

Bài 2

TOÁN TỬ, HÀM TRONG JAVA

Company: DEVPRO VIỆT NAM

Website: devpro.edu.vn

Design by Minh An

Nội dung

- Toán tử
- Chuyển đổi kiểu dữ liệu
- Hàm
- Truyền tham số cho hàm

Design by Minh An

1. Toán tử trong java

- Toán tử số học
- Toán tử quan hệ
- Toán tử logic
- Toán tử ba ngôi

Design by Minh An

1.1. Toán tử số học

- Toán tử hai ngôi: Gồm các phép toán cộng (+), trừ (-), nhân (*), chia (/), chia lấy dư (%).
- Ví dụ: Nhập hai số nguyên a và b. Tính tổng (a + b), hiệu (a - b), tích (a * b), thương (a / b), phần nguyên (a / b), phần dư (a % b) phép chia hai số nguyên a và b.
- Toán tử một ngôi: Là các toán tử thao tác với 1 toán hạng gồm toán tử đổi dấu (-), toán tử tăng (++), toán tử giảm (--), toán tử phủ định (!).

Ví dụ: Thực thi đoạn chương trình sau:

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

Design by Minh An

1.2. Toán tử quan hệ (so sánh)



- So sánh hai giá trị (số, ký tự, chuỗi) và trả về một giá trị logic (true hoặc false).
- Các toán tử quan hệ dùng để xây dựng biểu thức kiểm tra điều kiện trong các cấu trúc điều khiển.

Ký hiệu	Mô tả	Ví dụ	Kết quả
<	Nhỏ hơn	2 < 3	true
<=	Nhỏ hơn hoặc bằng	2 <= 3	true
>	Lớn hơn	2 > 3	false
>=	Lớn hơn hoặc bằng	2 >= 2	true
==	Bằng	2 == 3	false
!=	Khác	2 != 3	true

Design by Minh An

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

1.3. Toán tử logic



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

Ký hiệu	Mô tả
!	Phủ định
&&	Và
	Hoặc

A	B	!A	A && B	A B
true	true	false	true	true
true	false	false	false	true
false	true	true	false	true
false	false	true	false	false

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

1.4. 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 3 số nguyên

Design by Minh An

1.5. Toán tử gán



- Ký hiệu: =
- Cú pháp: <Biến> = <Biểu_Thức>;
- Ý nghĩa: Gán giá trị biểu thức bên phải cho biến ở bên trái
- Một số 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:
- Ví dụ:

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

Design by Minh An

1.6. Thứ tự ưu tiên của các toán tử



- Toán tử một ngôi
- Toán tử hai ngôi
 - Toán tử số học
 - Toán tử quan hệ
 - Toán tử logic

Design by Minh An

2. Chuyển đổi kiểu dữ liệu



- Khi chuyển đổi kiểu dữ liệu, dữ liệu thuộc kiểu này được chuyển sang kiểu dữ liệu khác.
- Có hai loại chuyển đổi kiểu là:
 - Implicit casting - chuyển đổi ngầm định (tự động)
 - Explicit casting - chuyển đổi tường minh (ép kiểu)

Design by Minh An

2.1. Chuyển kiểu ngầm định



- Máy tính tự động chuyển kiểu dữ liệu của 1 trong hai toán hạng khi thực hiện một phép toán để thực hiện phép toán với hai toán hạng cùng kiểu.
- Dữ liệu kiểu "hẹp hơn" được chuyển sang kiểu "rộng hơn".
- Lưu ý với toán tử gán: Kiểu dữ liệu của biến phải "tương đương" với biểu thức và/hoặc "rộng hơn" biểu thức.
- Việc chuyển kiểu ngầm định được thực hiện:
byte -> short -> int -> long -> float -> double

Design by Minh An

2.2. Chuyển kiểu tường minh



- Ép kiểu (casting).
- Để thực hiện phép chia (/) hai toán hạng nguyên với kết quả là số thực.
Ví dụ:

```
int a = 12, b = 5;  
float c = (float)a / b; //để được c = 2.4
```
- Để chuyển đổi một dữ liệu có kiểu rộng hơn khi gán cho biến có kiểu hẹp hơn;
Ví dụ:

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

Design by Minh An

3. Hàm trong java

- Định nghĩa hàm
- Sử dụng hàm
- Hàm đệ quy
- Truyền tham số

Design by Minh An

3.1. 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 vào (dữ liệu được hàm xử lý), dữ liệu hàm trả về (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

3.1. Định nghĩa hàm



- Cú pháp

```
public static <kiểu trả về> tên_Hàm([ds đối]){  
    //Thân hàm  
}
```

- Ví dụ 1

```
public static float dienTich(float a, float b, float h){  
    float s = (a + b) * h / 2;  
    return s;  
}
```

Design by Minh An

Định nghĩa hàm



Ví dụ 2:

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

3.2. 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

3.3. Hàm đệ quy



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

Ví dụ: tính giai thừa n

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

Design by Minh An

3.4. 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.
 - Các đối của hàm nhận giá trị của các tham số tương ứng
- 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.

Design by Minh An

Truyền tham số cho hàm



- Mặc định trong java khi truyền tham số với các đối của hàm thuộc tất cả các kiểu dữ liệu cơ bản đều là truyền tham trị.
int, float, double, long, short, char, boolean
- Khi các đối của hàm là các kiểu dữ liệu cấu trúc thì việc truyền tham số là truyền tham chiếu
Object, Interface, Array, Collection.

Design by Minh An

Bài tập



1. 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.
2. Cài đặt hàm đệ quy tìm số Fibonacci thứ n. Sử dụng hàm tìm và hiển thị ra màn hình dãy n số fibonacci đầu tiên
3. Thông tin về học sinh gồm: Mã học sinh, họ đệm, tên, ngày sinh, giới tính, điểm trung bình các môn học: toán, văn, lý, hóa, sử, địa, sinh, tiếng anh, giáo dục công dân.

Cài đặt chương trình gồm:

- Hàm cho phép nhập và trả về thông tin của 1 học sinh
- Hàm tính điểm trung bình học kỳ cho học sinh (điểm toán, văn tính hệ số 2, các môn còn lại hệ số 1).
- Hàm hiển thị thông tin của học sinh cùng với ĐTB học kỳ.
- Sử dụng các hàm để nhập và hiển thị thông tin của học sinh cùng với điểm trung bình học kỳ.

Design by Minh An

Bài tập



4. Cho các thông tin về các đối tượng như sau:
 - Thông tin về công dân gồm: số thẻ căn cước, họ đệm, tên, ngày sinh, giới tính, địa chỉ, điện thoại.
 - Thông tin về địa chỉ gồm: số nhà, tổ dân phố, phường, quận, thành phố.

Viết chương trình gồm:

- Hàm nhập vào thông tin địa chỉ.
- Hàm hiển thị thông tin địa chỉ
- Hàm nhập thông tin của công dân (sử dụng hàm nhập thông tin địa chỉ)
- Hàm hiển thị thông tin công dân (sử dụng hàm hiển thị thông tin địa chỉ).
- Sử dụng các hàm trên để nhập và hiển thị thông tin của 2 công dân.

Design by Minh An

Bài tập



5. Cho thông tin về các đối tượng như sau:

- Thông tin của sinh viên bao gồm các thuộc tính: Mã sinh viên, họ đệm, tên, ngày sinh và thông tin về 5 môn học mà sinh viên đã hoàn thành.
- Thông tin về môn học gồm: Mã môn học, tên môn học, hệ số môn học và điểm số của môn học.

Cài đặt chương trình gồm:

- Hàm nhập vào thông tin của một môn học.
- Hàm hiển thị thông tin của một môn học.
- Hàm nhập thông tin của một sinh viên (sử dụng hàm nhập thông tin môn học để nhập 5 môn học mà sinh viên đã hoàn thành).
- Hàm tính điểm trung bình của 5 môn học của một sinh viên theo điểm và hệ số của các môn học.
- Hiển thị thông tin của sinh viên ra màn hình (sử dụng hàm hiển thị thông tin môn học).

Design by Minh An

Bài tập



6. Cho thông tin về các đối tượng như sau:

- Thông tin về hàng hóa gồm: tên hàng, số lượng hàng, đơn giá.
- Thông tin về hóa đơn gồm: số hóa đơn, họ tên khách hàng, ngày mua hàng, danh sách chứa thông tin về 5 loại hàng hóa mà khách hàng chọn mua.

Viết chương trình gồm:

- Hàm nhập vào thông tin của một hàng hóa.
- Hàm để hiển thị thông tin của một hàng hóa sau khi đã tính thành tiền cho hàng hóa (thành tiền = số lượng * đơn giá).
- Hàm nhập vào thông tin của một hóa đơn (sử dụng hàm nhập thông tin hàng hóa).
- Hàm để hiển thị thông tin trên hóa đơn ra màn hình sau khi đã tính thành tiền cho mỗi loại hàng, tính tổng tiền hàng, tính tiền phải trả sau khi đã cộng thuế giá trị gia tăng (10%) – thông tin của mỗi loại hàng hóa được hiển thị trên 1 dòng (hàm này sử dụng hàm hiển thị thông tin hàng hóa).
- Sử dụng các hàm để nhập và hiển thị thông tin của một hóa đơn.

Design by Minh An