



Biến và Kiểu dữ liệu

Phiên 2

Chỉ dành cho Trung tâm Aptech sử dụng



Mục tiêu

- Thảo luận các biến
- Phân biệt giữa biến và hằng số
- Liệt kê các kiểu dữ liệu khác nhau và sử dụng chúng trong các chương trình C
- Thảo luận về toán tử số học

Biến số

Dữ liệu 15

Ký ức



| | | | |
|--|----------------------|--|--|
| | | | |
| | 15 | | |
| | Data trong bộ nhớ và | | |

Mỗi vị trí trong bộ nhớ là duy nhất

Biến cho phép cung cấp tên có ý nghĩa cho vị trí trong bộ nhớ



Ví dụ

BẮT ĐẦU

HIỂN THỊ 'Nhập 2 số'

ĐẦU VÀO A, B

$C = A + B$

HIỂN THỊ C

KẾT THÚC

A, B và C là các biến trong mã giả

Tên biến loại bỏ nhu cầu của một lập trình viên để truy cập vào các vị trí bộ nhớ sử dụng địa chỉ của họ

Hệ điều hành đảm nhiệm việc phân bổ không gian cho các biến

Để tham chiếu đến giá trị trong không gian bộ nhớ, chúng ta chỉ cần sử dụng tên biến



Hằng số

- Một **không thay đổi** là một giá trị mà giá trị của nó không bao giờ thay đổi
- Ví dụ
 - 5 **hằng số số / số nguyên hằng**
 - 5.3 **số số / số thực**
 - 'Đen' **hằng số chuỗi**
 - 'C' **Hằng số ký tự**
- Các biến giữ giá trị không đổi



Tên định danh

- Tên của các biến, hàm, nhãn và nhiều đối tượng khác do người dùng xác định được gọi là định danh
- Một số tên định danh chính xác
 - dấu trường
 - số_s
 - đánh dấu40
 - lớp_một
- Ví dụ về các định danh sai
 - lần thứ nhất
 - Ôi trời ơi !không hợp lệ
 - bắt đầu...kết thúc
- Mã định danh có thể có độ dài tùy ý, nhưng số lượng ký tự trong một biến được trình biên dịch nhận dạng sẽ khác nhau tùy theo trình biên dịch.
- Các định danh trong C phân biệt chữ hoa chữ thường



Hướng dẫn đặt tên Nhận dạng

Tên biến phải bắt đầu bằng một chữ cái

Ký tự đầu tiên có thể được theo sau bởi các ký tự chữ và số

Nên tránh sử dụng tên riêng khi đặt tên biến

Tên biến phải có ý nghĩa và mô tả

Nên tránh các chữ cái gây nhầm lẫn

Một số quy ước đặt tên biến chuẩn nên được tuân theo
trong khi lập trình



Từ khóa

- Từ khóa: Mọi ngôn ngữ đều dành riêng một số từ nhất định để sử dụng nội bộ
- Từ khóa có ý nghĩa đặc biệt trong bối cảnh của ngôn ngữ cụ thể
- Không có vấn đề xung đột miễn là có thể phân biệt được từ khóa và tên biến.
Ví dụ, có số *nguyên* vì tên biến hoàn toàn hợp lệ ngay cả khi nó chứa từ khóa **số nguyên**



Kiểu dữ liệu-1

- Các loại dữ liệu khác nhau được lưu trữ trong các biến.
Một số ví dụ là:

- Số

- Số nguyên. Ví dụ, 10 hoặc 178993455
- Số thực. Ví dụ: 15,22 hoặc 15463452,25
- Số dương
- Số âm

- Tên. Ví dụ: John Giá trị logic. Ví dụ:

- Y hoặc N



Kiểu dữ liệu-2

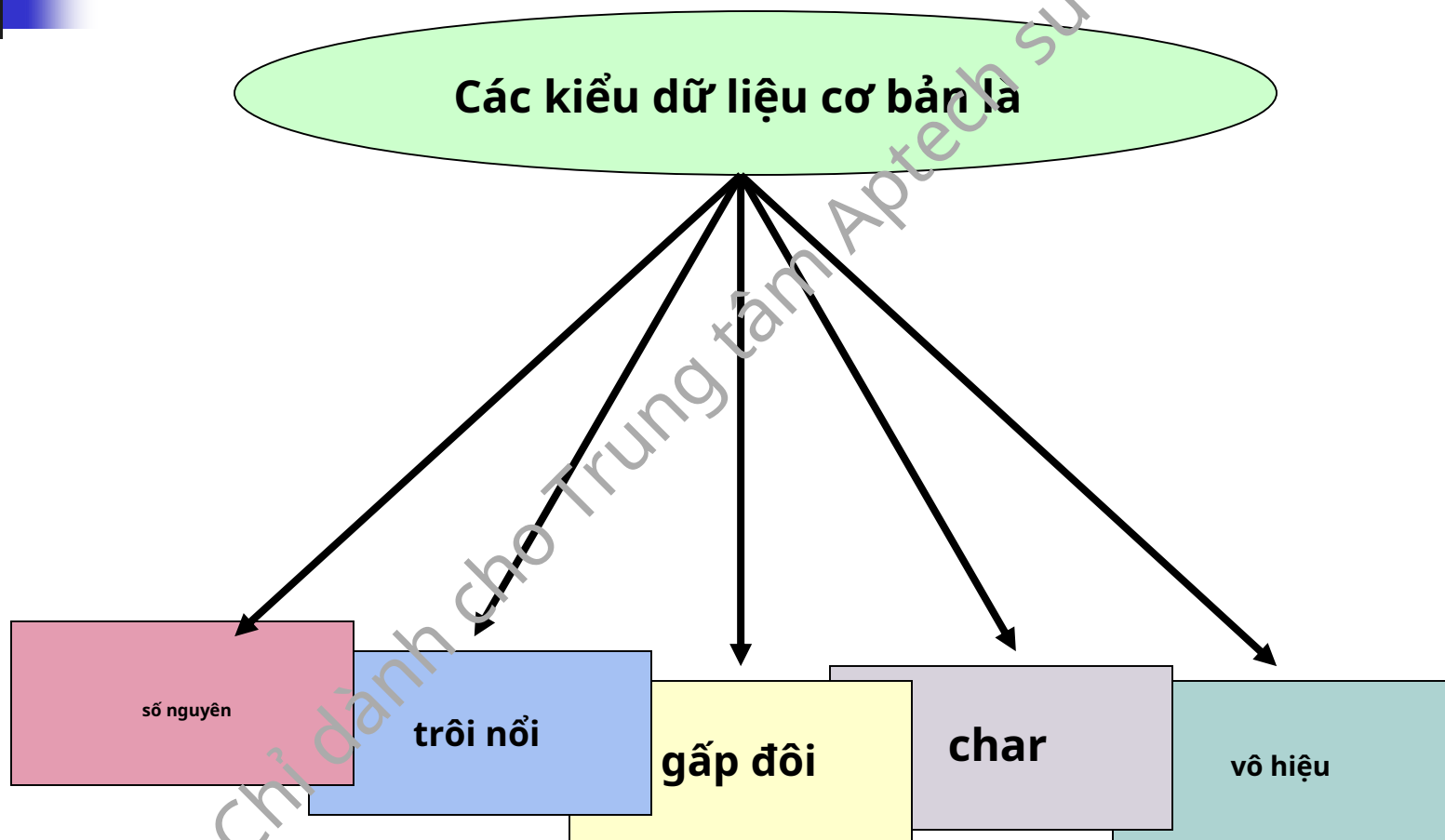
Một kiểu dữ liệu mô tả loại dữ liệu sẽ phù hợp với một
Tên của biến được đặt trước ^{biến đổi} bằng kiểu dữ liệu

Ví dụ, kiểu dữ liệu int sẽ đứng trước tên varName

Kiểu dữ liệu tên biến

int tên biến

Các kiểu dữ liệu cơ bản





Kiểu int

- Lưu trữ dữ liệu số
int số;
- Sau đó không thể lưu trữ bất kỳ loại dữ liệu nào khác như "Alan" hoặc "abc"
- 16 bit (2 byte)
- Số nguyên trong phạm vi -32768 đến 32767
- Ví dụ: 12322, 0, -232



Kiểu float

- Lưu trữ các giá trị có chứa chữ số thập phân

số thực;

- Độ chính xác lên đến 6 chữ số
- 32 bit (4 byte) bộ nhớ
- Ví dụ: 23.05, 56.5, 32



Loại đôi

- Lưu trữ các giá trị có chứa chữ số thập phân

số kép;

- Độ chính xác lên đến 10 chữ số
- 64 bit (8 byte) bộ nhớ
- Ví dụ: 'a', 'm', '\$' '%', '1', '5'



Kiểu char

- Lưu trữ một ký tự thông tin duy nhất

**giới tính nhân vật;
giới tính='M';**

- 8 bit (1 byte) bộ nhớ
- Ví dụ: 'a', 'm', '\$' '%', '1', '5'



Loại bỏ

- Không lưu trữ gì cả
- Chỉ ra trình biên dịch rằng không có gì mong đợi

Kiểu dữ liệu phái sinh

Kiểu dữ liệu
Các trình sửa đổi

+

Dữ liệu cơ bản
các loại

=

Kiểu dữ liệu phái sinh

chưa ký

+

số nguyên

=

số nguyên không dấu
(Chỉ có giấy phép
số dương)

ngắn

+

số nguyên

=

ngắn gọn
(Chiếm ít hơn
ký ức
khoảng cách hơn int)

dài

+

int/đôi

=

Số nguyên dài /longdouble
(Chiếm nhiều hơn
không gian hơn
số nguyên/đôi)

đã ký và chưa ký

Các loại

- **chưa ký** kiểu chỉ định rằng một biến chỉ có thể nhận các giá trị dương

```
int không dấu varNum;  
số=23123;
```

- varNum được phân bổ 2 byte
- có thể sử dụng trình sửa đổi với **số nguyên** và **trôi nổi** các kiểu dữ liệu
- unsigned int hỗ trợ phạm vi từ 0 đến 65535



Loại dài và ngắn

- **ngắn gọn** chiếm 8 bit (1 byte)
 - cho phép các số trong phạm vi -128 đến 127
- **số nguyên dài** chiếm 32 bit (4 byte)
 - 2.147.483.647 và -2.147.483.647
- **dài gấp đôi** chiếm 128 bit (16 byte)

Các kiểu dữ liệu và chúng phạm vi-1

| Kiểu | Kích thước gần đúng trong Bit | Phạm vi tối thiểu |
|---------------------|-------------------------------|---------------------|
| char | 8 | - 128 đến 127 |
| chưa ký | 8 | 0 đến 255 |
| ký tự đã ký | 8 | - 128 đến 127 |
| số nguyên | 16 | - 32.768 đến 32.767 |
| số nguyên không dấu | 16 | 0 đến 65.535 |
| đã ký | 16 | Giống như int |
| ngắn gọn | 16 | Giống như int |
| int ngắn không dấu | 8 | 0 đến 65, 535 |

Các kiểu dữ liệu và chúng phạm vi-2

| Kiểu | Xấp xỉ Kích thước tính theo Bit | Phạm vi tối thiểu |
|----------------------|------------------------------------|-----------------------------------|
| ký hiệu int ngắn | 8 | Giống như int ngắn |
| ký hiệu int ngắn | 8 | Giống như int ngắn |
| số nguyên dài | 32 | - 2.147.483.647 đến 2.147.483.647 |
| số nguyên dài có dấu | 32 | 0 đến 4.294.967.295 |
| int dài không dấu | 32 | 0 đến 4.294.967.295 |
| trôi nổi | 32 | Sáu chữ số chính xác |
| gấp đôi | 64 | Mười chữ số chính xác |
| dài gấp đôi | 128 | Mười chữ số chính xác |



Mẫu Tuyên Bố

chủ yếu ()

{

ký tự abc; /*abc kiểu ký tự */ /*xyz

số nguyên xyz; kiểu số nguyên */

chiều dài phao; /*chiều dài của kiểu float */

diện tích gấp đôi; /*diện tích kiểu double

đèn dài; */ /*liteyrs kiểu long int */

cánh tay ngắn; /*cánh tay kiểu số nguyên ngắn*/

}