





CẤU TRÚC VÒNG LẶP

Mai Trong Nhan (Mr.)

Lecturer

+84 (0) 385.852.284

nhanmt@imic.edu.vn



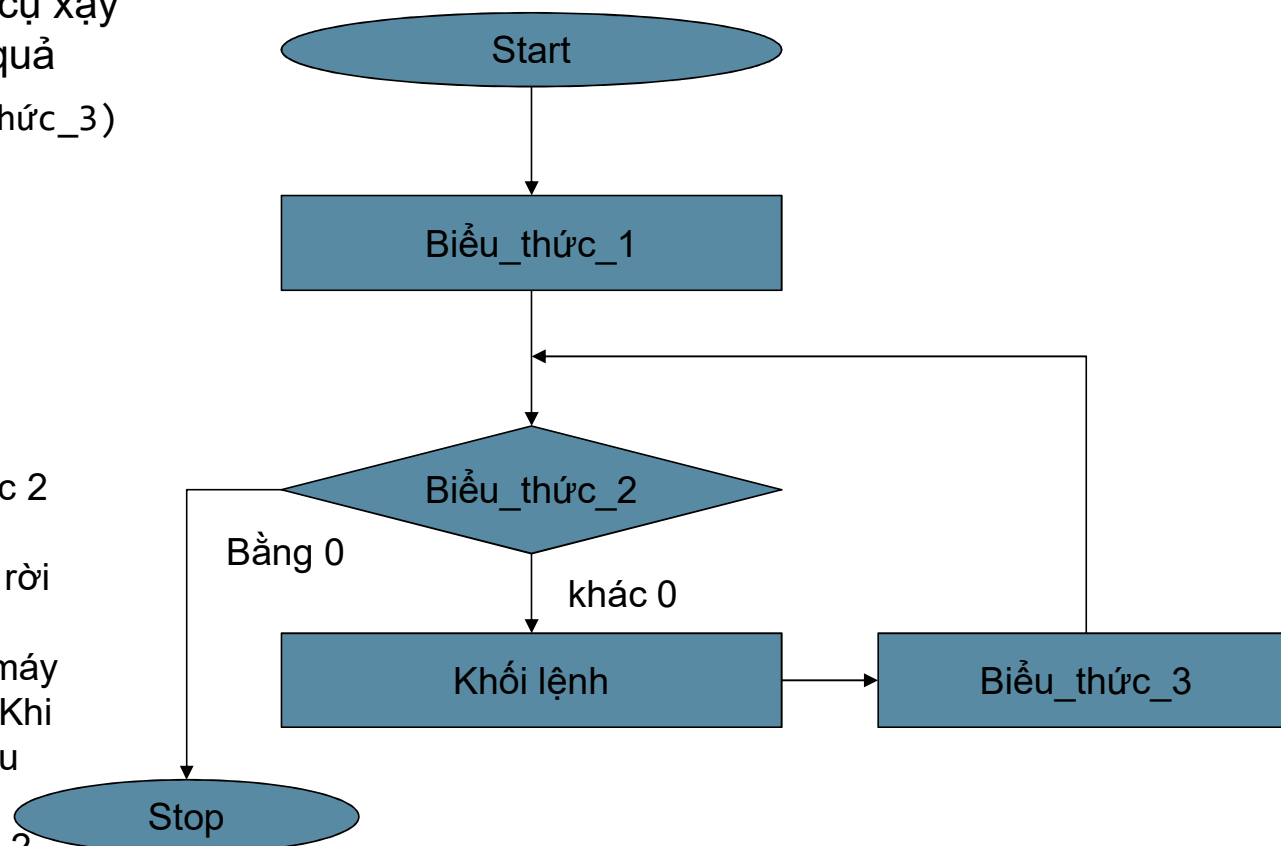
1.1. Toán tử for

Toán tử for cung cấp cho chúng ta công cụ xây dựng xây dựng chu trình một cách hiệu quả

```
for (biểu_thức_1; biểu_thức_2; biểu_thức_3)
{
    //khối lệnh
}
```

Toán tử for làm việc theo các bước sau:

1. Xác định biểu thức 1.
2. Xác định biểu thức 2.
3. Tùy thuộc vào tính đúng sai của biểu thức 2 máy sẽ lựa chọn một trong 2 nhánh.
 - a. Nếu biểu thức 2 có giá trị 0 (sai), máy sẽ rời khỏi for.
 - b. Nếu biểu thức 2 có giá trị khác 0 (đúng) máy sẽ thực hiện các câu lệnh trong thân for, Khi gặp dấu } cuối cùng của thân for hoặc câu lệnh continue máy sẽ chuyển tới bước 4.
4. Tính biểu thức 3, sau đó quay trở lại bước 2 để bắt đầu một vòng mới của chu trình





1.2. Một vài chú ý trong toán tử for

Nhận xét:

1. biểu thức 1 bao giờ cũng chỉ được tính một lần.
2. biểu thức 2, biểu thức 3 và thân for có thể được thực hiện lặp đi lặp lại nhiều lần.

Chú ý:

1. Biểu thức có thể vắng mặt trong for, khi đó nó được xem là đúng.
2. Bên trong thân for có thể sử dụng thêm toán tử for (được gọi là for lồng)
3. Khi gặp câu lệnh **break** trong thân for. Máy sẽ lập tức thoát khỏi for.
4. Trong thân for có thể dùng câu lệnh **continue** để chuyển đến đầu một vòng lặp mới của chu trình. Lúc đó máy sẽ bỏ qua những câu lệnh còn lại trong thân chu trình for.



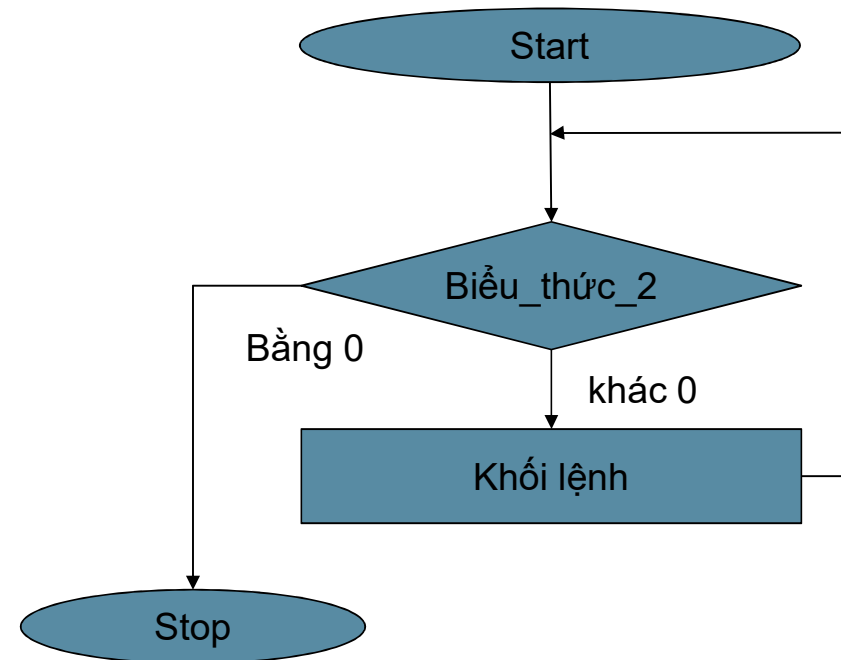
2.1. Toán tử while

- Cũng giống như for, toán tử while dùng để xây dựng các chu trình,

```
while (biểu_thức)
{
    //Khối lệnh - Thân chu trình
}
```

Toán tử while làm việc theo các bước sau:

- Xác định giá trị của biểu thức
- Tùy thuộc vào giá trị của biểu thức, máy sẽ lựa chọn một trong 2 nhánh:
 - Nếu biểu thức có giá trị 0 (sai), máy sẽ ra khỏi chu trình.
 - Nếu biểu thức có giá trị khác 0 (đúng) máy sẽ thực hiện các câu lệnh trong thân while, khi gặp dấu } cuối cùng của thân while máy sẽ quay lại bước 1.





2.2. Một vài chú ý trong toán tử while

Nhận xét:

Thân của while thực hiện một lần hoặc nhiều lần và cũng có thể không thực hiện lần nào nếu ngay từ đầu biểu thức (sau while) sai

Chú ý:

1. Bên trong thân của toán tử while có thể sử dụng các toán tử for hoặc while khác
2. Khi gặp câu lệnh break trong thân while, ngay lập tức máy sẽ thoát khỏi toán tử while, tức là lúc đó máy sẽ không thực hiện nhưng câu lệnh trong thân while phía dưới break
3. Trong thân while có thể dùng câu lệnh continue để chuyển đến một vòng lặp mới của chu trình. Tức là, khi gặp continue thì máy sẽ bỏ qua các câu lệnh còn lại trong thân chu trình để trở lại bước tính và kiểm tra biểu thức.



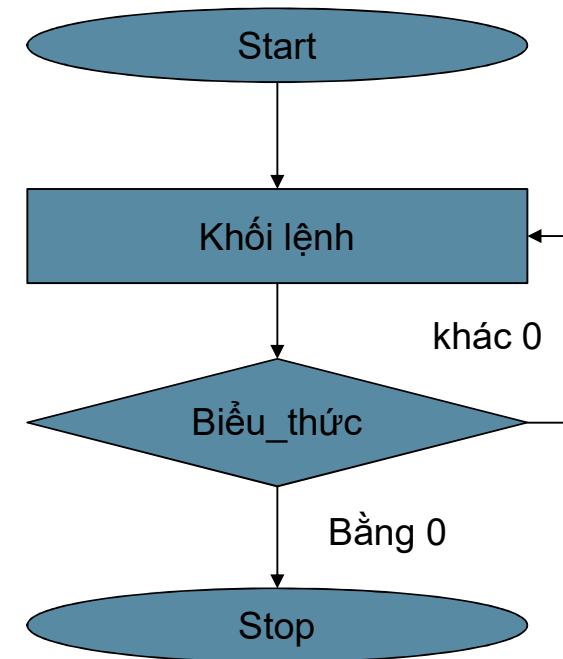
3.1. Toán tử do while

Trong các toán tử while và for việc kiểm tra điều kiện kết thúc đặt ở đầu chu trình. Khác với 2 toán tử trên, trong chu trình do-while việc kiểm tra điều kiện kết thúc đặt cuối chu trình. Như vậy thân của chu trình bao giờ cũng thực hiện ít nhất một lần.

```
do
{
    //thân chu trình
}
while(biểu_thức)
```

Hoạt động của do-while làm việc theo các bước sau:

1. Thực hiện các câu lệnh trong thân do-while
2. Khi gặp dấu } cuối cùng của thân do-while, máy sẽ xác định giá trị của biểu thức
3. Máy sẽ phân nhánh theo giá trị của biểu thức vừa nhận được:
 - a. Nếu biểu thức có giá trị khác 0 (đúng), máy sẽ trở lại bước 1 để tiếp tục thực hiện chu trình do-while mới
 - b. Nếu biểu thức có giá trị bằng 0 (sai), máy sẽ rời khỏi chu trình





3.2. Một vài chú ý trong toán tử do-while

Nhận xét:

Thân của do-while thực hiện ít nhất 1 lần

Chú ý:

1. Bên trong thân của toán tử do-while có thể sử dụng các toán tử lặp khác
2. Khi gặp câu lệnh break trong thân do-while, ngay lập tức máy sẽ thoát khỏi toán tử do-while, tức là lúc đó máy sẽ không thực hiện nhưng câu lệnh trong thân while phía dưới break
3. Trong thân do-while có thể dùng câu lệnh continue để chuyển đến một vòng lặp mới của chu trình. Tức là, khi gặp continue thì máy sẽ bỏ qua các câu lệnh còn lại trong thân chu trình để trở lại bước tính và kiểm tra biểu thức.



Thank you.