1. Tìm hiểu về các phép tính logic cơ bản (logical operator) của Java

2. Tìm hiểu về các phép tính so sánh của Java

3. Tìm hiểu về khối lệnh if, if-else, switch

Bài Làm

1

* Phép tính AND (‘&&’):
* khi sử dụng trong biểu thức, phép tính ‘&&’ trả về ‘true’ nếu cả hai toán hạng đều là ‘true’, ngược lại nó trả về ‘false’.

Ví dụ: boolean a = true;

boolean b = false;

boolean result = a && b; // result sẽ là false

* phép tính OR (‘||’);
* Phép tính || trả về true nếu ít nhất một trong hai toán hạng là ‘true’. Nó chỉ trả về ‘false’ nếu cả hai đều là ‘false’.

Ví dụ: boolean a = true;

boolean b = false;

boolean result = a || b; // result sẽ là true

* phép tính NOT (‘!’);
* Phép tính ! (phủ định) được sử dụng để đảo ngược giá trị của một biểu thức boolean. Nếu biểu thức là ‘true’, thì ! biểu thức sẽ trả về ‘false’, và ngược lại.

Ví dụ: boolean a = true;

boolean result = !a ; // result sẽ là false

2

* Phép So Sánh bằng (‘==’):
* phép so sánh bằng ‘==’ kiểm tra xem hai biểu thức hoặc biến có giá trị bằng nhau không.

Ví dụ: int a = 5;

int b = 5

boolean result = (a == b); // result sẽ là true

* Phép So sánh không bằng (‘!=’);
* Phép so sánh không bằng != kiểm tra xem hai biểu thức hoặc biến có giá trị khác nhau không. Nếu khác nhau, nó trả về ‘true’, ngược lại trả về ‘false’

Ví dụ: int a = 5;

int b = 7;

boolean result = (a != b); // result sẽ là true

* So sánh lớn hơn (‘>’):
* Phép so sánh lớn hơn > kiểm tra xem biểu thức hoặc biến bên trái có lớn hơn biểu thức hoặc biến bên phải không. Nếu đúng, nó trả về ‘true’, ngược lại trả về ‘false’.

Ví dụ: int a = 10;

int b = 5;

boolean result = (a > b) ; // result sẽ là true

* Phép So Sánh Nhỏ Hơn (‘<’);
* Phép so sánh nhỏ hơn < kiểm tra xem biểu thức hoặc biến bên trái có nhỏ hơn biểu thức hoặc biến bên phải không. Nếu đúng, nó trả về ‘true’, ngược lại trả về ‘false’.

Ví dụ: int a = 5;

int b = 10;

boolean result = (a<b); // result sẽ là true

* Phép so sánh lớn hơn hoặc bằng (‘>=’);
* Phép so sánh lớn hơn hoặc bằng ‘>=’ kiểm tra xem biểu thức hoặc biến bên trái có lớn hơn hoặc bằng biểu thức hoặc biến bên phải không. Nếu đúng, nó trả về ‘true’, ngược lại trả về ‘false’.

Ví Dụ: int a = 10;

int b = 5;

boolean result = (a >= b); // result sẽ là true

* Phép so sánh nhỏ hơn hoặc bằng(‘<=’);
* Phép so sánh nhỏ hơn hoặc bằng ‘<=’ kiểm tra xem biểu thức hoặc biến bên trái có nhỏ hơn hoặc bằng biểu thức hoặc biến bên phải không. Nếu đúng, nó trả về ‘true’, ngược lại trả về ‘false’.

Ví Dụ: int a = 5;

int b = 10;

boolean result = (a <= b); // result sẽ là true

3

* Khối lệnh ‘if’:
* Khối lệnh ‘if’ được sử dụng để kiểm tra một điều kiện và thực thi một khối mã chỉ khi điều kiện đó là ‘true’. Nếu điều kiện là ‘false’, thì khối mã bên trong ‘if’ sẽ bị bỏ qua.

Ví dụ: int age = 25;

if (age >= 18) { System.out.println(“bạn đã đủ tuổi để bỏ phiếu”); }

* Khối Lệnh ‘if-else’:Được sử dụng để thực thi một khối mã nếu điều kiện trong ‘if’ là ‘true’ và một khối mã khác nếu điều kiện là ‘false’.

Ví dụ: int age = 15;

if (age >= 18) {

System.out.println(“Bạn đã đủ tuổi để bỏ phiếu.’’);

} else {

System.out.println(“bạn chưa đủ tuổi bỏ phiếu.’’);

}

* Khối Lệnh ‘switch’:được sử dụng để thực hiện một loạt các lựa chọn dựa trên giá trị của biến hoặc biểu thức. Một switch có thể đi kèm với nhiều trường hợp (case) và một trường hợp mặc định (default)

Ví dụ: int day0fWeek = 2;

String dayName;

switch (day0fWeek) {

case 1:

dayName = “Chủ Nhật”;

break;

case 2:

dayName = “Thứ Hai’’;

break;

case 3:

dayName = “Thứ Ba”;

break;

// thêm các trường hợp khác nếu cần

default:

dayName = “không xác định’’;

}

System.out.println(“hôm nay là ‘’ + dayName);