

DỮ LIỆU CÓ CẤU TRÚC



Mục tiêu

- Nắm vững khái niệm kiểu dữ liệu trừu tượng.
- Nắm vững và cài đặt được các kiểu dữ liệu trừu tượng cơ bản như mảng, xâu, struct bằng một ngôn ngữ lập trình C++.
- Vận dụng được các kiểu dữ liệu trừu tượng để giải quyết bài toán đơn giản trong thực tế.

Nội dung

- Kiểu mảng
 - Mảng một chiều
 - Mảng hai chiều
- Kiểu xâu
- Các kiểu dữ liệu có cấu trúc

Mảng

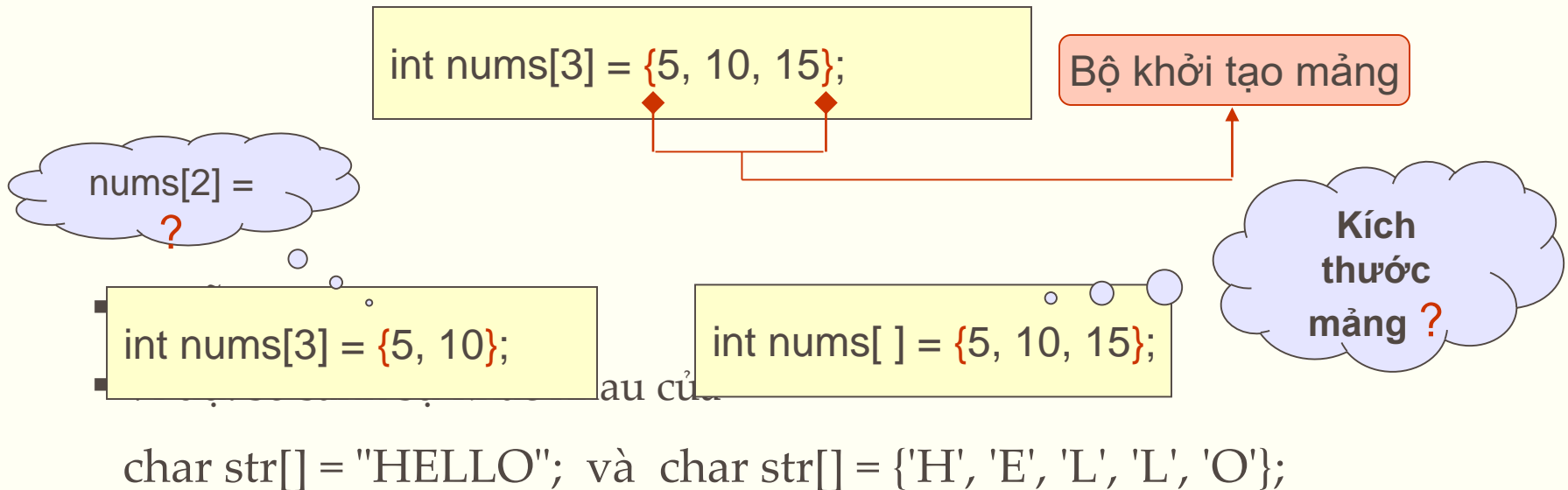
- Mảng (array)
 - Dãy hữu hạn các đối tượng cùng kiểu và được sắp xếp liên tiếp trong bộ nhớ
 - Mỗi **phần tử mảng** được xác định bởi một **chỉ số** biểu thị vị trí của phần tử trong mảng
 - Phần tử đầu tiên của mảng được mặc định có chỉ số 0
 - Số lượng phần tử trong mảng được gọi là **kích thước** của mảng (cố định; xác định trước)
 - Gồm mảng một chiều và mảng nhiều chiều

Biến Mảng

- Được định nghĩa theo cú pháp:
 - Kiểu_dữ_liệu tên_mảng[kích_thước];
 - Ví dụ: `int heights[10];`
- Truy xuất 1 phần tử qua chỉ số mảng
 - Ví dụ: `heights[0]= 210; cout<< heights[3];`
- Truy xuất phần tử không tồn tại → lỗi vượt ngoài biên
 - Ví dụ: `cout<<heights[-1]; cout<<heights[10];`

Bộ Khởi Tạo Mảng

- Mỗi mảng có một **bộ khởi tạo mảng**
- Ví dụ

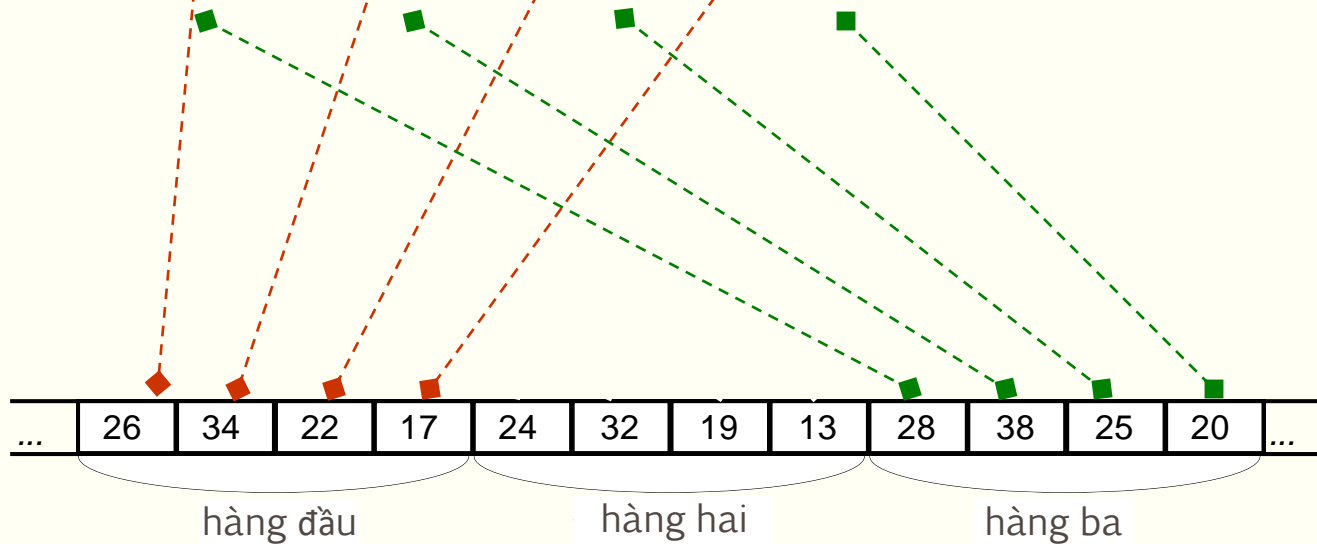


Mảng hai chiều (tiếp...)

- Khởi tạo, mỗi dòng trong 1 cặp ngoặc
 - `int b[2][2] = { { 1, 2 }, { 3, 4 } };`
 - `int b[2][2] = { { 1 }, { 3, 4 } };`

Mảng hai Chiều

	Mùa xuân	Mùa hè	Mùa thu	Mùa đông
Sydney	26	34	22	17
Melbourne	24	32	19	13
Brisbane	28	38	25	20



Cách tổ chức trong bộ nhớ

Xâu ký tự

- Định nghĩa: Là một dãy các ký tự liên tiếp nhau.
- Chuỗi các ký tự được coi như là một single unit
- Có thể bao gồm chữ cái, chữ số, ký tự đặc biệt `+`, `-`, `*` ...
- Hằng xâu ký tự - String literal (string constants)
 - Viết trong cặp nháy kép, ví dụ: `"I like C++"`
- Mảng của các ký tự, kết thúc với ký tự rỗng (null character) `'\0'`
- Xâu là một hằng con trỏ (constant pointer)
 - Trỏ đến ký tự đầu tiên của xâu
 - Giống như với mảng

Xâu ký tự (tiếp...)

- Gán giá trị cho xâu - String assignment
 - Mảng của ký tự
 - **char color[] = "blue";**
 - Tạo mảng **color** 5 phần tử kiểu **char**
 - phần tử cuối cùng là '**\0**'
 - Biến kiểu **char ***
 - **char *colorPtr = "blue";**
 - Tạo con trỏ **colorPtr** trỏ đến chữ **b** trong xâu **"blue"**
 - **"blue"** ở đâu đó trong bộ nhớ
 - Một cách khác cho mảng ký tự
 - **char color[] = { 'b', 'l', 'u', 'e', '\0' };**

- Các kiểu dữ liệu phức hợp cấu tạo từ các thành phần thuộc các kiểu dữ liệu khác
 - Tạo kiểu dữ liệu mới - kiểu dữ liệu người dùng tự định nghĩa (user-defined data type)
- Bản ghi
 - Gồm nhiều trường, mỗi trường lưu trữ một thành viên dữ liệu thuộc một kiểu dữ liệu cài sẵn hoặc một kiểu dữ liệu người dùng khác.

Kiểu cấu trúc (tiếp...)

- **struct** definition

```
struct Time {  
    int hour;  
    int minute;  
    int second;  
};
```

- Quy tắc đặt tên cho các thành viên của cấu trúc
 - Trong cùng **struct**: không thể trùng tên
 - Trong các **struct** khác nhau: có thể trùng tên
- Định nghĩa **struct** phải kết thúc bằng dấu chấm phẩy.
 - Các biến kiểu cấu trúc được khai báo như các biến thuộc các loại khác

Tổng kết

- Dữ liệu kiểu cấu trúc giúp linh hoạt trong việc tổ chức dữ liệu để giải quyết bài toán

Tiếp theo ...

- Kiểu dữ liệu danh sách, tập hợp