

Hằng (Constant/Literal) trong C/C++

Constant liên quan tới các giá trị cố định mà chương trình không thể thay đổi và chúng được gọi là literals.

Constant là một kiểu dữ liệu thay thế cho Literal, còn Literal thể hiện chính nó. Trong ví dụ: `const PI = 3.14` thì Constant ở đây là `PI`, còn Literal là `3.14`.

Constant có thể là bất kỳ kiểu dữ liệu cơ bản nào trong C/C++, và có thể được phân chia thành giá trị hằng số nguyên, hằng số thực, hằng ký tự, hằng chuỗi và Boolean literal (tạm dịch: hằng logic).

Ngoài ra, constant được đối xử giống như các biến thông thường, ngoại trừ việc giá trị của chúng là không thể thay đổi sau khi định nghĩa.

Hằng số nguyên trong C/C++

Hằng số nguyên có thể là decimal (cơ số 10), octal (cơ số 8) hay hexadecimal (cơ số 16). Giá trị có tiền tố (prefix) là `0` cho octal, là `0x` hay `0X` cho hexadecimal và không có gì cho decimal.

ADVERTISING

Một hằng số nguyên cũng có các hậu tố (suffix) `U` hay `L` thể hiện kiểu unsigned hay long. Hậu tố có thể là chữ hoa hoặc chữ thường và có thể trong bất kỳ thứ tự nào.

Ví dụ về các hằng số nguyên:

```
212 // Hop 1e
```

```
215u // Hop 1e
```

```
0xFeEL // Hop le
```

```
048 // Khong hop le: 8 khong phai la mot ky so trong he octal
```

```
032UU // Khong hop le: hau to (suffix) khong the bi lap
```

Dưới đây là các kiểu hằng số nguyên đa dạng:

```
85 // Hang so dang decimal
```

```
0213 // Hang so dang octal
```

```
0x4b // Hang so dang hexadecimal
```

```
30 // Hang so dang so nguyen (int)
```

```
30u // Hang so dang so nguyen khong dau (unsigned int)
```

```
30l // Hang so dang long
```

```
30ul // Hang so dang unsigned long
```

Hằng số thực trong C/C++

Một hằng số thực bao gồm phần nguyên (integer part), dấu chấm thập phân (decimal point), phần lẻ (fraction part) và phần mũ (exponent part). Chúng ta có thể biểu diễn hằng số thực theo dạng decimal hay dạng mũ.

Khi thể hiện dạng decimal phải bao gồm dấu chấm thập phân, phần mũ hoặc cả hai. Khi thể hiện dạng mũ phải gồm phần nguyên, phần lẻ hoặc cả hai. Dạng mũ đi kèm với kí tự E hoặc e.

Xét các ví dụ sau:

```
3.14159 // Hop le
```

```
314159E-5L // Hop le
```

```
510E // Khong hop le: pham mu (exponent) con thieu
```

```
210f // Khong hop le: khong co phan thap pha decimal hoac phan mu  
exponent
```

```
.e55 // Khong hop le: thieu phan nguyen integer hoac phan fraction
```

Boolean literal trong C/C++

Có hai kiểu Boolean literal và chúng là một phần của các từ khóa C/C++:

- Giá trị true
- Và giá trị false

Bạn không nên đồng nhất giá trị true với 1 và giá trị false với 0.

Hằng ký tự trong C/C++

Các hằng kí tự trong C/C++ mở đầu và kết thúc bởi dấu nháy đơn. Nếu hằng ký tự bắt đầu với L (ví dụ L'x') thì nó là kiểu wchar_t. Nếu không thì, nó là hằng ký tự kiểu char, ví dụ như 'x'.

Hằng kí tự có thể là một kí tự (như 'X'), một escape sequence (như '\t') hay một kí tự mở rộng (như '\u02c0').

Một số kí tự trong C/C++ khi được đứng trước bởi dấu \ thì chúng sẽ mang một ý nghĩa đặc biệt như bắt đầu dòng mới '\n' hay tạo một tab '\t'. Chúng được biết như là escape sequence (dãy thoát). Bảng dưới đây thể hiện một số mã escape sequence phổ biến:

Escape sequence	Ý nghĩa
\\	Ký tự \
\'	Ký tự '
\"	Ký tự "

\?	Ký tự ?
\a	Alert hoặc bell
\b	Backspace
\f	Form feed
\n	Newline
\r	Carriage return
\t	tab ngang
\v	tab dọc
\ooo	Số hệ cơ số 8 của một tới 3 chữ số
\xhh . . .	Số hệ cơ số 16 của một hoặc nhiều chữ số

Dưới đây là ví dụ minh họa một số ký tự escape sequence:

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello\tWorld\n\n";

    return 0;
}
```

```
}
```

Khi code trên được biên dịch và thực thi, nó cho kết quả sau:

```
Hello World
```

Hằng chuỗi trong C/C++

Hằng chuỗi chứa trong dấu nháy kép, ví dụ "abc". Một chuỗi sẽ chứa các kí tự tương tự hằng kí tự, gồm các ký tự thuần, escape sequence, và các ký tự mở rộng.

Có thể ngắt một dòng dài thành nhiều dòng bởi sử dụng hằng chuỗi và phân biệt chúng bởi sử dụng khoảng trắng (whitespace).

Xét ví dụ một hằng chuỗi trong C/C++ thể hiện theo 3 cách khác nhau:

```
"hoc, lap trinh"
```

```
"hoc, \
```

```
lap trinh"
```

```
"hoc, " "lap" "trinh"
```

Định nghĩa hằng trong C/C++

Có hai cách định nghĩa hằng trong C/C++ là:

- Sử dụng bộ tiền xử lý #define.
- Sử dụng từ khóa const.

Sử dụng bộ tiền xử lý #define trong C/C++

Dưới đây là cú pháp để sử dụng tiền xử lý #define để định nghĩa một constant trong C/C++:

```
#define identifier value
```

Ví dụ sau giải thích cú pháp trên:

```
#include <iostream>
```

```
using namespace std;
```

```
#define CHIEUDAI 10
```

```
#define CHIEURONG 5
```

```
#define NEWLINE '\n'
```

```
int main()
```

```
{
```

```
    int dientich;
```

```
    dientich = CHIEUDAI * CHIEURONG;
```

```
    cout << dientich;
```

```
    cout << NEWLINE;
```

```
    return 0;
```

```
}
```

Khi code trên được biên dịch và thực thi, nó cho kết quả sau:

```
50
```

Sử dụng từ khóa const trong C/C++

Bạn có thể sử dụng tiền tố const để định nghĩa hằng trong C/C++ với một kiểu cụ thể, như sau:

```
const kiểu_gia_tri bien = giaTri;
```

Ví dụ sau giải thích chi tiết cú pháp trên:

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    const int CHIEUDAI = 10;
```

```
    const int CHIEURONG = 5;
```

```
    const char NEWLINE = '\\n';
```

```
    int area;
```

```
    dientich = CHIEUDAI * CHIEURONG;
```

```
    cout << dientich;
```

```
    cout << NEWLINE;
```

```
    return 0;
```

```
}
```

Khi code trên được biên dịch và thực thi, nó cho kết quả sau:

```
50
```

Ghi chú: Nó là bước thực hành tốt cho bạn khi định nghĩa các hằng ở kiểu chữ hoa trong C/C++.