



Lập trình Java – Câu lệnh điều khiển

Ths. Vũ Duy Khương

1

Các cấu trúc lựa chọn : if else, switch

2

Các cấu trúc lặp: while, do while, for

3

Từ khóa break và continue

4

Ví dụ và Bài tập

5

Q&A

Các lệnh lựa chọn

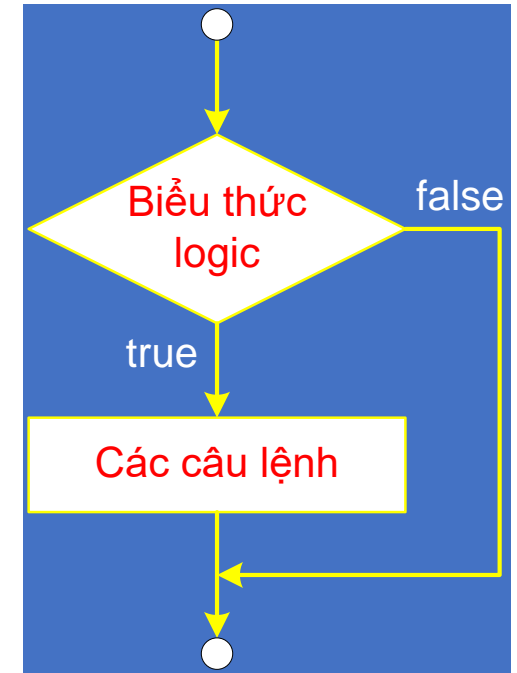
- Lệnh if
- Lệnh switch
- Toán tử điều kiện

Câu lệnh If

```
if (Biểu_thức_logic) {  
    các_câu_lệnh;  
}
```

Ví dụ:

```
if ((i > 0) && (i < 10)) {  
    System.out.println("i là một " + "số nguyên nằm giữa 0  
    và 10");  
}
```



Thận trọng

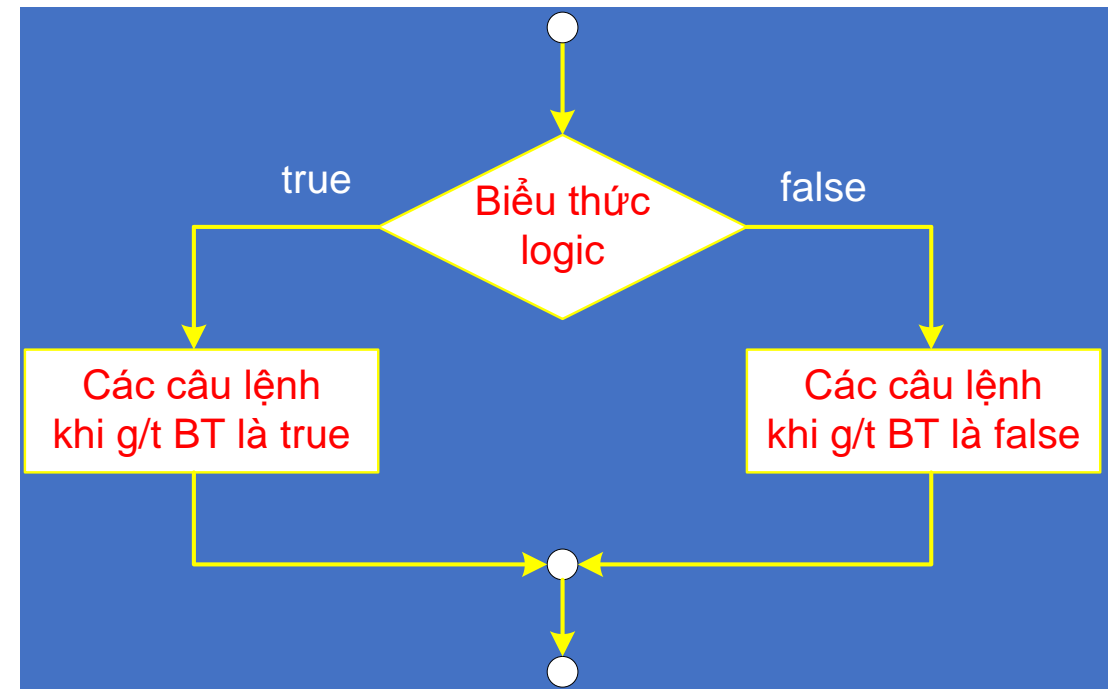
Lỗi phổ biến: thêm một dấu chấm phẩy ở cuối mệnh đề if.

```
if (radius >= 0);  
{  
    area = radius*radius*PI;  
    System.out.println("The area for the circle of radius " + radius + " is “  
                        + area);  
}
```

Lỗi này rất khó tìm, vì nó không phải là lỗi biên dịch hay lỗi chạy chương trình, nó là một lỗi logic.

Lệnh if...else

```
if (Biểu_thức_logic) {  
    Các_câu_lệnh_ứng_BT_đúng;  
}else {  
    Các_câu_lệnh_ứng_BT_sai;  
}
```



Ví dụ if...else

```
if (bankinh >= 0) {  
    dientich = bankinh*bankinh*PI;  
  
    System.out.println("Dien tich hinh tron co ban kinh " + bankinh + " la  
    " + dientich);  
  
}else {  
    System.out.println("Du lieu khong hop le!");  
}
```

Nhiều lệnh if luân phiên

```
if (score >= 90)
    grade = 'A';
else
    if (score >= 80)
        grade = 'B';
    else
        if (score >= 70)
            grade = 'C';
        else
            if (score >= 60)
                grade = 'D';
            else
                grade = 'F';
```

```
if (score >= 90)
    grade = 'A';
else if (score >= 80)
    grade = 'B';
else if (score >= 70)
    grade = 'C';
else if (score >= 60)
    grade = 'D';
else
    grade = 'F';
```


Chú ý 1

Mệnh đề else gắn với mệnh đề if gần nhất trong cùng một khối.

Ví dụ, đoạn lệnh sau:

```
int i = 1; int j = 2; int k = 3;
if (i > j)
    if (i > k)
        System.out.println("A");
else
    System.out.println("B");
```

là tương đương với:

```
int i = 1; int j = 2; int k = 3;
if (i > j)
    if (i > k)
        System.out.println("A");
else
    System.out.println("B");
```

Chú ý 2

Đoạn lệnh trước sẽ không in ra gì cả. Để bắt mệnh đề else gắn với mệnh đề if đầu tiên, bạn phải thêm một cặp ngoặc nhọn:

```
int i = 1;
int j = 2;
int k = 3;
    if (i > j) {
        if (i > k)
            System.out.println("A");
    }
else
    System.out.println("B");
```

Đoạn lệnh trên sẽ in ra ký tự B.

Chú ý 3

```
if (n % 2 == 0)
    iseven = true;
else
    iseven = false;
```

tương
đương

```
boolean iseven
    = (n % 2 == 0)
```

```
if (n == true)
    system.out.println
        ("So chan");
```

tương
đương

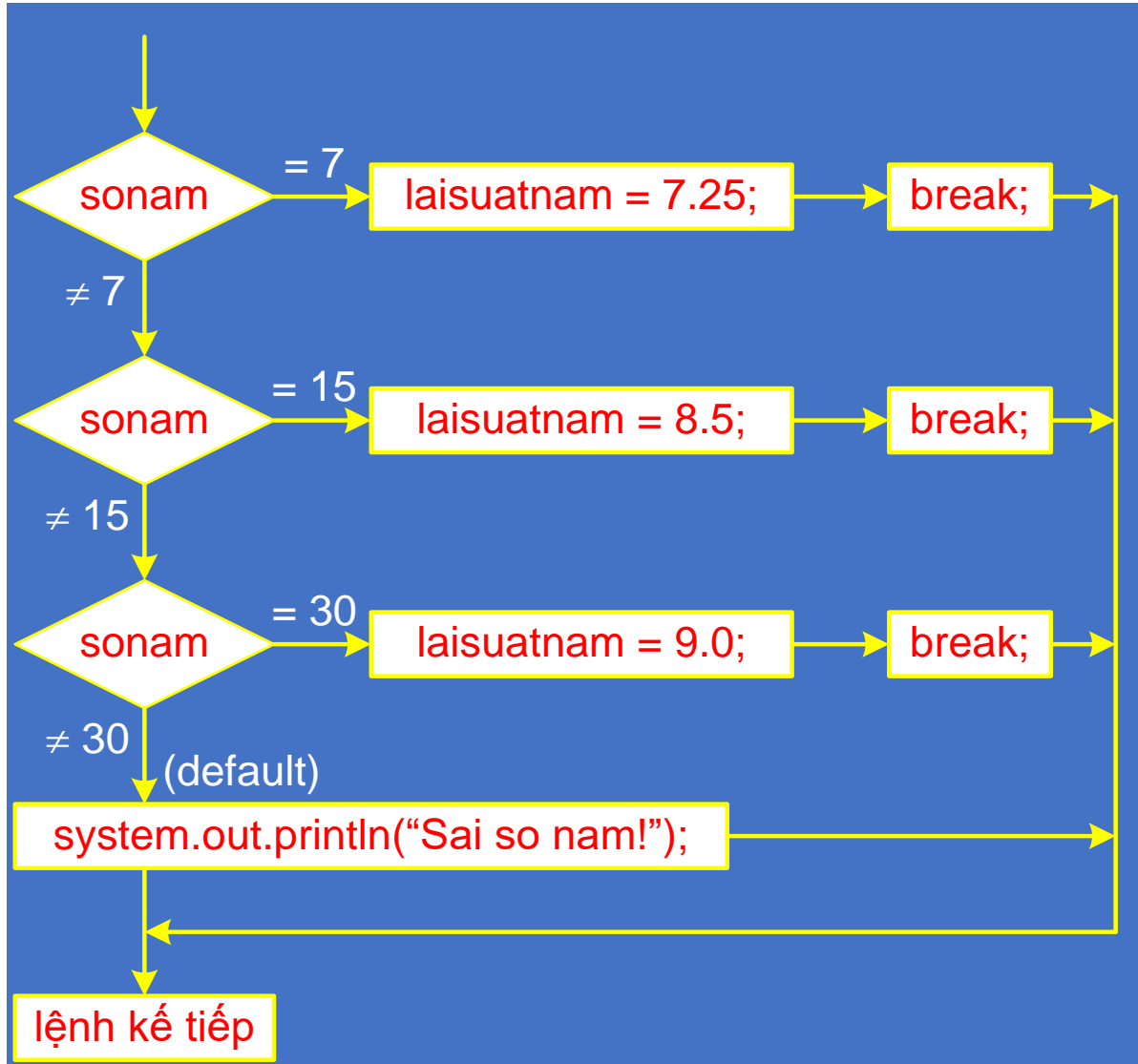
```
if (n)
    system.out.println
        ("So chan");
```

Lệnh switch

```
switch (bt_switch) {  
    case gtri1: lenh(s)1;  
        break;  
    case gtri2: lenh(s)2;  
        break;  
    .....  
    case gtriN: lenh(s)N;  
        break;  
    default: lenh(s)-khi-  
        default;  
}
```

```
switch (sonam) {  
    case 7:    laisuatnam = 7.25;  
        break;  
    case 15:   laisuatnam = 8.50;  
        break;  
    case 30:   laisuatnam = 9.0;  
        break;  
    default: System.out.println  
        ("Sai so nam, nhap 7, 15,  
        hoac 30");  
}
```

Lưu đồ lệnh switch



Quy tắc lệnh switch

- Biểu thức switch phải sinh ra một giá trị kiểu char, byte, short, hoặc int, và phải luôn được bao trong cặp dấu ngoặc tròn.
- gtri1, ..., gtriN phải có cùng kiểu dữ liệu với giá trị của biểu thức switch.
- Từ khóa break là tùy chọn, nhưng nên được sử dụng cuối mỗi trường hợp để thoát khỏi phần còn lại của lệnh switch. Nếu không có lệnh break, lệnh case tiếp theo sẽ được thực hiện.

```
switch (bt_switch) {  
    case gtri1: lenh(s)1;  
                break;  
    case gtri2: lenh(s)2;  
                break;  
    .....  
    case gtriN: lenh(s)N;  
                break;  
    default: lenh(s)-khi-  
              default;  
}
```

Quy tắc lệnh switch (tiếp)

- Trường hợp **default** là tùy chọn, có thể sử dụng để thực hiện các lệnh khi không có trường hợp nào ở trên là đúng.
- Thứ tự của các trường hợp (gồm cả trường hợp default) là không quan trọng. Tuy nhiên, phong cách lập trình tốt là nên theo một trình tự logic của các trường hợp và đặt trường hợp default cuối cùng.

Lưu ý

- Đừng quên dùng lệnh break khi cần thiết. ví dụ đoạn mã sau luôn hiển thị "Sai so nam!" bất chấp **sonam** là bao nhiêu. Giả sử **sonam** bằng 15. Lệnh **laisuatnam = 8.50** được thực hiện, tiếp theo là lệnh **laisuatnam = 9.0**, và cuối cùng là lệnh **System.out.println("Sai so nam!")**.

```
switch (sonam) {  
    case 7: laisuatnam = 7.25;  
    case 15: laisuatnam = 8.50;  
    case 30: laisuatnam = 9.0;  
    default: System.out.println("Sai so nam!");  
}
```


Toán tử điều kiện

(BT_logic) ? bt1 : bt2 ;

- **Ví dụ 1:**

```
if (x > 0) y = 1  
else y = -1;
```

- **tương đương với:**

```
y = (x > 0) ? 1 : -1;
```

Toán tử điều kiện (tiếp)

Ví dụ 2:

```
System.out.println(  
    (so % 2 == 0)? so + "la so chan" : so + "la so  
    le");
```

- tương đương với:

```
if (so % 2 == 0)  
    System.out.println(so+"la so chan");  
else  
    System.out.println(so + "la so le");
```

Các lệnh lặp

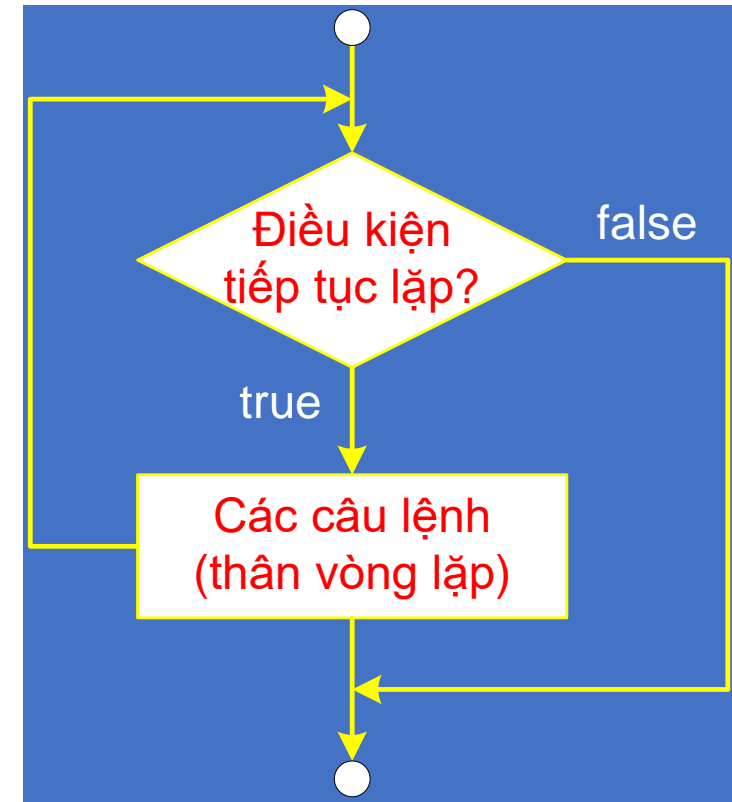
- ➡ Lệnh lặp `while`
- ➡ Lệnh lặp `do-while`
- ➡ Lệnh lặp `for`
- ➡ `break` và `continue`

Lệnh lặp while

```
while  
(đk_tiếp_tục_lặp) {  
    // thân_vòng_lặp;  
    các_câu_lệnh;  
}
```

Ví dụ:

```
int i = 0;  
while (i < 100) {  
    System.out.println("Welcome to  
Java!");  
    i++;  
}
```



Lưu ý

- Đừng sử dụng giá trị dấu chấm động để kiểm tra đẳng thức trong một điều khiển lặp. Vì giá trị dấu chấm động là gần đúng, sử dụng chúng có thể dẫn đến bộ đếm thiếu chính xác và kết quả sai. Ví dụ sau nên sử dụng giá trị int cho biến data. Nếu data có kiểu thực thì data != 0 có thể là true dù data bằng 0.

```
// data should be zero
double data = Math.pow(Math.sqrt(2), 2) - 2;

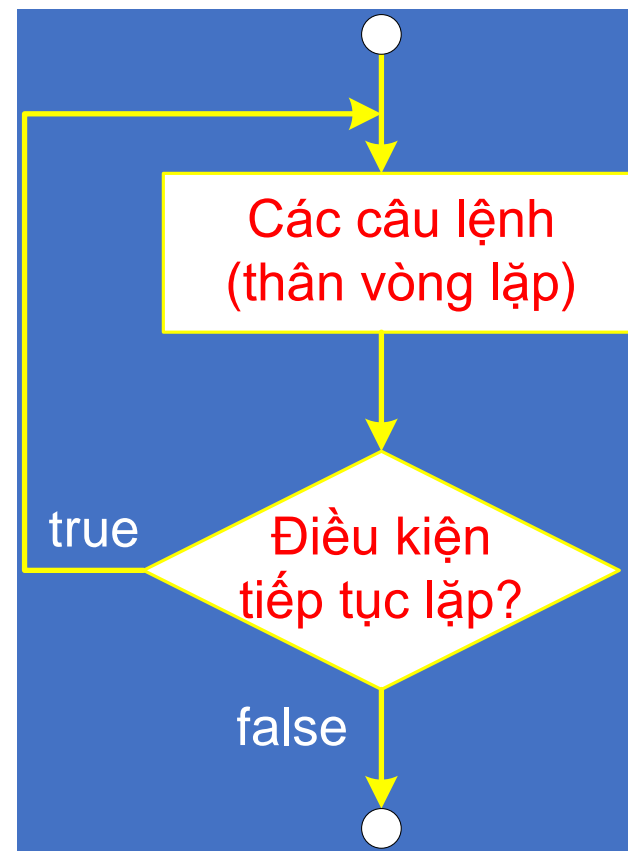
if (data == 0)
    System.out.println("data is zero");
else
    System.out.println("data is not zero");
```

Lệnh lặp do-while

```
do {  
    // thân_vòng_lặp;  
    các_câu_lệnh;  
} while (đk_tiếp_tục_lặp);
```

Ví dụ:

```
int i = 0;  
do {  
    System.out.println("Welcome to Java!");  
    i++;  
} while (i < 100)
```



Lệnh lặp for

```
for (khởi_tạo; đk_tiếp_tục_lặp; việc_sau_mỗi_lần_lặp)
{
    // thân vòng lặp;
    các_câu_lệnh;
}
```

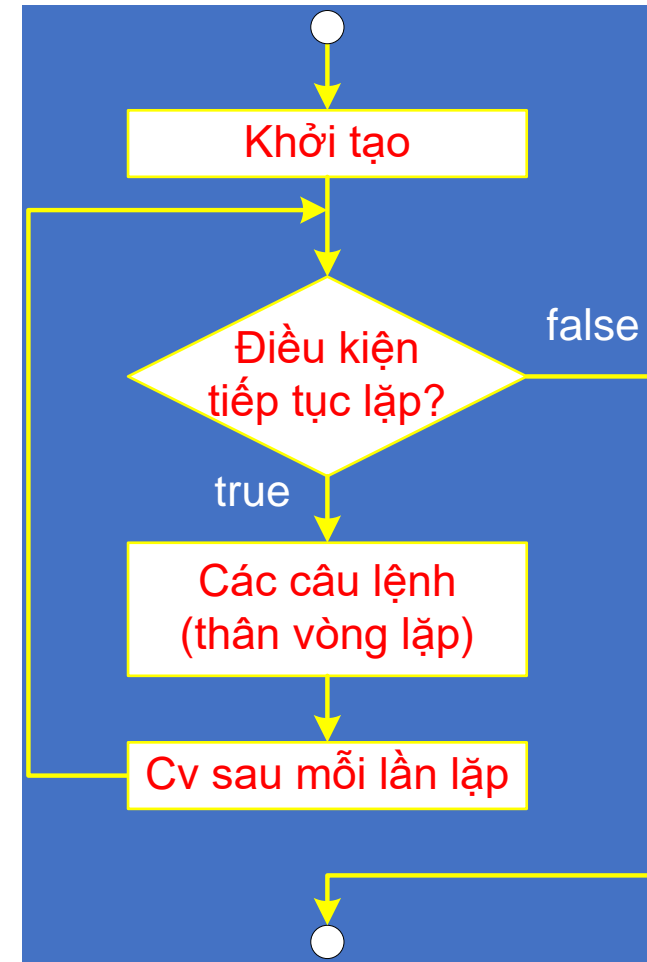
```
int i = 0;
while (i < 100) {
    System.out.println("Welcome to Java! " + i);
    i++;
}
```

Example:

```
int i;
for (i = 0; i < 100; i++) {
    System.out.println("Welcome to Java! " + i);
}
```

Lưu đồ lệnh lặp for

```
for (khởi_tạo; đk_tiếp_tục_lặp;  
cviệc_sau_mỗi_lần_lặp)  
{  
    // thân vòng lặp;  
    các_câu_lệnh;  
}
```



Lưu ý

- Các trường hợp sau đây là đúng:

```
for (int i = 1; i < 100; System.out.println(i++));
```

```
for (int i = 0, j = 0; (i + j < 10); i++, j++) {  
    // Do something  
}
```

```
for ( ; ; ) {  
    // Do something  
}
```

tương
đương

```
while (true) {  
    // Do something  
}
```

Các ví dụ lệnh lặp for

➡ Ví dụ: Sử dụng các lệnh lặp for

```
public class ForExample {  
    public static void main(String[] args) {  
        for (int i = 1; i <= 10; i++) {  
            System.out.println(i);  
        }  
    }  
}
```

Các ví dụ lệnh lặp for

➡ Ví dụ: Sử dụng các lệnh for lồng nhau

```
Product Owner, BApublic class NestedForExample {  
    public static void main(String[] args) {  
        // Vòng lặp i  
        for(int i=1;i<=3;i++){  
            // Vòng lặp j  
            for(int j=1;j<=3;j++){  
                System.out.println(i+" "+j);  
            } // Kết thúc vòng lặp i  
        } // Kết thúc vòng lặp j  
    }  
}
```

Vòng lặp for cải tiến

- Vòng lặp for cải tiến được sử dụng để lặp mảng(array) hoặc collection trong java.

- **Cú pháp :**

```
for (Type var : array) {  
    // Khối lệnh được thực thi  
}
```

- **Ví Dụ:**

```
public class ForEachExample {  
    public static void main(String[] args) {  
        int arr[] = { 12, 23, 44, 56, 78 };  
        for (int i : arr) {  
            System.out.println(i);  
        }  
    }  
}
```

Vòng lặp for gán nhãn

➤ Chúng ta có thể đặt tên cho mỗi vòng lặp for bằng cách gán nhãn trước vòng lặp for. Điều này rất hữu dụng khi chúng ta muốn thoát/tiếp tục(break/continues) chạy vòng lặp for.

➤ Cú pháp :

ten_nhan:

```
for (khởi_tạo_bien ; check_dieu_kien ; tang/giam_bien) {  
    // Khối lệnh được thực thi  
}
```

Vòng lặp for gán nhãn

➤ Ví Dụ:

```
public class LabeledForExample {  
    public static void main(String[] args) {  
        aa: for (int i = 1; i <= 3; i++) {  
            bb: for (int j = 1; j <= 3; j++) {  
                if (i == 2 && j == 2) {  
                    break aa;  
                }  
                System.out.println(i + " " + j);  
            }  
        }  
    }  
}
```

Cách sử dụng lệnh lặp?

- Ba lệnh lặp while, do, và for là tương đương nhau trong nhiều trường hợp; nghĩa là bạn có thể viết một vòng lặp bằng một dạng bất kỳ trong 3 dạng trên.
- Lệnh lặp for có thể sử dụng khi biết trước số lần lặp, ví dụ khi bạn muốn in ra một thông báo 100 lần.
- Lệnh lặp while có thể sử dụng khi không biết trước số lần lặp, như trong trường hợp đọc vào các số đến khi gặp số 0.
- Lệnh lặp do-while có thể sử dụng thay lệnh while khi thân vòng lặp phải được thực hiện trước khi kiểm tra điều kiện tiếp tục lặp.

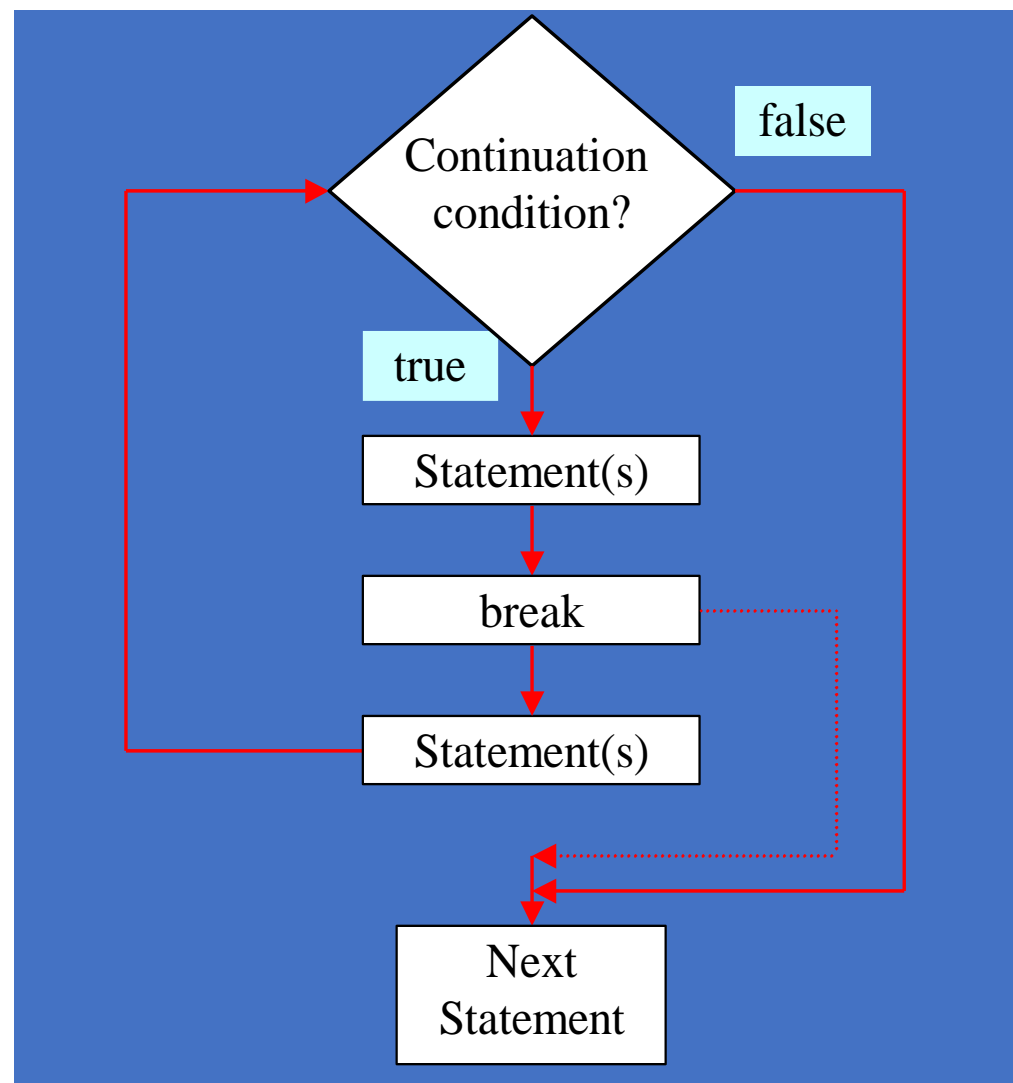
Lưu ý

```
for (int i=0; i<10; i++);  
{  
    System.out.println("i is " + i);  
}
```

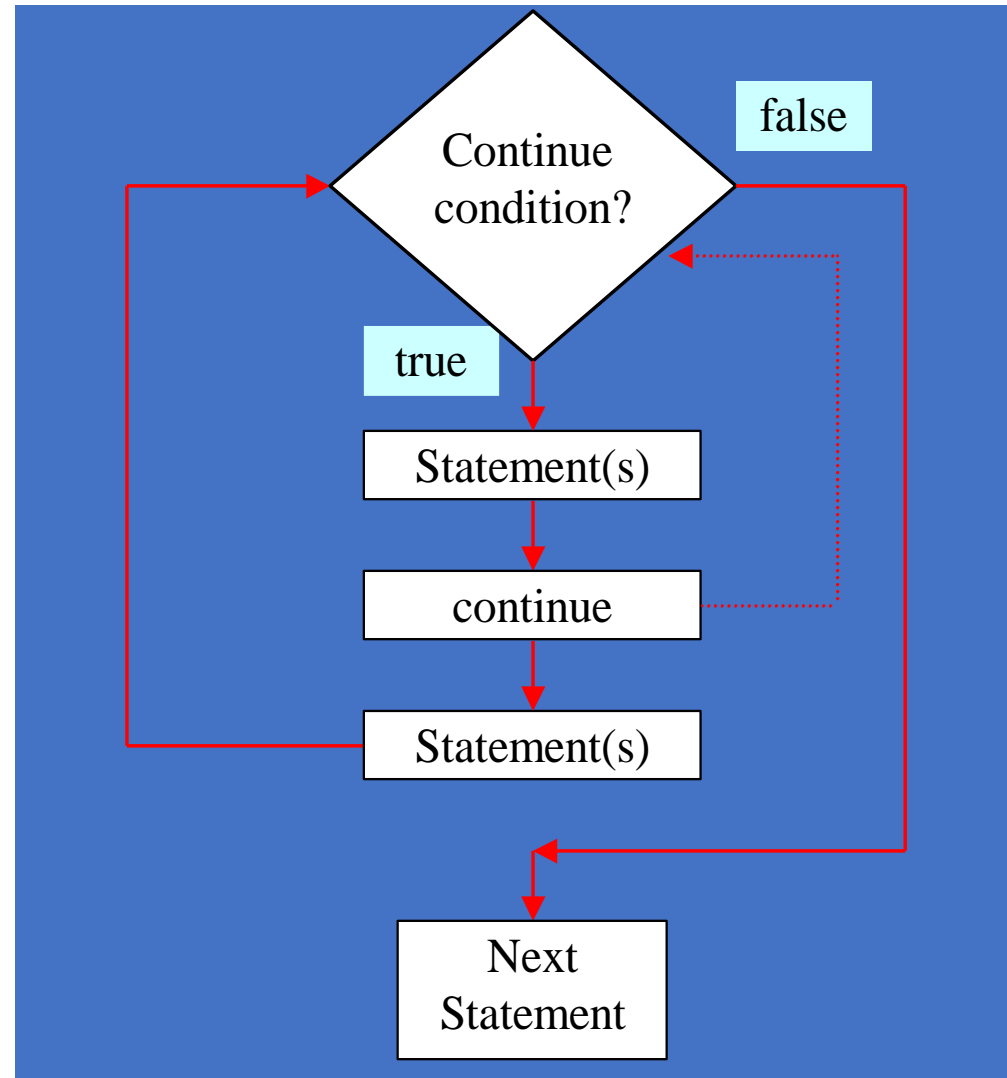
```
int i=0;  
while (i<10);  
{  
    System.out.println("i is " + i);  
    i++;  
}
```

```
int i=0;  
do {  
    System.out.println("i is " + i);  
    i++;  
} while (i<10);
```


Từ khóa break



Từ khóa continue



Sử dụng break và continue

Ví dụ sử dụng Break trong java với vòng lặp for:

```
90 public class BreakExample {  
91     public static void main(String[] args) {  
92         for (int i = 1; i <= 10; i++) {  
93             if (i == 5) {  
94                 break;  
95             }  
96             System.out.println(i);  
97         }  
98     }  
99 }
```

Sử dụng break và continue

Ví dụ sử dụng Continue trong java với vòng lặp for:

```
102 public class ContinueExample {  
103     public static void main(String[] args) {  
104         for (int i = 1; i <= 10; i++) {  
105             if (i == 5) {  
106                 continue;  
107             }  
108             // Khi i == 5 thì không in i = 5 ra màn hình  
109             System.out.println(i);  
110         }  
111     }  
112 }
```

Ví dụ: Hiển thị ước số chung lớn nhất

Trong ví dụ này chúng ta sẽ gán sẵn hai số là **num1** = 55 và **num2** = 121, sau đó dùng vòng lặp For để xét điều kiện cho hai số.

```
114 public class UocChungLonNhat {  
115     public static void main(String[] args) {  
116         int num1 = 55, num2 = 121, gcd = 1;  
117         for(int i = 1; i <= num1 && i <= num2; i++)  
118         {  
119             if(num1%i==0 && num2%i==0)  
120                 gcd = i;  
121         }  
122         System.out.printf("Ước chung lớn nhất của %d và %d là: %d", num1, num2, gcd);  
123     }  
124 }
```

Ví dụ: Hiển thị ước số chung lớn nhất

Trong ví dụ này chúng ta sẽ gán sẵn hai số là **num1** = 55 và **num2** = 121, sau đó dùng vòng lặp While để xét điều kiện cho hai số.

```
128 import java.util.Scanner;
129 public class UocChungLonNhat {
130     public static void main(String[] args) {
131         int num1, num2;
132         Scanner scanner = new Scanner(System.in);
133         System.out.print("Nhập số thứ nhất:");
134         num1 = (int)scanner.nextInt();
135         System.out.print("Nhập số thứ hai:");
136         num2 = (int)scanner.nextInt();
137         scanner.close();
138         while (num1 != num2) {
139             if(num1 > num2)
140                 num1 = num1 - num2;
141             else
142                 num2 = num2 - num1;
143         }
144         System.out.printf("Ước chung lớn nhất là: %d", num2);
145     }
146 }
```

Ví dụ: Tìm lượng tiền bán hàng

Bạn vừa mới bắt đầu công việc bán hàng trong một cửa hàng. Thu nhập của bạn bao gồm một lương cơ bản \$5,000/năm và tiền hoa hồng được tính như sau:

Sales Amount	Commission Rate
\$0.01–\$5,000	8 %
\$5,000.01–\$10,000	10 %
\geq \$10,000.01	12 %

Mục đích của bạn là kiếm được \$30,000/năm. Viết chương trình tìm lượng tiền bán hàng nhỏ nhất bạn phải tạo ra để giúp bạn đạt được mục đích.

Ví dụ: Hiển thị kim tự tháp số

Sử dụng các lệnh lặp lồng nhau để in ra màn hình:

```
0
1 0 1
2 1 0 1 2
3 2 1 0 1 2 3
4 3 2 1 0 1 2 3 4
```

Mỗi dòng gồm 3 phần:

- Các ký tự trống đầu dòng,
- Các số đầu tiên, ví dụ như 2 1 0 trên dòng 3,
- Các số cuối cùng, ví dụ như 1 2 trên dòng 3.

Ví dụ: Hiển thị kim tự tháp số

```
190 public static void main(String[] args) {
191     // TODO code application logic here
192     int n;
193     System.out.println("Nhap vao do cao cua thap");
194     Scanner sc = new Scanner(System.in);
195     n = sc.nextInt();
196
197     int i = 0, j;
198     while (i < n)
199     {
200         for(j = n - 1; j > i; j--)
201         {
202             System.out.print(" ");
203         }
204
205         for (j = i; j >= 0; j--)
206         {
207             System.out.printf("%d", j);
208         }
209         for (j = 1; j <= i; j++)
210         {
211             System.out.printf("%d", j);
212         }
213         System.out.println(" ");
214         i = i + 1;
215     }
216 }
```

Ví dụ: Hiển thị các số nguyên tố

Ví dụ này hiển thị 100 số nguyên tố đầu tiên trên 1 dòng

Ví dụ: Hiển thị các số nguyên tố

```
220 import java.util.Scanner;
221 class KiemTraSoNguyenTo
222 {
223     public static void main (String[] args)
224     {
225         int i =0;
226         int num =0;
227         //Empty String
228         String primeNumbers = "";
229         for (i = 1; i <= 100; i++)
230         {
231             int counter=0;
232             for(num =i; num>=1; num--)
233             {
234                 if(i%num==0)
235                 {
236                     counter = counter + 1;
237                 }
238             }
239             if (counter ==2)
240             {
241                 //Hợp nhất các số nguyên tố thành 1 chuỗi
242                 primeNumbers = primeNumbers + i + " ";
243             }
244         }
245         System.out.println("Các số nguyên tố từ 1 đến 100 là:");
246         System.out.println(primeNumbers);
247     }
248 }
```





THANK YOU