

# CHƯƠNG 2: ĐẠI CƯƠNG VỀ KIỂU DỮ LIỆU

---

# Nội dung

- Đối tượng dữ liệu - Kiểu dữ liệu
- Đặc tả và hiện thực
- Khai báo - Kiểm tra kiểu
- Gán - Khởi động trị
- Ràng buộc

# Đối tượng dữ liệu

- Một đối tượng dữ liệu (đtdl) là nơi chứa giá trị dữ liệu.

`p = (int *) malloc(sizeof(int))`



90

0 1 0 1 1 0 1 0

`*p = 90`



# Đối tượng dữ liệu

- Thuộc tính
  - Tên
  - Kiểu dữ liệu
  - Thời gian sống
  - Cơ bản - Cấu trúc
- Biến: đtdl có tên và giá trị dữ liệu thay đổi trong thời gian sống
  - `var no: integer;`
- Hằng: đtdl có tên và giá trị dữ liệu không đổi trong thời gian sống
  - Hằng theo ký tự: tên  $\leftrightarrow$  hằng “This is a string”
  - Hằng được định nghĩa: `const ss = 100;`

# Kiểu dữ liệu

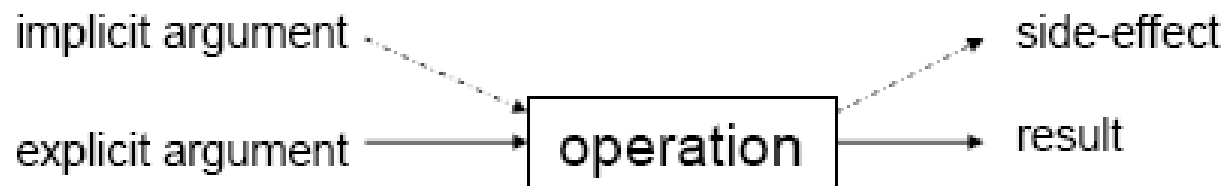
- Kiểu dữ liệu là một lớp các đtdl có cùng một số tính chất.
- Kiểu dữ liệu cơ bản (elementary) - cấu trúc (structured)
- Kiểu dữ liệu nguyên thủy (primitive) - do người lập trình định nghĩa (programmer-defined)
- Mô tả kiểu dữ liệu
  - Đặc tả
  - Hiện thực

# Đặc tả kiểu dữ liệu

- Thuộc tính (attribute)
  - type arr = array [1..2] of integer;
- Các giá trị (value)
  - <1,3>, <2,5>...
- Các tác vụ (operation)
  - A[1], khởi tạo dãy, +, -, \* ...
  - Operation-name: argument type × argument type × ...  
→ result type × result type × ...
  - >: integer × integer → boolean

# Vấn đề khi đặc tả tác vụ

- Không xác định trên 1 tập đối số
  - $N = \{\text{MIN\_INT}, \dots, \text{MAX\_INT}\}$
  - $+: N \times N \rightarrow ?$
- Các đối số ngầm (implicit argument)
- Hiệu ứng lề (side-effect)



- Tính tự sửa đổi (self-modification)

# Hiện thực kiểu dữ liệu

- Dạng lưu trữ (storage representation)
  - Thuộc tính không được lưu trữ - được xác định trong thời gian dịch
  - Thuộc tính được lưu trữ trong phần mô tả (descriptor)

01011001
----------

Int	01011001
-----	----------

- Tác vụ
  - Tác vụ phần cứng
  - Chương trình con
  - Đoạn mã trực tiếp (inline code)



# Khai báo

- Khai báo (declaration) là 1 phát biểu cung cấp thông tin về đtđl cho chương trình dịch
  - Tên, số lượng, kiểu, giá trị
  - Thời gian sống
  - Dạng lưu trữ
  - Số lượng, thứ tự và kiểu của các đối số; kết quả của mỗi tác vụ
- Khai báo bằng tên kiểu hoặc đặc tả kiểu
  - `Var i : integer; // i có kiểu integer`
  - `Var arr : array [1..10] of integer; // arr có kiểu vô danh`
- Khai báo tường minh - Khai báo ngầm

# Mục đích khai báo

- Chọn dạng lưu trữ
  - Tiết kiệm bộ nhớ
  - Tăng tốc độ thực thi
- Quản lý bộ nhớ hiệu quả
  - Cấp phát trên stack
  - Cấp phát và thu hồi bộ nhớ đơn giản
- Các tác vụ chung
  - Vd: tác vụ + có thể áp dụng trên nhiều kiểu dữ liệu khác nhau
- Cho phép chương trình dịch sinh mã thích hợp
  - Kiểm tra kiểu

# Kiểm tra kiểu

- Kiểm tra mỗi tác vụ có nhận được số lượng và kiểu các đối số phù hợp?
- Phân loại
  - Kiểm tra kiểu tĩnh
  - Kiểm tra kiểu động
- Chuyển đổi kiểu
- Tương hợp kiểu

# Kiểm tra kiểu động

- Thực hiện trong thời gian thực thi
- Mỗi đtdl phải lưu trữ đuôi kiểu (type tag)
- Ưu
  - Không cần khai báo
  - Kiểu có thể thay đổi
  - Có thể kiểm tra những trường hợp phức tạp
- Nhược
  - Người lập trình khó phát hiện lỗi
  - Tốn vùng nhớ
  - Tăng chi phí khi thực thi

# Kiểm tra kiểu tĩnh

- Thực hiện trong thời gian dịch
- Dựa vào thông tin được khai báo
  - Số lượng, thứ tự và kiểu của đối số và kết quả
  - Tên và kiểu dữ liệu của biến
  - Kiểu dữ liệu của hằng

# Chuyển đổi kiểu

- Chuyển đổi kiểu (type conversion) là tác vụ chuyển 1 đtdl thuộc kiểu này sang kiểu khác
  - Sử dụng các hàm có sẵn (built-in function)
  - Bằng áp đặt kiểu (type coercion)
- Widening
  - *short int*  $\rightarrow$  *long int*
- Narrowing
  - $1.0 \rightarrow 1$
  - $9+10/3 \rightarrow 2.33333... ?$

# Tương hợp kiểu

- Hai kiểu T1 và T2 là tương hợp kiểu (type compatibility) nếu đtdl kiểu T1 có thể xuất hiện ở vị trí của đtdl kiểu T2 và ngược lại
- Tương đương tên
- Tương đương cấu trúc

```
type T1 = array[1..20] of integer;
```

```
T2 = array[1..20] of integer;
```

```
var A: T1;
```

```
B: T2;
```

```
C: T1;
```

# Tác vụ gán

- Làm thay đổi giá trị của đtdl
- Gán tường minh hoặc gán ngầm (nhập trị)
- Đặc tả:

assignment:  $\text{type} \times \text{type} \rightarrow \text{void}$

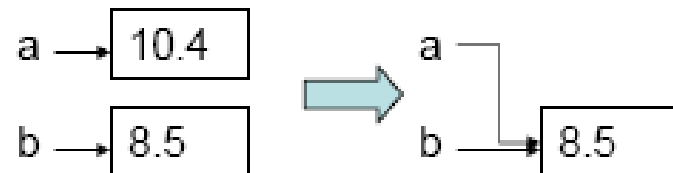
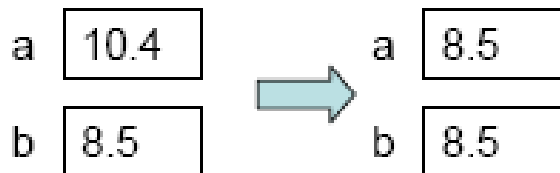
$a := b + c$

$d := x + a$

assignment:  $\text{type} \times \text{type} \rightarrow \text{type}$

$d = x + (a = b + c)$

- Ngữ nghĩa  $a := b$





# Tác vụ khởi động trị

- Là tác vụ gán giá trị đầu tiên cho đtdl
- Tác vụ khởi động trị tường minh qua tác vụ gán
- Tác vụ khởi động trị ngầm
  - Định sẵn

```
var a: integer;
```

- Khi khai báo biến

```
int a = 10;
```

# Ràng buộc

- Sự ràng buộc (binding) một phần tử chương trình vào 1 tính chất nào đó là sự lựa chọn tính chất đó từ 1 tập các tính chất có thể có của phần tử chương trình  
`var a: integer;`
- Thời gian ràng buộc
  - Thời gian định nghĩa ngôn ngữ
  - Thời gian hiện thực ngôn ngữ
  - Thời gian dịch
  - Thời gian thực thi