



Session 02:

Câu lệnh điều khiển, vòng lặp



- 1. Câu lệnh if else, nguyên lý hoạt động**
- 2. Câu lệnh Switch Case**
- 3. Giới thiệu vòng lặp, vòng lặp for**
- 4. Vòng lặp while, do while**
- 5. Sử dụng các câu lệnh nhảy break, continue**

Câu lệnh điều khiển - Control Statement

Conditional Statement



Jump Statement

Iterative Statement



1. Câu lệnh if else, nguyên lý hoạt động



Bài toán:

- Giá vé xe bus niêm yết: 7.000 VNĐ
- Trẻ em có tuổi nhỏ hơn 6 được miễn phí xe bus
- Học sinh tiểu học, trung học cơ sở và trung học phổ thông có tuổi từ 6 đến 18 được giảm 50% giá vé
- Sinh viên, người lao động có tuổi từ 18 đến 60 tuổi tính 100% giá vé
- Người cao tuổi có tuổi trên 60 được miễn phí

1. Câu lệnh if else, nguyên lý hoạt động

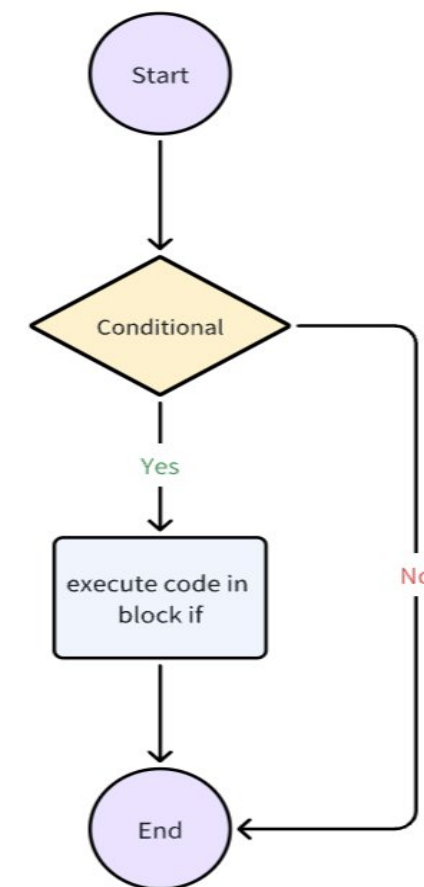
if thiếu (if không có else)

Cú pháp:

```
if ( condition ) {  
    // Execute body if  
}
```

Ý tưởng:

- Dùng if khi **chỉ quan tâm một trường hợp đặc biệt**
- Nếu điều kiện đúng → thực hiện
- Nếu sai → **không làm gì**



1. Câu lệnh if else, nguyên lý hoạt động

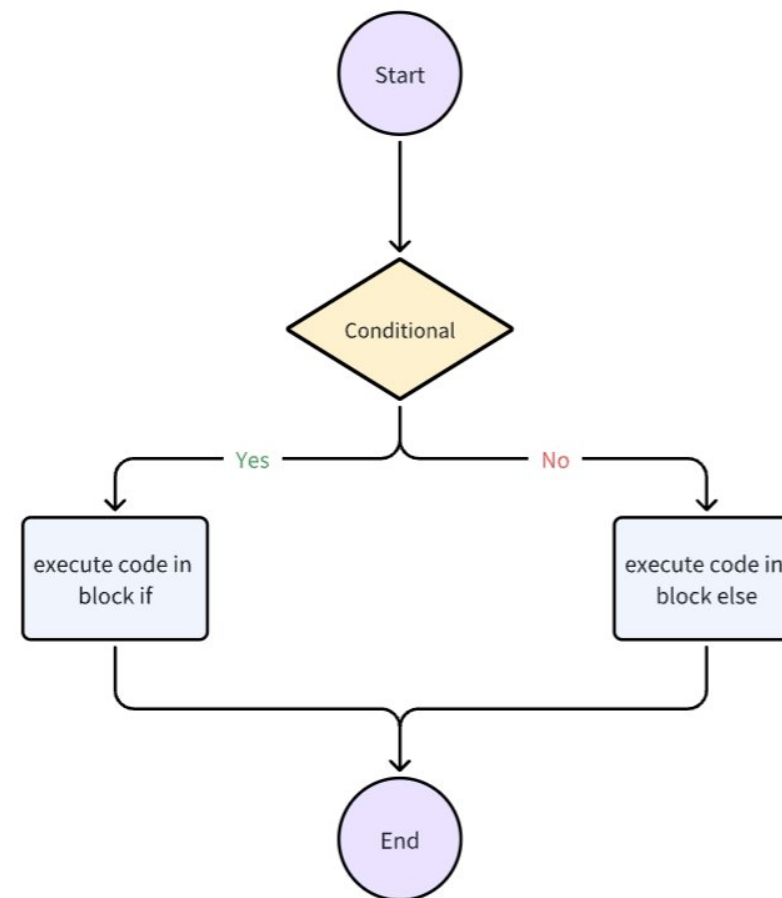
if đủ (if có else)

Cú pháp:

```
if ( condition ) {  
    // Execute body if  
} else {  
    // Execute body else  
}
```

Ý tưởng:

- Dùng khi chương trình có **2 lựa chọn rõ ràng**
- Điều kiện đúng → nhánh **if**
- Điều kiện sai → nhánh **else**



1. Câu lệnh if else, nguyên lý hoạt động

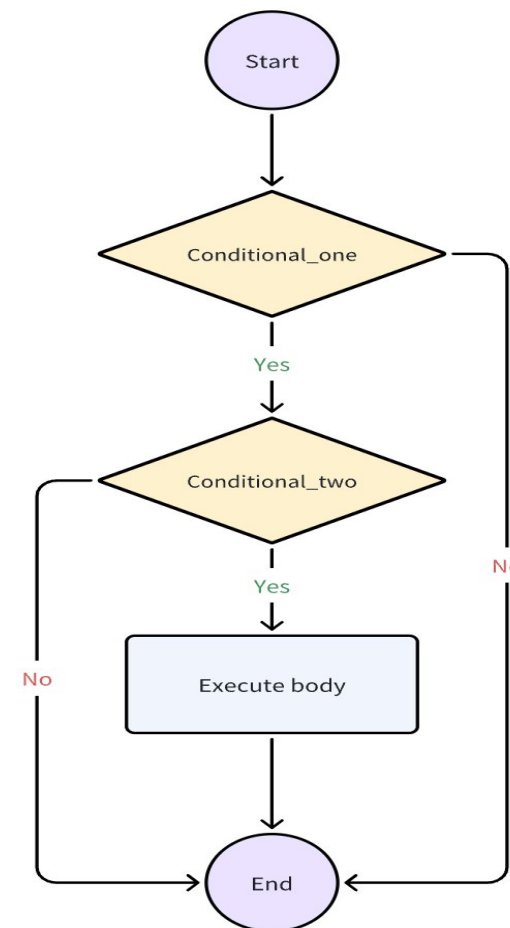
if lồng (nested if)

Cú pháp:

```
if ( condition_one ) {  
    // Execute body if one  
} else {  
    if ( condition_two ) {  
        // Execute body if two  
    } else {  
        // Execute body else two  
    }  
}
```

Ý tưởng:

- Trước tiên kiểm tra **tuổi nhỏ nhất**
- Nếu không thỏa mãn → tiếp tục kiểm tra nhóm tuổi tiếp theo
- Mỗi lần vào **else** → thu hẹp phạm vi tuổi



1. Câu lệnh if else, nguyên lý hoạt động

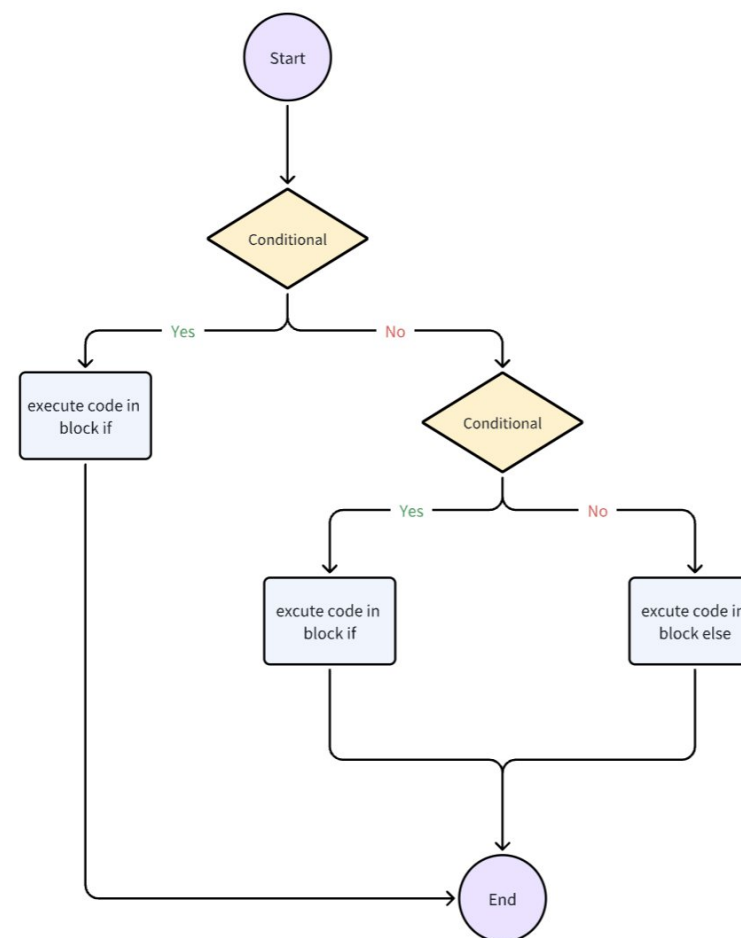
if bậc thang (ladder if)

Cú pháp:

```
if ( conditions_one ) {
    // Body if
} else if ( conditions_two ) {
    // Body else if
} else {
    // Body else
}
```

Ý tưởng:

- Kiểm tra **lần lượt từ trên xuống**
- Khi gặp điều kiện đúng → **dừng lại ngay**
- Chỉ **1 nhánh duy nhất** được thực hiện



2. Câu lệnh Switch Case



Bài toán:

Tính tiền gửi xe sinh viên trong trường đại học

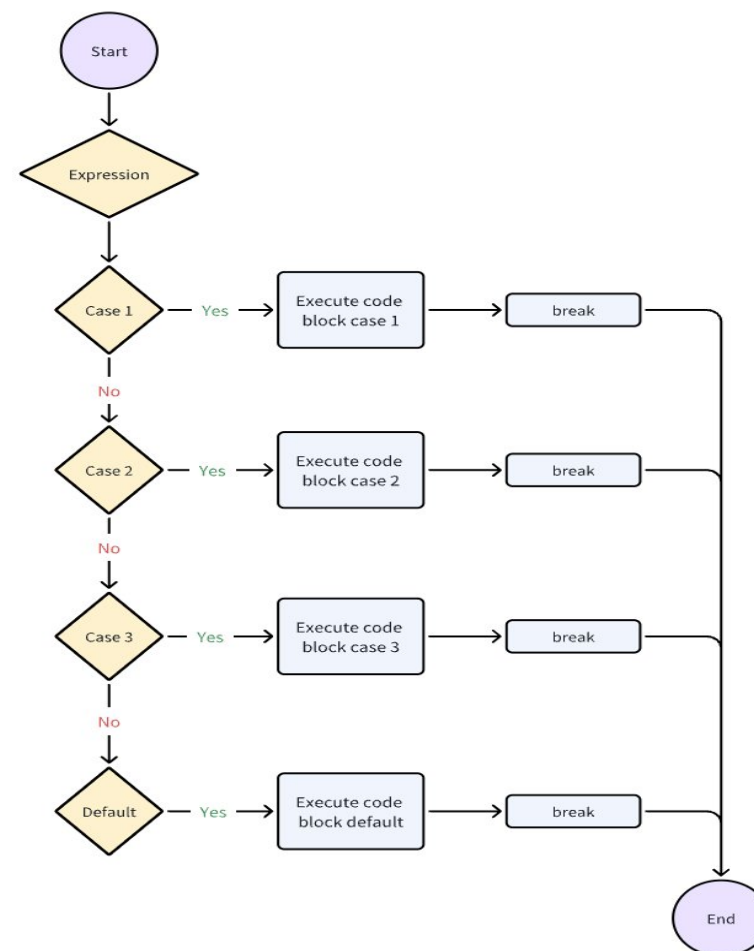
- Xe đạp → 2000 đ
- Xe máy → 5000 đ
- Ô tô → 20000 đ
- Xe điện → 3000 đ

Sinh viên nhập loại xe, chương trình sẽ tính và in ra số tiền gửi xe

2. Câu lệnh Switch Case

Ý tưởng:

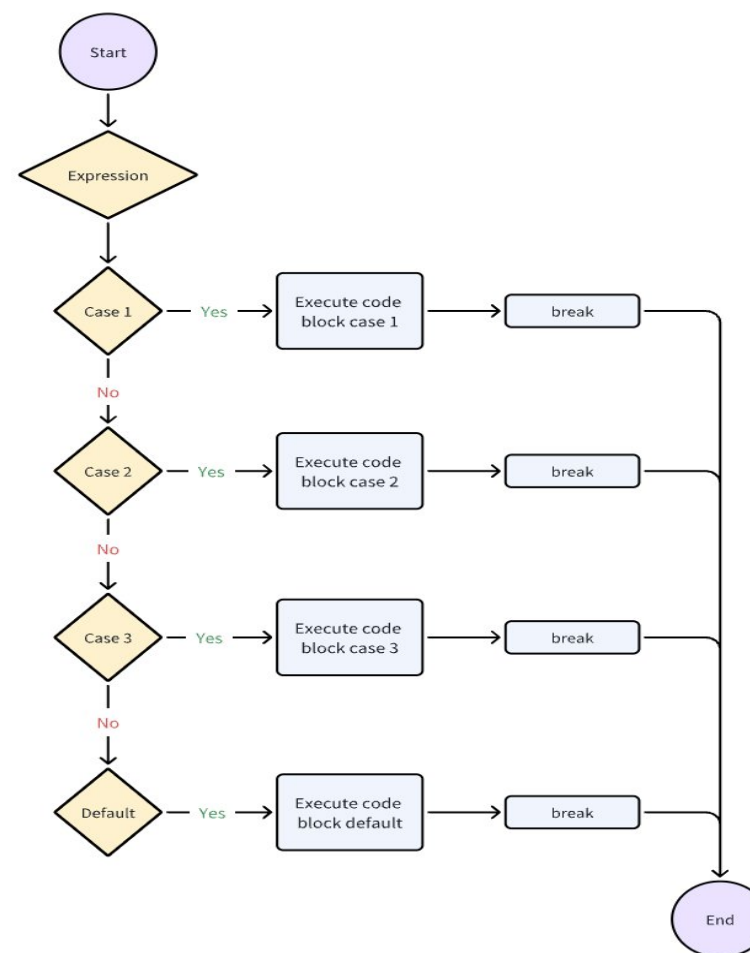
- Dùng để rẽ nhánh theo nhiều lựa chọn
- So sánh 1 biến duy nhất với nhiều giá trị cố định
- Khi khớp case → thực hiện và dừng lại
- Cần **break** để không rơi xuống case tiếp theo
- Chỉ 1 case duy nhất được thực hiện



2. Câu lệnh Switch Case

Cú pháp:

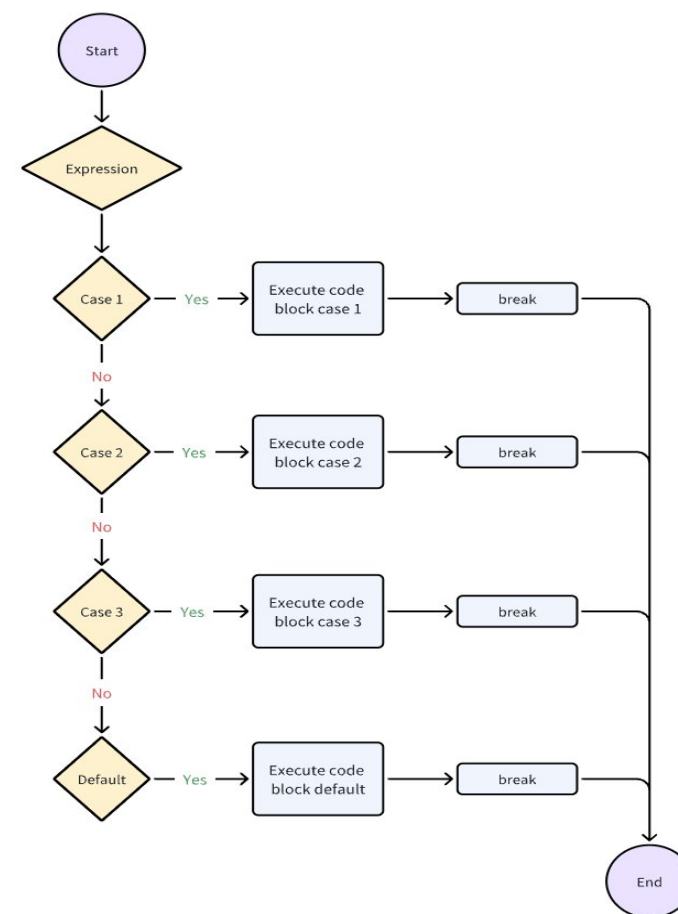
```
switch (expression) {
    case value1:
        // statement block 1
        break;
    case value2:
        // statement block 2
        break;
    case value3:
        // statement block 3
        break;
    default:
        // default statement block
}
```



2. Câu lệnh Switch Case

Áp dụng vào bài toán:

- Quy ước các loại xe:
 - 1 → Xe đạp → 2000 đ
 - 2 → Xe máy → 5000 đ
 - 3 → Ô tô → 20000 đ
 - 4 → Xe điện → 3000 đ
 - default → Loại xe không hợp lệ



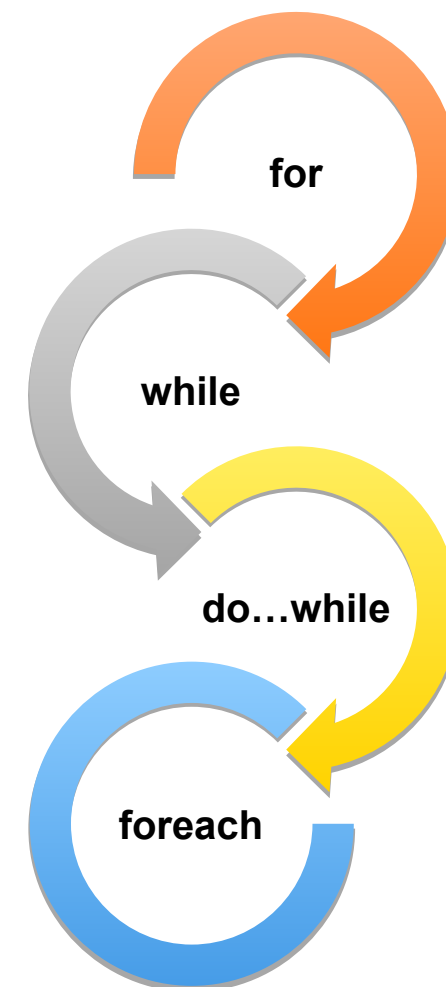
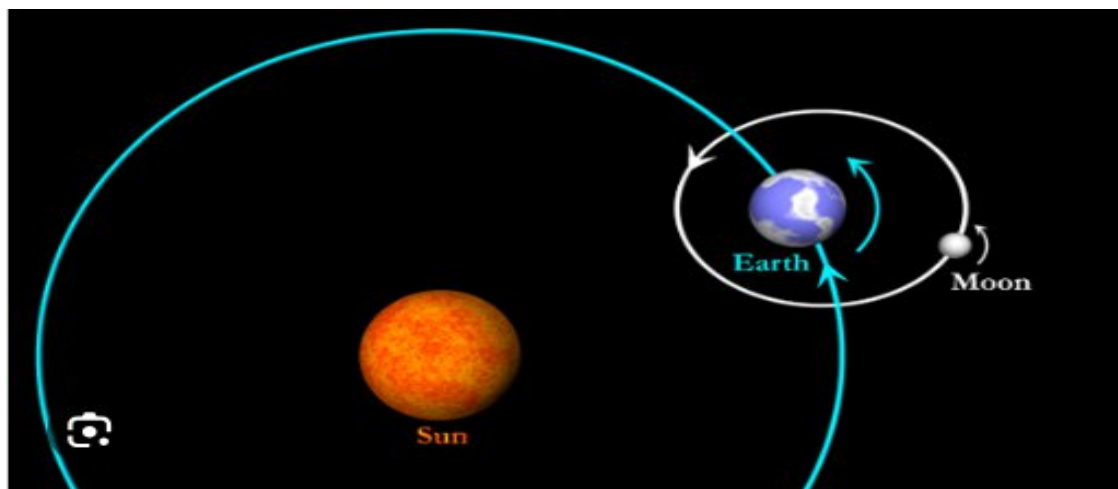
2. Câu lệnh Switch Case

if	Switch - case
Có thể sử dụng để so sánh lớn hơn , bé hơn,...	Chỉ có thể sử dụng để so sánh bằng hoặc khác nhau.
Mỗi câu lệnh if có một biểu thức điều kiện với giá trị trả về là true hoặc false	Tất cả các trường hợp (case) đều được so sánh với giá trị của biểu thức trong switch
Biểu thức điều kiện cần trả về boolean	Biểu thức trong switch có thể trả về các kiểu như : byte, short, char, int, String
Chỉ có một khối lệnh được thực thi nếu điều kiện đúng	Nếu điều kiện đúng mà không có lệnh break thì khối lệnh ở case tiếp theo vẫn được thực thi .

3. Giới thiệu vòng lặp, vòng lặp for

Vòng lặp:

- Giúp chương trình **thực hiện lặp lại** một hoặc nhiều câu lệnh
- **Giảm trùng lặp code**, chương trình ngắn gọn hơn
- Giúp chương trình **tự động hóa các thao tác lặp**
- Là công cụ **không thể thiếu** trong lập trình



3. Giới thiệu vòng lặp, vòng lặp for



Bài toán:

Một sinh viên vào **nhà sách** và mua **n cuốn sách**.

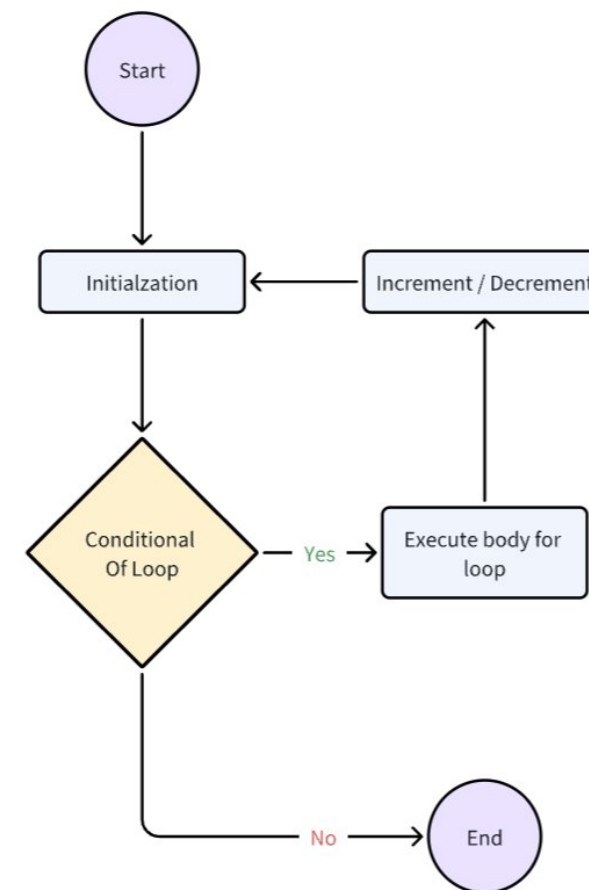
- Số lượng sách **n** được người dùng nhập từ bàn phím
- Mỗi cuốn sách có **giá tiền riêng**
- Chương trình cần:
 - Nhập giá từng cuốn sách (lần lượt)
 - Tính **tổng số tiền phải trả**

3. Giới thiệu vòng lặp, vòng lặp for

```
for ( initialize counters ; condition ; increment | decrement ) {  
    // Body  
}
```

Trong đó:

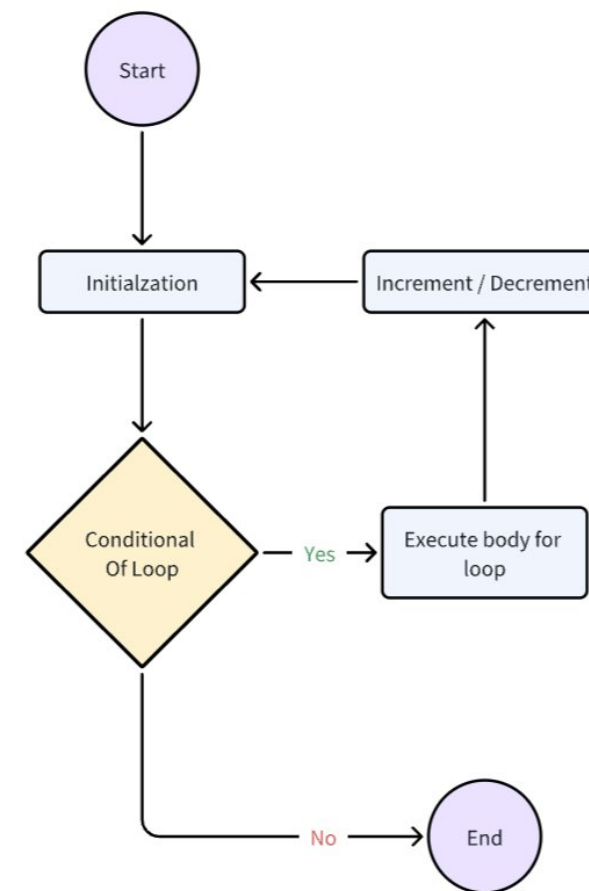
- **Initialize counters:** khởi tạo các biến điều khiển
- **Condition:** điều kiện để thực hiện vòng lặp
- **Increment | decrement:** tăng | giảm các biến điều khiển
- **Body:** Phần thân vòng lặp để thực hiện các logic cần lặp lại



3. Giới thiệu vòng lặp, vòng lặp for

Ý tưởng:

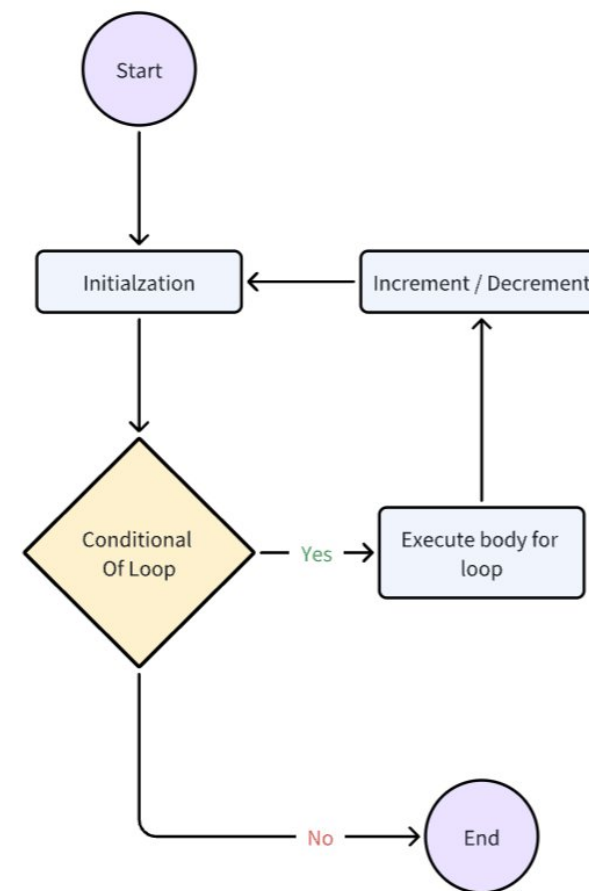
- Người dùng nhập số lượng sách n
- Tổng tiền ban đầu bằng 0
- Dùng **vòng lặp for** lặp từ 1 đến n
- Mỗi vòng lặp:
 - Nhập giá của 1 cuốn sách
 - Cộng giá đó vào tổng tiền
- Sau khi lặp xong → in ra **tổng tiền phải trả**



3. Giới thiệu vòng lặp, vòng lặp for

Luồng thực thi:

1. Bắt đầu chương trình
2. Nhập số lượng sách **n**
3. Khởi tạo **total = 0**
4. Thực hiện vòng lặp **for**
5. Nhập giá từng cuốn sách
6. Cộng dồn vào **total**
7. Kết thúc vòng lặp
8. In ra tổng tiền



4. Vòng lặp while, do while



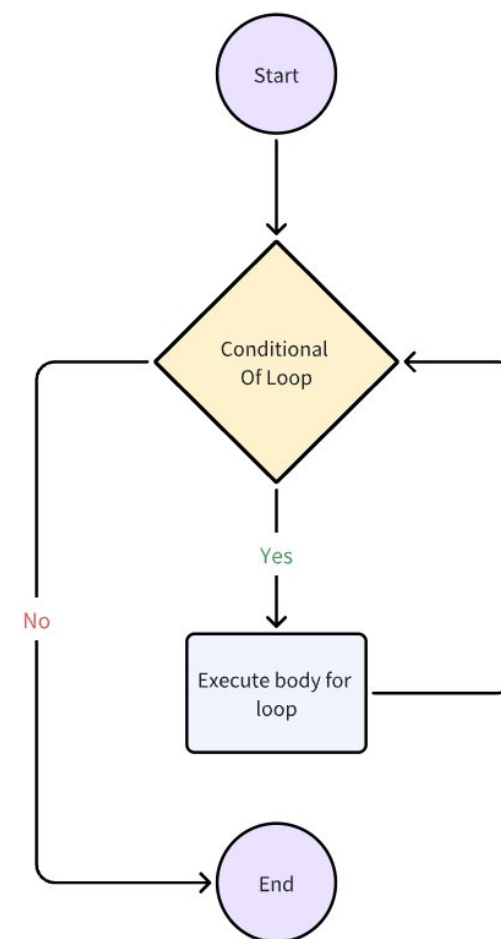
```
while ( conditions ) {  
    // Body  
}
```

Vòng lặp While:

- Vòng lặp while được sử dụng trong trường hợp chưa xác định được số lần lặp

Trong đó:

- **Conditions:** Là điều kiện để thực hiện vòng lặp
- **Body:** Phần thân vòng lặp để thực hiện các logic cần lặp lại



4. Vòng lặp while, do while

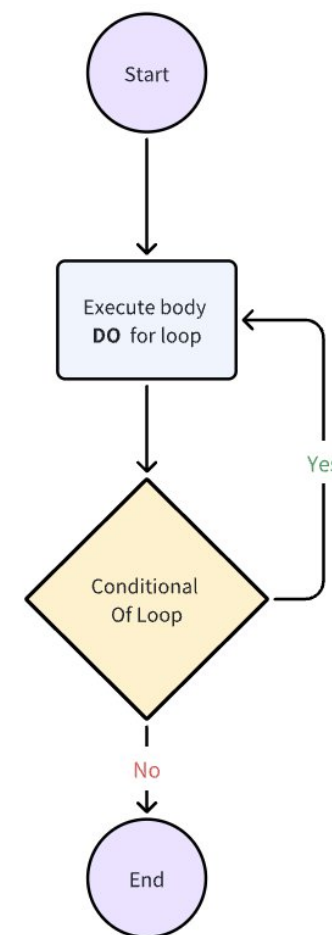
```
do {  
    // Body  
} while ( conditions );
```

Vòng lặp Do While:

- Vòng lặp do...while được sử dụng khi chưa biết trước số lần lặp và cần thực hiện ít nhất 1 lần

Trong đó:

- **Do:** Là phần thân vòng lặp
- **Conditions:** Là điều kiện để thực hiện vòng lặp
- **Body:** Phần thân vòng lặp để thực hiện các logic cần lặp lại



4. Vòng lặp while, do while

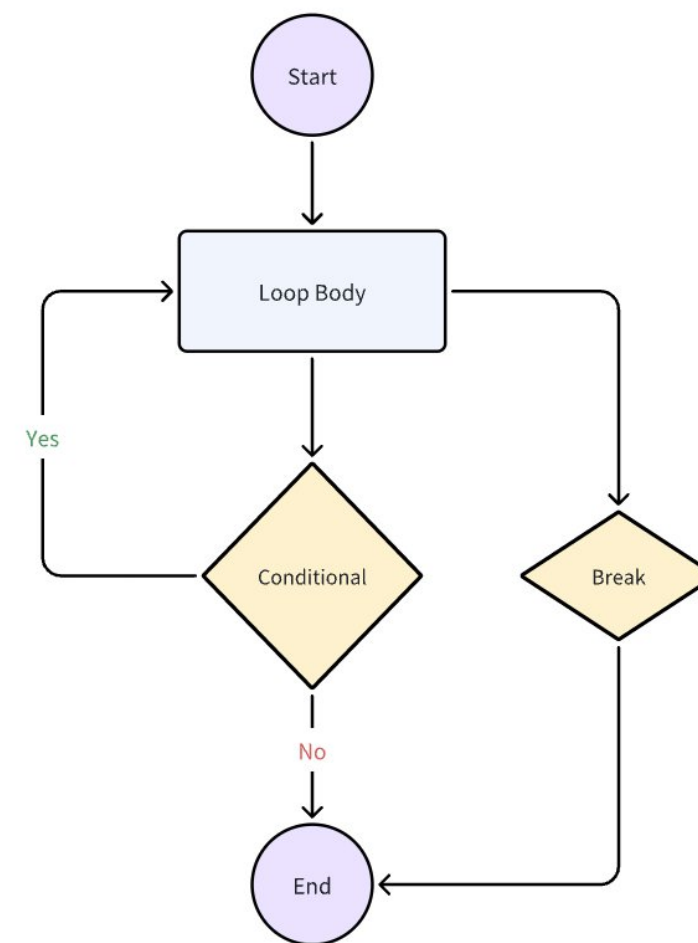
Loại vòng lặp	Khi nào nên dùng
for	Khi biết trước số lần lặp
while	Khi chưa biết trước số lần lặp , phụ thuộc vào điều kiện
do ... while	Khi vòng lặp cần chạy ít nhất 1 lần

5. Sử dụng các câu lệnh nhảy break, continue

```
for (initialization counter ; conditional ; increment | decrement) {  
    if( conditional ) {  
        break;  
    }  
}
```

Câu lệnh break:

- Kết thúc câu lệnh switch...case, câu lệnh lặp
- Điều khiển được chuyển đến câu lệnh tiếp theo ngoài vòng lặp

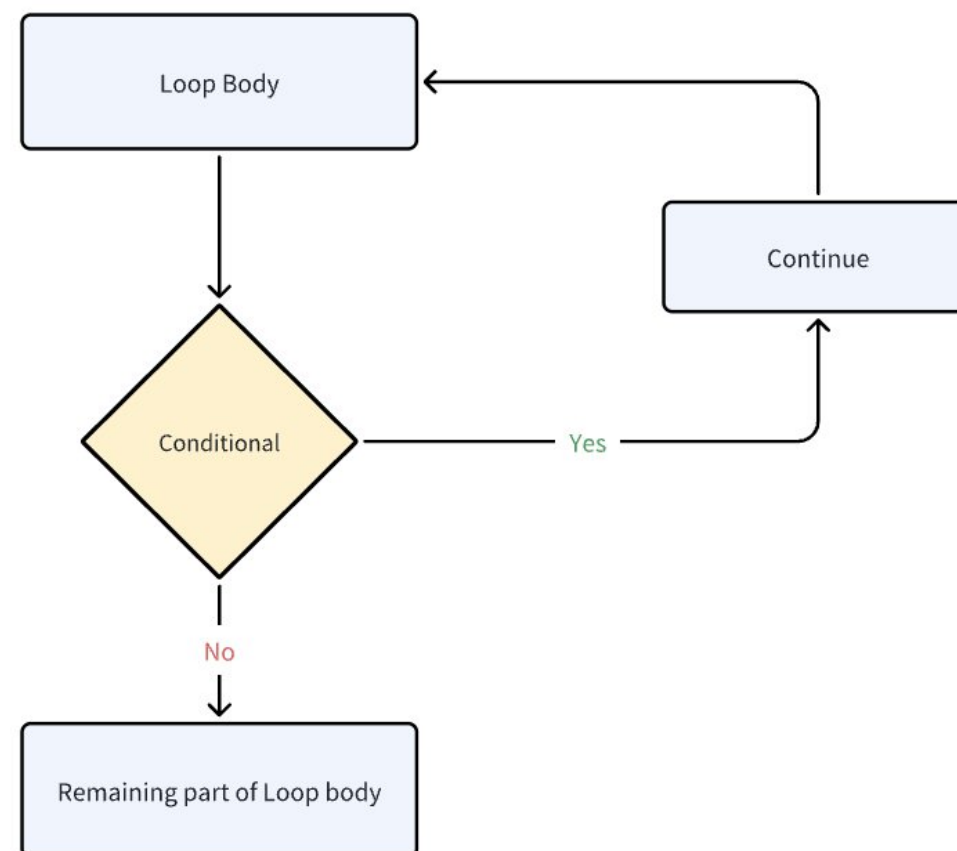


5. Sử dụng các câu lệnh nhảy break, continue

```
for (initialization counter ; conditional ; increment | decrement) {  
    if( conditional ) {  
        continue;  
    }  
}
```

Câu lệnh continue:

- Dừng vòng lặp hiện tại và chuyển đến vòng lặp tiếp theo
- Các câu lệnh bên dưới continue trong thân vòng lặp sẽ bị bỏ qua và điều khiển được chuyển tới vòng lặp kế tiếp



- ❑ Hiểu và nắm được kiến thức về Câu lệnh điều khiển
- ❑ Biết cách sử dụng các câu lệnh điều kiện if else
- ❑ Biết cách sử dụng câu lệnh điều kiện switch case
- ❑ Biết cách sử dụng các câu lệnh vòng lặp
- ❑ Nắm được các kiến thức về break và continue



KẾT THÚC

HỌC VIỆN ĐÀO TẠO LẬP TRÌNH CHẤT LƯỢNG NHẬT BẢN