



Từ khóa `typedef`

Tự định nghĩa một  
kiểu dữ liệu

Tại sao cần phải tự định nghĩa  
kiểu dữ liệu?

Cấu trúc trong C/C++

Sử dụng kiểu dữ liệu tự định  
nghĩa

# Kiểu dữ liệu do NGƯỜI DÙNG TỰ ĐỊNH NGHĨA

*Kỹ thuật Lập trình (CO1027)*

Ngày 8 tháng 3 năm 2021

**ThS. Trần Ngọc Bảo Duy**

*Khoa Khoa học và Kỹ thuật Máy tính*

*Trường Đại học Bách Khoa, ĐHQG-HCM*



## ① Từ khóa `typedef`

## ② Tự định nghĩa một kiểu dữ liệu

Tại sao cần phải tự định nghĩa kiểu dữ liệu?

Cấu trúc trong C/C++

Sử dụng kiểu dữ liệu tự định nghĩa



Từ khóa `typedef` cho phép người lập trình tạo ra tên mới cho một kiểu dữ liệu đã có:

- Tên mới mang lại tính dễ hiểu hơn, trong ngữ cảnh của bài toán đang xét.

### Từ khóa `typedef`

#### Tự định nghĩa một kiểu dữ liệu

Tại sao cần phải tự định nghĩa kiểu dữ liệu?

Cấu trúc trong C/C++

Sử dụng kiểu dữ liệu tự định nghĩa



Từ khóa `typedef` cho phép người lập trình tạo ra tên mới cho một kiểu dữ liệu đã có:

- Tên mới mang lại tính dễ hiểu hơn, trong ngữ cảnh của bài toán đang xét.
- Tên mới giúp rút ngắn mã nguồn.

### Từ khóa `typedef`

#### Tự định nghĩa một kiểu dữ liệu

Tại sao cần phải tự định nghĩa kiểu dữ liệu?

Cấu trúc trong C/C++

Sử dụng kiểu dữ liệu tự định nghĩa



Từ khóa `typedef` cho phép người lập trình tạo ra tên mới cho một kiểu dữ liệu đã có:

- Tên mới mang lại tính dễ hiểu hơn, trong ngữ cảnh của bài toán đang xét.
- Tên mới giúp rút ngắn mã nguồn.
- Tên kiểu mới có thể được dùng như kiểu cơ bản trong định nghĩa của tên này.

### Từ khóa `typedef`

#### Tự định nghĩa một kiểu dữ liệu

Tại sao cần phải tự định nghĩa kiểu dữ liệu?

Cấu trúc trong C/C++

Sử dụng kiểu dữ liệu tự định nghĩa



### Từ khóa `typedef`

#### Tự định nghĩa một kiểu dữ liệu

Tại sao cần phải tự định nghĩa  
kiểu dữ liệu?

Cấu trúc trong C/C++

Sử dụng kiểu dữ liệu tự định  
nghĩa

Từ khóa `typedef` cho phép người lập trình tạo ra tên mới cho một kiểu dữ liệu đã có:

- Tên mới mang lại tính dễ hiểu hơn, trong ngữ cảnh của bài toán đang xét.
- Tên mới giúp rút ngắn mã nguồn.
- Tên kiểu mới có thể được dùng như kiểu cơ bản trong định nghĩa của tên này.

Cú pháp cho một lệnh `typedef`:

```
typedef <data_type> <new_data_type>;
```

# Từ khóa typedef: Ví dụ

User-defined  
data types

ThS.  
Trần Ngọc Bảo Duy



## Từ khóa typedef

### Tự định nghĩa một kiểu dữ liệu

Tại sao cần phải tự định nghĩa  
kiểu dữ liệu?

Cấu trúc trong C/C++

Sử dụng kiểu dữ liệu tự định  
nghĩa

```
1  #include <iostream>
2  using namespace std;
3
4  typedef unsigned char byte;
5  int main()
6  {
7      byte a = 78;
8      unsigned char b = 'A', c;
9      c = a; a = b;
10     cout << "a=" << a << ":" << (int) a << endl;
11     cout << "b=" << b << ":" << (int) b << endl;
12     cout << "c=" << c << ":" << (int) c << endl;
13     return 0;
14 }
```

# Sự cần thiết của kiểu dữ liệu tự định nghĩa

User-defined  
data types

ThS.  
Trần Ngọc Bảo Duy



Từ khóa `typedef`

Tự định nghĩa một  
kiểu dữ liệu

Tại sao cần phải tự định nghĩa  
kiểu dữ liệu?

Cấu trúc trong C/C++

Sử dụng kiểu dữ liệu tự định  
nghĩa

## Vấn đề

Một chương trình cần lưu và xử lý các dữ liệu của nhiều sinh viên, trong đó thông tin của một sinh viên bao gồm:

- ① Mã số sinh viên
- ② Họ và tên
- ③ Ngày sinh
- ④ Giới tính
- ⑤ Địa chỉ
- ⑥ Số điện thoại
- ⑦ Email
- ⑧ ...



# Sự cần thiết của kiểu dữ liệu tự định nghĩa

User-defined  
data types

ThS.  
Trần Ngọc Bảo Duy



Từ khóa `typedef`

Tự định nghĩa một  
kiểu dữ liệu

Tại sao cần phải tự định nghĩa  
kiểu dữ liệu?

Cấu trúc trong C/C++

Sử dụng kiểu dữ liệu tự định  
nghĩa

## Vấn đề

Nếu chỉ sử dụng các kiểu cơ bản để lưu trữ tạm một sinh viên trong bộ nhớ, người lập trình cần khai báo **NHIỀU** biến đơn lẻ, mỗi cho một dữ liệu của một sinh viên:

- Quá bất tiện: dài dòng, khó hiểu, v.v
- Khi cần vài sinh viên trong bộ nhớ: các dòng khai báo biến đã chiếm một vùng lớn mã nguồn!

# Sự cần thiết của kiểu dữ liệu tự định nghĩa

User-defined  
data types

ThS.  
Trần Ngọc Bảo Duy



Từ khóa `typedef`

Tự định nghĩa một  
kiểu dữ liệu

Tại sao cần phải tự định nghĩa  
kiểu dữ liệu?

Cấu trúc trong C/C++

Sử dụng kiểu dữ liệu tự định  
nghĩa

## Giải pháp

GOM tất cả các dữ liệu thành phần có quan hệ với nhau thành một khối.

- Luôn luôn cấp phát CÙNG NHAU, LIÊN TỤC trong bộ nhớ
- Luôn luôn được huỷ khỏi bộ nhớ CÙNG NHAU
- Các dữ liệu thành phần có thể truy xuất độc lập, thông qua tên gọi của nó.

Đó là **cấu trúc** (`struct`) trong C và **lớp** (`class`) trong C++.

# Cấu trúc trong C/C++

User-defined  
data types

ThS.  
Trần Ngọc Bảo Duy



Từ khóa `typedef`

Tự định nghĩa một  
kiểu dữ liệu

Tại sao cần phải tự định nghĩa  
kiểu dữ liệu?

Cấu trúc trong C/C++

Sử dụng kiểu dữ liệu tự định  
nghĩa

## Định nghĩa

**Cấu trúc** là một kiểu dữ liệu mô tả một tổ hợp của các kiểu dữ liệu thành phần khác. Các kiểu dữ liệu thành phần có thể có cùng kiểu hay khác kiểu, thậm chí là một kiểu cấu trúc khác.

# Cấu trúc trong C/C++

User-defined  
data types

ThS.  
Trần Ngọc Bảo Duy



Từ khóa `typedef`

Tự định nghĩa một  
kiểu dữ liệu

Tại sao cần phải tự định nghĩa  
kiểu dữ liệu?

Cấu trúc trong C/C++

Sử dụng kiểu dữ liệu tự định  
nghĩa

## Định nghĩa

**Cấu trúc** là một kiểu dữ liệu mô tả một tổ hợp của các kiểu dữ liệu thành phần khác. Các kiểu dữ liệu thành phần có thể có cùng kiểu hay khác kiểu, thậm chí là một kiểu cấu trúc khác.

Cú pháp định nghĩa:

```
struct <struct_name> {  
    <datatype1> <member1>;  
    <datatype2> <member2>;  
    ...  
};
```

# Cấu trúc: Ví dụ

```
struct Student {  
    string id;  
    string name;  
    float gpa;  
};  
  
struct Point3D {  
    float x, y, z;  
};  
  
struct Vector3D {  
    float x, y, z;  
}  
  
struct Table {  
    int code;  
    float width, length, height;  
}
```

User-defined  
data types

ThS.  
Trần Ngọc Bảo Duy



Từ khóa typedef

Tự định nghĩa một  
kiểu dữ liệu

Tại sao cần phải tự định nghĩa  
kiểu dữ liệu?

Cấu trúc trong C/C++

Sử dụng kiểu dữ liệu tự định  
nghĩa

# Khai báo biến có kiểu dữ liệu tự định nghĩa

User-defined  
data types

ThS.  
Trần Ngọc Bảo Duy



Từ khóa `typedef`

Tự định nghĩa một  
kiểu dữ liệu

Tại sao cần phải tự định nghĩa  
kiểu dữ liệu?

Cấu trúc trong C/C++

Sử dụng kiểu dữ liệu tự định  
nghĩa

Trong C, khi sử dụng kiểu dữ liệu dữ liệu tự định nghĩa để khai báo biến thì phải thêm từ khóa `struct` trước tên kiểu dữ liệu, còn trong C++ thì không cần.

Ví dụ:

`struct` Student s1; trong C tương đương với:

Student s1; trong C++.

# Truy cập đến từng thành phần của kiểu

```
struct Student {  
    string id;  
    string name;  
    float gpa;  
};  
  
int main() {  
    Student s1;  
    Student s2 = {"001", "Nguyen_Van_A"};  
    Student s3 = {"002", "Tran_Thi_B", 9.5f};  
  
    cout << "ID:_" << s3.id;  
    cout << "Name:_" << s3.name;  
    cout << "GPA:_" << s3.gpa;  
}
```

User-defined  
data types

ThS.  
Trần Ngọc Bảo Duy



Từ khóa typedef

Tự định nghĩa một  
kiểu dữ liệu

Tại sao cần phải tự định nghĩa  
kiểu dữ liệu?

Cấu trúc trong C/C++

Sử dụng kiểu dữ liệu tự định  
nghĩa

# Truy cập đến từng thành phần của kiểu

```
struct Student {  
    string id;  
    string name;  
    float gpa;  
};  
  
int main() {  
    Student s1;  
    Student s2 = {"001", "Nguyen_Van_A"};  
    Student s3 = {"002", "Tran_Thi_B", 9.5f};  
  
    cout << "ID:_" << s3.id;  
    cout << "Name:_" << s3.name;  
    cout << "GPA:_" << s3.gpa;  
}
```

Quy tắc truy xuất dữ liệu thành phần thông qua tên gọi:

<Tên biến>.<Tên thành phần>

User-defined  
data types

ThS.  
Trần Ngọc Bảo Duy



Từ khóa typedef

Tự định nghĩa một  
kiểu dữ liệu

Tại sao cần phải tự định nghĩa  
kiểu dữ liệu?

Cấu trúc trong C/C++

Sử dụng kiểu dữ liệu tự định  
nghĩa