

Còn khi bạn muốn gọi một biến ra để lấy giá trị thì bạn chỉ cần gọi tên của nó ra là được. Ví dụ:

C#:

```
Console.WriteLine(BienKieuSoNguyen);  
// In giá trị của biến tên là BienKieuSoNguyen ra màn hình. Kết quả là 9
```

hoặc

C#:

```
int a = 1, b = 2 ;
int c = a + b;
// Biến a và biến b được gọi để lấy giá trị sau đó cộng chúng lại rồi gán cho biến c.
```

Quy tắc đặt tên biến

Một số quy tắc khi đặt tên biến cũng như là các định danh khác:

- Tên biến là một chuỗi ký tự liên kết (không có khoảng trắng) và không chứa ký tự đặc biệt.
- Tên biến không được đặt bằng tiếng việt có dấu.
- Tên không được bắt đầu bằng số.
- Tên biến không được trùng nhau.
- Tên biến không được trùng với từ khóa:

Dưới đây là danh sách các từ khóa trong C#, các bạn chỉ cần nắm để tránh đặt tên trùng với từ khóa còn việc mặt ý nghĩa từ khóa sẽ được trình bày trong suốt các bài học sau này.

abstract	default	foreach	object	sizeof	unsafe
as	delegate	goto	operator	stackalloc	ushort
base	do	if	out	static	using
bool	double	implicit	override	string	virtual
break	else	in	params	struct	volatile
byte	enum	int	private	switch	void
case	event	interface	protected	this	while
catch	explicit	internal	public	throw	
char	extern	is	readonly	true	
checked	false	lock	ref	try	
class	finally	long	return	typeof	
const	fixed	namespace	sbyte	uint	
continue	float	new	sealed	ulong	
decimal	for	null	short	unchecked	

Từ khóa của ngôn ngữ C#.

Ngoài ra các lập trình viên cũng đưa ra một số quy tắc chung trong việc đặt tên để dễ quản lý và giúp cho người khác có thể dễ dàng đọc code của mình

Quy tắc Lạc Đà

Viết thường từ đầu tiên và viết hoa chữ cái đầu tiên của những từ tiếp theo.

Thường được dùng để đặt tên cho các biến có phạm vi truy cập là [private](#) hoặc [protected](#) (phạm vi truy cập sẽ được trình bày chi tiết trong [CLASS TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG C#](#)) và các tham số của hàm (tham số của hàm sẽ được trình bày chi tiết trong bài [CẤU TRÚC HÀM CƠ BẢN TRONG C#](#)).

Ví dụ: educationFree, howKteam, ...

Quy tắc Pascal

Viết hoa chữ cái đầu tiên của mỗi từ.

Thường được dùng để đặt tên cho những thành phần còn lại như hàm (sẽ được trình bày trong bài [CẤU TRÚC HÀM CƠ BẢN TRONG C#](#)), Interface (sẽ được trình bày trong bài [INTERFACE TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG C#](#)), Enum (sẽ được trình bày trong bài [ENUM TRONG C#](#)), Sự kiện (sẽ được trình bày trong bài [EVENT TRONG C# NÂNG CAO](#)), ...

Một số lưu ý khi đặt tên biến:

- Nên đặt tên ngắn gọn dễ hiểu, thể hiện rõ mục đích của biến. Ví dụ như: Name, Tuổi, GiớiTinh, . . .
- Không nên đặt tên biến bằng một ký tự như i, k, m, . . . như vậy sau này khi xem lại code hoặc đưa code cho người khác đọc thì chúng ta sẽ không hiểu biến này dùng để làm gì. Trừ những trường hợp đặc biệt (sẽ nói trong những bài học sau).
- C# có phân biệt chữ hoa chữ thường. Ví dụ biến `a` khác biến `A` hoặc lệnh `Console.WriteLine()` khác lệnh `Console.WRITELINE()`.

Những điều này là chuẩn chung của mọi lập trình viên, nếu bạn làm khác đi cũng không có vấn đề gì nhưng khi chương trình bạn gặp lỗi thì người khác khó giúp đỡ bạn giải quyết vì họ không hiểu được code bạn viết thì làm sao giúp bạn được!

Do bài này thiên về lý thuyết nhiều nên Kteam sẽ không có ví dụ demo. Kiến thức bài này chủ yếu làm nền tảng cho những bài sau nên các bạn cần nắm rõ để những bài học không bỏ ngỡ.

Kết luận

Nội dung bài này giúp các bạn nắm được:

- Khái niệm về biến và tại sao lại phải sử dụng biến.
- Cách khai báo và sử dụng biến.
- Các quy tắc đặt tên biến.

Bài học sau chúng ta sẽ cùng tìm hiểu một khái niệm tiếp theo đó là [KIỂU DỮ LIỆU TRONG C#](#)

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên "**Luyện tập – Thử thách – Không ngại khó**".