

Bài 2

TOÁN TỬ, HÀM ĐỆ QUY, THAM CHIẾU, THAM TRỊ

Company: DEVPRO VIỆT NAM

Website: devpro.edu.vn

Design by Minh An

Nội dung

- **Toán tử**
Tất cả các toán tử của java
- **Hàm**
Định nghĩa, cách viết hàm, gọi hàm
- **Tham chiếu, tham trị**
Khái niệm, phân biệt tham chiếu, tham trị

Design by Minh An

Các toán tử trong java

- **Toán tử gán**
- **Toán tử số học**
- **Toán tử trên bit**
- **Toán tử quan hệ**
- **Toán tử logic**
- **Toán tử một ngôi, ba ngôi**

Design by Minh An

Các toán tử số học

- Các toán hạng của toán tử số học phải là kiểu số.
- Toán hạng kiểu boolean không thể được sử dụng, nhưng kiểu character thì được phép
- Các toán hạng được sử dụng trong các biểu thức tính toán.

Operator	Description
+	Addition - Returns the sum of the operands
-	Subtraction - Returns the difference of two operands
*	Multiplication - Returns the product of operands
/	Division - Returns the result of division operation
%	Remainder - Returns the remainder from a division operation

Design by Minh An

Các toán tử số học

- Ví dụ: Tính tổng, hiệu, tích, thương, phần nguyên, phần dư phép chia hai số nguyên a và b.

Design by Minh An

Các toán tử một ngôi

Toán tử một ngôi chỉ cần có một toán hạng

Operator	Description
+	Unary plus - Indicates a positive value
-	Unary minus - Negates an expression
++	Increment operator - Increments value of a variable by 1
--	Decrement operator - Decrements value of a variable by 1
!	Logical complement operator - Inverts a boolean value

Design by Minh An

Các toán tử một ngôi

Ví dụ: Thực thi đoạn chương trình sau:

```
int x = 5, y = 5, a, b, k, m;  
x++; y--;           //Display x, y  
a = x++;           //Display x, a  
b = y--;           //Display b, y  
a = ++x;           //Display x, a  
b = --y;           //Display x, y  
k = x++ + --y;     //Display x, y, k  
m = ++x + --y;     //Display x, y, m
```

Design by Minh An

Các toán tử quan hệ (so sánh)



- Các toán tử quan hệ kiểm tra mối quan hệ giữa hai toán hạng.
- Kết quả của biểu thức sử dụng các toán tử quan hệ là dạng boolean (true hoặc false).
- Các toán tử quan hệ được sử dụng trong các cấu trúc điều khiển.

Operator	Description
==	Equal to - Checks equality of two numbers
!=	Not Equal to - Checks inequality of two values
>	Greater than - Checks if value on left is greater than the value on the right
<	Less than - Checks if the value on the left is lesser than the value on the right
>=	Greater than or equal to - Checks if the value on the left is greater than or equal to the value on the right
<=	Less than or equal to - Checks if the value on the left is less than or equal to the value on the right

Design by Minh An

Các toán tử quan hệ (so sánh)



- Thực thi đoạn chương trình sau

```
int a = 6, b = 4;  
float x = 10;  
//Display a == b  
//Display a < b  
//Display a <= b  
//Display a > b;  
//Display a >= b  
//Display a != b  
//Display x > a + b  
//Display x == a + b  
//Display x != a + b
```

Design by Minh An

Các toán tử logic



- Thể hiện mối quan hệ logic giữa các biểu thức điều kiện.

Operator	Description
&&	Conditional AND - Returns true only if both the expressions are true
	Conditional OR - Returns true if either expression is true or both the expressions are true

Design by Minh An

Các toán tử logic



- Thực thi đoạn chương trình sau.

```
int a = 6, b = 4;  
float x = 10;  
//Display x > a && x > b  
//Display x > a && x < b  
//Display x < a && x > b  
//Display x < a && x < b  
//Display x > a || x > b  
//Display x > a || x < b  
//Display x < a || x > b  
//Display x < a || x < b  
//Display !(x < a)  
//Display !(x > a)
```

Design by Minh An

Toán tử ba ngôi



- Toán tử ba ngôi, ký hiệu là (? :). Nó nhận vào ba toán hạng
- Cú pháp:

expression1 ? expression2 : expression3;

- Ví dụ: Tìm giá trị lớn nhất trong 2 số nguyên

Design by Minh An

Các toán tử thao tác bit



- Các toán tử tương tác trên bit (bitwise) và các toán tử dịch bit.
 - Các toán tử bitwise cho phép thao tác trên các bit trong toán hạng.
 - Các toán tử bit shift được sử dụng để dịch chuyển tất cả các bit trong toán hạng sang trái hoặc sang phải n lần.

Operator	Description
&	Bitwise AND - compares two bits and generates a result of 1 if both bits are 1; otherwise, it returns 0.
	Bitwise OR - compares two bits and generates a result of 1 if the bits are complementary; otherwise, it returns 0.
^	Exclusive OR - compares two bits and generates a result of 1 if either or both bits are 1; otherwise, it returns 0.
~	Complement operator - used to invert all of the bits of the operand.
>>	Shift Right operator - Moves all bits in a number to the right by one position retaining the sign of negative numbers.
<<	Shift Left operator - Moves all the bits in a number to the left by the specified position.

Design by Minh An

Toán tử thao tác bit



a = 0011 1100

b = 0000 1101

a & b = 0000 1100

a | b = 0011 1101

a ^ b = 0011 0001

~a = 1100 0011

~ (bitwise compliment)	Binary Ones Complement Operator is unary and has the effect of 'flipping' bits.	(~A) will give -61 which is 1100 0011 in 2's complement form due to a signed binary number.
<< (left shift)	Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right operand.	A << 2 will give 240 which is 1111 0000
>> (right shift)	Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.	A >> 2 will give 15 which is 1111
>>> (zero fill right shift)	Shift right zero fill operator. The left operands value is moved right by the number of bits specified by the right operand and shifted values are filled up with zeros.	A >>> 2 will give 15 which is 0000 1111

Design by Minh An

Toán tử gán



= gán giá trị của biểu thức bên phải cho biến ở bên trái

Các phép gán kết hợp:

`+=` , `-=` , `*=` , `/=` , `%=`
`^=` , `&=` , `|=` , `>>=` , `<<=`

Lấy giá trị toán hạng bên trái thực thi phép toán với toán hạng bên phải rồi gán giá trị trở lại cho toán hạng bên trái:

```
int x = 5;  
x += 10;    // x = 15
```

Design by Minh An

Thứ tự ưu tiên của các toán tử



Postfix	() [] . (toán tử dot)	Trái sang phải
Unary	++ -- ! ~	Phải sang trái
Tính nhân	* / %	Trái sang phải
Tính cộng	+ -	Trái sang phải
Dịch chuyển	>> >>> <<	Trái sang phải
Quan hệ	> >= < <=	Trái sang phải
Cân bằng	== !=	Trái sang phải
Phép AND bit	&	Trái sang phải
Phép XOR bit	^	Trái sang phải
Phép OR bit		Trái sang phải
Phép AND logic	&&	Trái sang phải
Phép OR logic		Trái sang phải
Điều kiện	?:	Phải sang trái
Gán	= += -= *= /= %= >>= <<= &= ^= =	Phải sang trái
Dấu phẩy	,	Trái sang phải

Design by Minh An

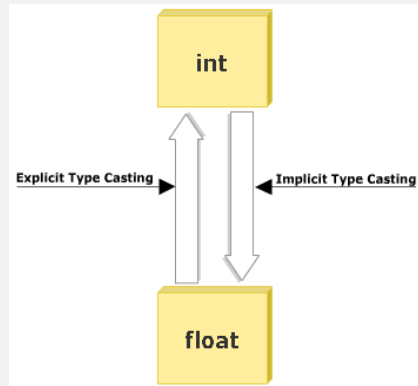
Chuyển đổi kiểu dữ liệu



- Khi chuyển đổi kiểu dữ liệu, một kiểu dữ liệu này sẽ được chuyển sang thành một kiểu dữ liệu khác.
- Có hai loại chuyển đổi kiểu là:

Implicit casting - chuyển đổi ngầm định

Explicit casting - chuyển đổi tường minh



Design by Minh An

Chuyển kiểu ngầm định



- Khi gán một kiểu dữ liệu này cho một kiểu dữ liệu khác thì sự chuyển đổi kiểu dữ liệu tự động (ngầm định) sẽ xảy ra nếu hai điều kiện sau được thỏa mãn:
 - Hai kiểu dữ liệu phải tương thích với nhau
 - Kiểu dữ liệu đích phải rộng hơn kiểu dữ liệu nguồn.
- Các kiểu dữ liệu nguyên thủy dạng số có thể chuyển kiểu một cách ngầm định là:

byte -> short -> int -> long -> float -> double

Design by Minh An

Chuyển kiểu tường minh



- Để chuyển đổi một kiểu dữ liệu độ chính xác cao xuống một kiểu dữ liệu có độ chính xác thấp hơn thì ta phải dùng explicit type casting hay ép kiểu
- Ví dụ:

```
float a = 21.23456f;  
int b = (int) a + 5;
```

Design by Minh An

Định nghĩa hàm



- Hàm là cách gói gọn một đoạn code thực hiện một tác vụ được định nghĩa cụ thể.
- Mỗi hàm có một tên (ngắn gọn và dễ nhớ), danh sách các đối biểu diễn dữ liệu mà hàm xử lý, giá trị trả về biểu diễn kết quả xử lý.
- Hàm có thể trả về giá trị, hoặc không trả về giá trị.
- Hàm có thể có đối hoặc không có đối.
- Sử dụng hàm giúp giảm bớt những đoạn code giống nhau, chương trình rõ ràng và dễ sửa lỗi hơn.

Design by Minh An

Định nghĩa hàm



Ví dụ:

Hàm không có đối, không trả về giá trị

```
public void printHello(){  
    System.out.println("Xin chào các bạn");  
}
```

Hàm có đối, có trả về giá trị

```
public int sumTwoNumber(int a, int b){  
    return a + b;  
}
```

Hàm `sumTwoNumber` có hai đối kiểu `int` là `a` và `b`, hàm trả về một giá trị kiểu `int`.

Design by Minh An

Sử dụng hàm



Để sử dụng hàm:

- Gọi hàm thông qua tên hàm
- Truyền tham số cho hàm

Ví dụ

```
printHello();
```

```
int c = sumTwoNumber(15, 18);
```

- `printHello()` hay `sumTwoNumber(15, 18)` là các lời gọi hàm.
- Hàm `printHello()` không có đối nên trong lời gọi không truyền tham số.
- Hàm `sumTwoNumber(...)` có hai đối nên khi gọi hàm ta truyền hai tham số (15 và 18), giá trị hàm trả về được gán cho biến `c`.

Design by Minh An

Hàm đệ quy



Hàm đệ quy là hàm có lời gọi đến chính nó trong phần thân hàm

ví dụ tính giai thừa

```
static int giaiThua(int n){  
    if(n == 0)  
        return 1;  
    else return n * giaiThua(n - 1);  
}
```

Design by Minh An

Truyền tham số cho hàm



- Có hai loại tham số là tham trị và tham chiếu.
- Truyền tham trị:
 - Tham số truyền vào là các hằng giá trị, hoặc giá trị chứa trong các biến.
 - Nếu tham số là biến thì giá trị của biến sẽ được truyền vào hàm. Hàm không thể thay đổi giá trị của biến.
- Truyền tham chiếu:
 - Tham số truyền vào hàm phải là các biến
 - Địa chỉ của biến sẽ được truyền vào hàm.
 - Hàm có thể làm thay đổi giá trị của biến truyền vào.
- Mặc định trong java tất cả các kiểu dữ liệu cơ bản đều là tham trị
int, float, double, long, short, char, boolean
- Tất cả các kiểu dữ liệu cấu trúc đều là tham chiếu
Object, Interface, Array, Collection.

Design by Minh An

Bài tập



1. Cài đặt chương trình có hàm tính diện tích hình thang.
2. Cài đặt chương trình có các hàm tính chu vi, diện tích tam giác.
3. Cài đặt chương trình có các hàm tính chu vi, diện tích hình tròn.
4. Cài đặt hàm đệ quy tìm ước chung lớn nhất của 2 số nguyên dương (theo thuật toán Euclid). Sử dụng hàm để tìm ước chung của 4 số nguyên dương.
5. Cài đặt hàm đệ quy tìm số Fibonacci thứ n.
6. Cài đặt chương trình có hàm cho phép nhập và trả về một số, hiển thị số.
7. Cài đặt chương trình có hàm cho phép nhập và trả về một chuỗi, hiển thị chuỗi.
8. Cài đặt chương trình có hàm cho phép nhập và trả về thông tin của 1 học sinh, hàm hiển thị thông tin của học sinh.
9. Cài đặt chương trình có hàm tìm max của ba số, sử dụng toán tử 3 ngôi.

Design by Minh An