## **Struct:Kiểu dữ liệu cấu trúc người dùng tự định nghĩa**

Khi nào chúng ta cần phải tự định nghĩa 1 kiểu cấu trúc? Khi bạn cần lưu trữ một đối tượng có nhiều thuộc tính. Ví dụ, đối tượng SinhVien có các thuộc tính (Mã sinh viên, họ, tên, giới tính, quê quán,…) hay đối tượng LopHoc có các thuộc tính (Mã lớp, tên lớp, giáo viên chủ nhiệm, sĩ số,…). Khi đó chúng ta nên dùng struct để quản lý chương trình.

#### **Cú pháp định nghĩa struct trong C**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | struct structureName  {      dataType member1;      dataType member2;      ...  }; |

Dưới đây là 1 ví dụ:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | struct SinhVien {      int maSV;      char ho[20];      char ten[20];      bool gioiTinh;      char queQuan[100];  }; |

Như vậy, kiểu dữ liệu SinhVien đã được định nghĩa. Từ đây chúng ta có thể khai báo các biến với kiểu dữ liệu này.

## **Union :Kiểu dữ liệu cấu trúc người dùng tự định nghĩa giống struct nhưng các member dùng chung 1 vùng nhớ.**

Định nghĩa giống struct

## **So sánh struct và union**

Về mặt ý nghĩa, struct và union cơ bản giống nhau. Tuy nhiên, về mặt lưu trữ trong bộ nhớ, chúng có sự khác biệt rõ rệt như sau:

* struct: Dữ liệu của các thành viên của struct được lưu trữ ở những vùng nhớ khác nhau. Do đó kích thước của 1 struct tối thiểu bằng kích thước các thành viên cộng lại tại vì còn phụ thuộc vào bộ nhớ đệm (struct padding)
* Union : Dữ liệu các thành viên sẽ dùng chung 1 vùng nhớ. Kích thước của union được tính là kích thước lớn nhất của kiểu dữ liệu trong union. Việc thay đổi nội dung của 1 thành viên sẽ dẫn đến thay đổi nội dung của các thành viên khác.