Căn bản về ngôn ngữ Java

GV: ThS. Phan Nguyệt Minh

minhpn@uit.edu.vn

http://courses.uit.edu.vn

Nội dung

- Biến & Hằng
- Kiểu dữ liệu (kiểu cơ sở, kiểu tham chiếu)
- Toán tử, biểu thức
- Các cấu trúc điều khiển (chọn, rẽ nhánh, lặp)
- Lớp bao kiểu cơ sở
- Phương thức và cách sử dụng
- Một số ví dụ minh họa

Biến

- Biến là một vùng nhớ lưu các giá trị của chương trình
- Mỗi biến gắn với 1 kiểu dữ liệu và 1 định danh duy nhất là tên biến
- Tên biến phân biệt chữ hoa và chữ thường. Tên biến bắt đầu bằng 1 dấu _, $, hay 1 ký tự, không được bắt đầu bằng 1 ký số.
- **Khai báo**
  <kiểu dữ liệu> <tên biến>;
  <kiểu dữ liệu> <tên biến> = <giá trị>;
- **Gán giá trị**
  <tên biến> = <giá trị>;

- **Lưu ý**: trong java nếu lúc khai báo không khởi tạo giá trị cho biến thì nó sẽ nhận 1 giá trị mặc định. Mỗi kiểu dữ liệu có 1 kiểu dữ liệu mặc định khác nhau.

Hằng

- Là một giá trị bất biến trong chương trình
- Tên đặt theo qui ước như tên biến
- Được khai báo dùng từ khóa **final,** và thường dùng tiếp vĩ ngữ đối với các hằng số (l, L, d, D, f, F)
- Ví dụ:

  final int x = 10; // khai báo hằng số nguyên x = 10

  final long y = 20L; // khai báo hằng số long y = 20
- Hằng ký tự: đặt giữa cặp nháy đơn ‘’
- Hằng chuỗi: là một dãy ký tự đặt giữa cặp nháy đôi “”

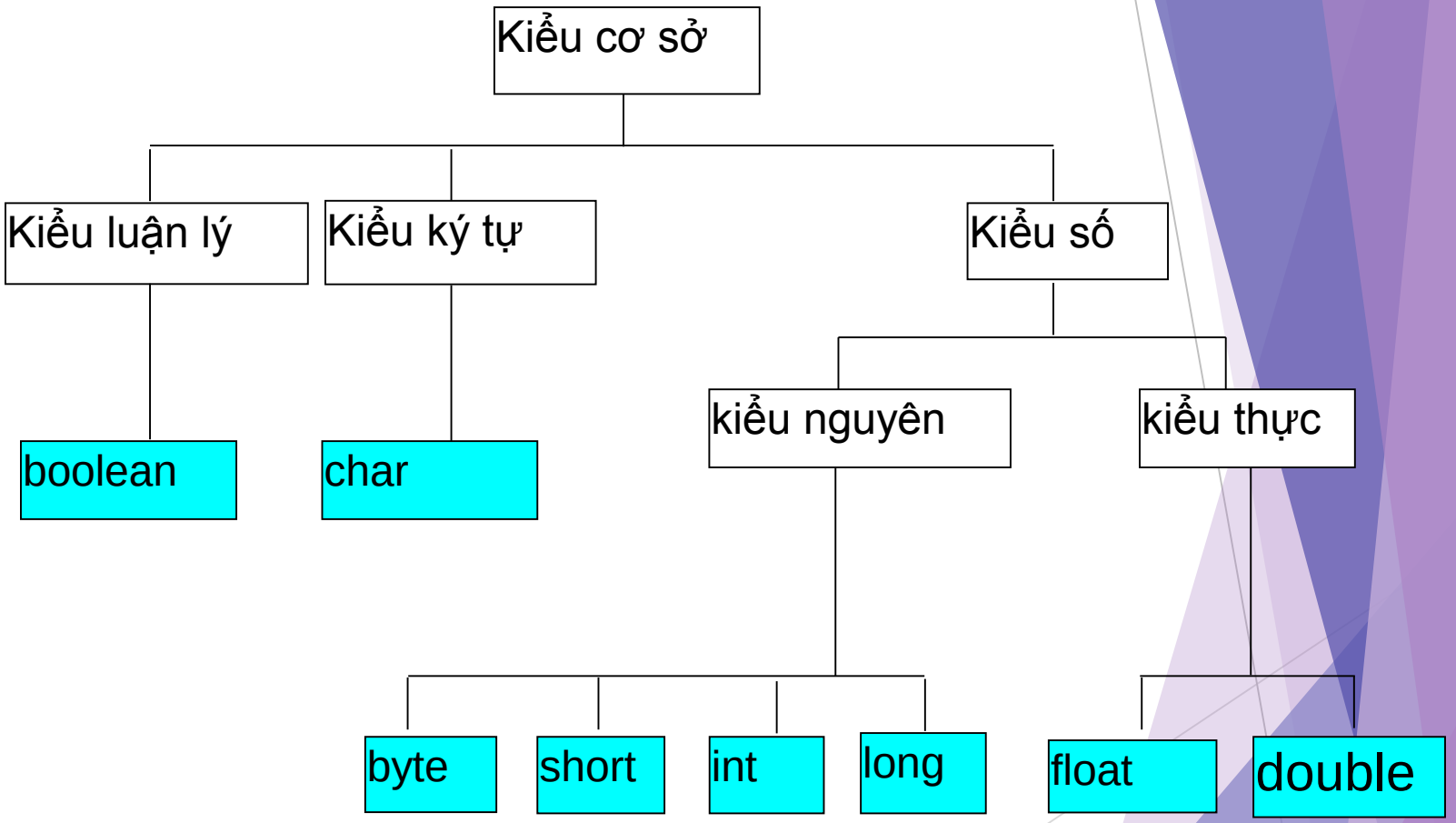
Hằng ký tự đặc biệt

Ký tự	Ý nghĩa
\b	Xóa lùi (BackSpace)
\t	Tab
\n	Xuống hàng
\r	Dấu enter
\”	Nháy kép
\’	Nháy đơn
\\	\
\f	Đẩy trang
\uxxxx	Ký tự unicode

Kiểu dữ liệu

- Kiểu dữ liệu cơ sở (primitive data type)
- Kiểu dữ liệu tham chiếu (reference data type)

Kiểu dữ liệu cơ sở

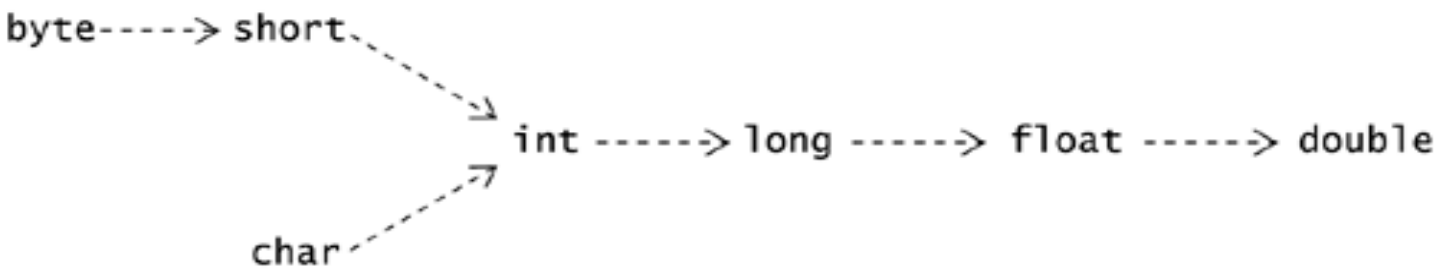


Kiểu dữ liệu cơ sở (tt)

Kiểu	Kích thước (bits)	Giá trị	Giá trị mặc định
boolean	[*Note:* The representation of a boolean is specific to the Java Virtual Machine on each computer platform.]	true và false	**false**
char	16	'\u0000' to '\uFFFF' (0 to 65535)	**null**
byte	8	–128 to +127 (-2^7 to $2^7 - 1$)	**0**
short	16	–32,768 to +32,767 (-2^{15} to $2^{15} - 1$)	**0**
int	32	–2,147,483,648 to +2,147,483,647 (-2^{31} to $2^{31} - 1$)	**0**
long	64	–9,223,372,036,854,775,808 to +9,223,372,036,854,775,807 (–263 to 263 – 1)	**0l**
float	32	1.4012984632481707e–45 to 3.4028234663852886E+38	**0.0f**
double	64	4.94065645841246544e–324 to 1.7976931348623157E+308	**0.0d**

Kiểu dữ liệu cơ sở (tt)

- **Chuyển đổi kiểu dữ liệu**: khi có sự không tương thích về kiểu dữ liệu (gán, tính toán biểu thức, truyền đối số gọi phương thức)
  - ✓ Chuyển kiểu hẹp (lớn → nhỏ): ***cần ép kiểu***
    <tên biến 2> = (kiểu dữ liệu) <tên biến 1>;
  - ✓ Chuyển kiểu rộng (nhỏ → lớn): ***tự động chuyển***



Kiểu dữ liệu cơ sở (tt)

- **Lưu ý**

1. *Không thể chuyển đổi giữa kiểu boolean với int và ngược lại.*
2. ***Nếu** 1 toán hạng kiểu **double** thì*
   *"Toán hạng kia chuyển thành **double**"*
   ***Nếu** 1 toán hạng kiểu **float** thì*
   *"Toán hạng kia chuyển thành **float**"*
   ***Nếu** 1 toán hạng kiểu **long** thì*
   *"Toán hạng kia chuyển thành **long**"*
   ***Ngược lại** "Tất cả chuyển thành **int** để tính toán"*

Kiểu dữ liệu cơ sở (tt)

- **Ví dụ minh họa**

1. byte x = 5;
2. byte y = 10;
3. byte z = x + y;
// Dòng lệnh thứ 3 báo lỗi chuyển kiểu cần sửa lại
// byte z = (byte) (x + y);

Kiểu dữ liệu tham chiếu

- **Kiểu mảng**
  - ✓ Mảng là tập hợp các phần tử có cùng tên và cùng kiểu dữ liệu.
  - ✓ Mỗi phần tử được truy xuất thông qua chỉ số
  - **Khai báo mảng**

<kiểu dữ liệu>[] <tên mảng>; // mảng 1 chiều

<kiểu dữ liệu> <tên mảng>[]; // mảng 1 chiều

<kiểu dữ liệu>[][] <tên mảng>; // mảng 2 chiều

<kiểu dữ liệu> <tên mảng>[][]; // mảng 2 chiều

Kiểu dữ liệu tham chiếu (tt)

- **Khởi tạo**

int arrInt[] = {1, 2, 3};

char arrChar[] = {'a', 'b', 'c'};

String arrString[] = {"ABC", "EFG", "GHI"};

- **Cấp phát & truy cập mảng**

*int arrInt = **new** int[100];*

int arrInt[100]; // Khai báo này trong Java sẽ bị báo lỗi.

Chỉ số mảng ***n*** phần tử: từ ***0*** đến ***n-1***

Kiểu dữ liệu tham chiếu (tt)

- **Kiểu đối tượng**
  - **Khai báo đối tượng**

    <Kiểu đối tượng> <biến ĐT>;
  - **Khởi tạo đối tượng**

    <Kiểu đối tượng> <biến ĐT> = ***new*** *<Kiểu đối tượng>;*
  - **Truy xuất thành phần đối tượng**

    <biến ĐT>.<thuộc tính>

    <biến ĐT>.<phương thức>

Toán tử, biểu thức

- Toán tử số học

Toán tử	Ý nghĩa
+	Cộng
-	Trừ
*	Nhân
/	Chia nguyên
%	Chia dư
++	Tăng 1
--	Giảm 1

Toán tử, biểu thức (tt)

- Phép toán trên bit

Toán tử	Ý nghĩa
&	AND
|	OR
^	XOR
<<	Dịch trái
>>	Dịch phải
~	Bù bit

Toán tử, biểu thức (tt)

- Toán tử quan hệ và logic

Toán tử	Ý nghĩa
= =	So sánh bằng
!=	So sánh khác
>	So sánh lớn hơn
<	So sánh nhỏ hơn
>=	So sánh lớn hơn hay bằng
<=	So sánh nhỏ hơn hay bằng
||	OR (biểu thức logic)
&&	AND (biểu thức logic)
!	NOT (biểu thức logic)

Toán tử, biểu thức (tt)

- Toán tử gán

Toán tử	Ví dụ	Ý nghĩa
=	a = b	gán a = b
+=	a += 5	a = a + 5
-=	b -= 10	b = b – 10
*=	c *= 3	c = c * 3
/=	d /= 2	d = d/2
%=	e %= 4	e = e % 4

Toán tử, biểu thức (tt)

- **Toán tử điều kiện**

Cú pháp: *<điều kiện> ? <biểu thức 1> : < biểu thức 2>*

- ***Ví dụ***:
  - *int x = 10;*
  - *int y = 20;*
  - *int Z = (x<y) ? 30 : 40;*
  - *// Kết quả z = 30 do biểu thức (x < y) là đúng.*

Cấu trúc điều khiển

- **Cấu trúc *if … else***
  - **Dạng 1:**

```
if (<điều_kiện>) {
<khối_lệnh>;
}
```

  - **Dạng 2:**

```
if (<điều_kiện>) {
<khối _lệnh1>;
}
else {
<khối _lệnh2>;
}
```

Cấu trúc điều khiển (tt)

- **Cấu trúc *switch … case***
  - *switch (<biến>)*

```
{   case <giátrị_1>:
            <khối_lệnh_1>;
            break;
        ….
        case <giátrị_n>:
            <khối_lệnh_n>;
            break;
        default:
            <khối lệnh default>;
}
```

Cấu trúc điều khiển (tt)

- **Cấu trúc lặp**
  - **Dạng 1:**
```
while (<điều_kiện_lặp>) {
        <khối _lệnh>;
    }
```
  - **Dạng 2:**
```
do {
    <khối_lệnh>;
} while (điều_kiện);
```
  - **Dạng 3:**
```
for (khởi_tạo_biến_đếm;đk_lặp;tăng_biến) {
    <khối _lệnh>;
    }
```

Cấu trúc điều khiển (tt)

- **Cấu trúc lệnh nhảy jump:** dùng kết hợp nhãn (label) với từ khóa ***break*** và ***continue*** để thay thế cho lệnh ***goto*** (trong C).

- ***Ví dụ:***

```
label:
for (…) {
   for (…) {
     if (<biểu thức điều kiện>)
          break label;
     else
          continue label;
   }
}
```

Lớp bao kiểu dữ liệu cơ sở

Data type	Wrapper Class (java.lang.*)	Ghi chú
boolean	Boolean	- Gói (package): chứa nhóm nhiều class. - Ngoài các Wrapper Class, gói java.lang còn cung cấp các lớp nền tảng cho việc thiết kế ngôn ngữ java như: String, Math, …
byte	Byte	
short	Short	
char	Character	
int	Integer	
long	Long	
float	Float	
double	Double	

Q/A