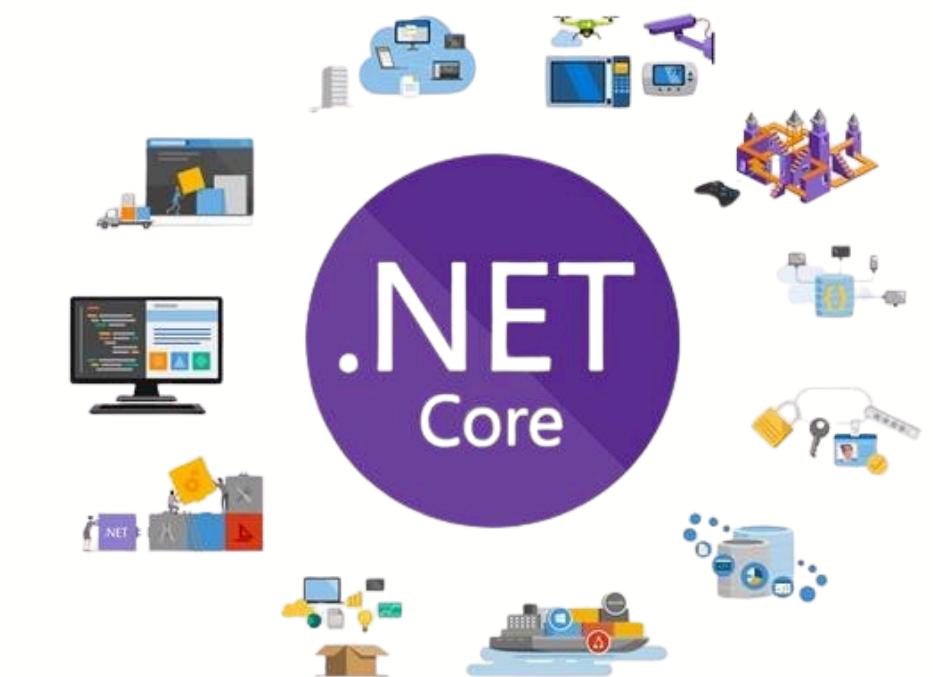
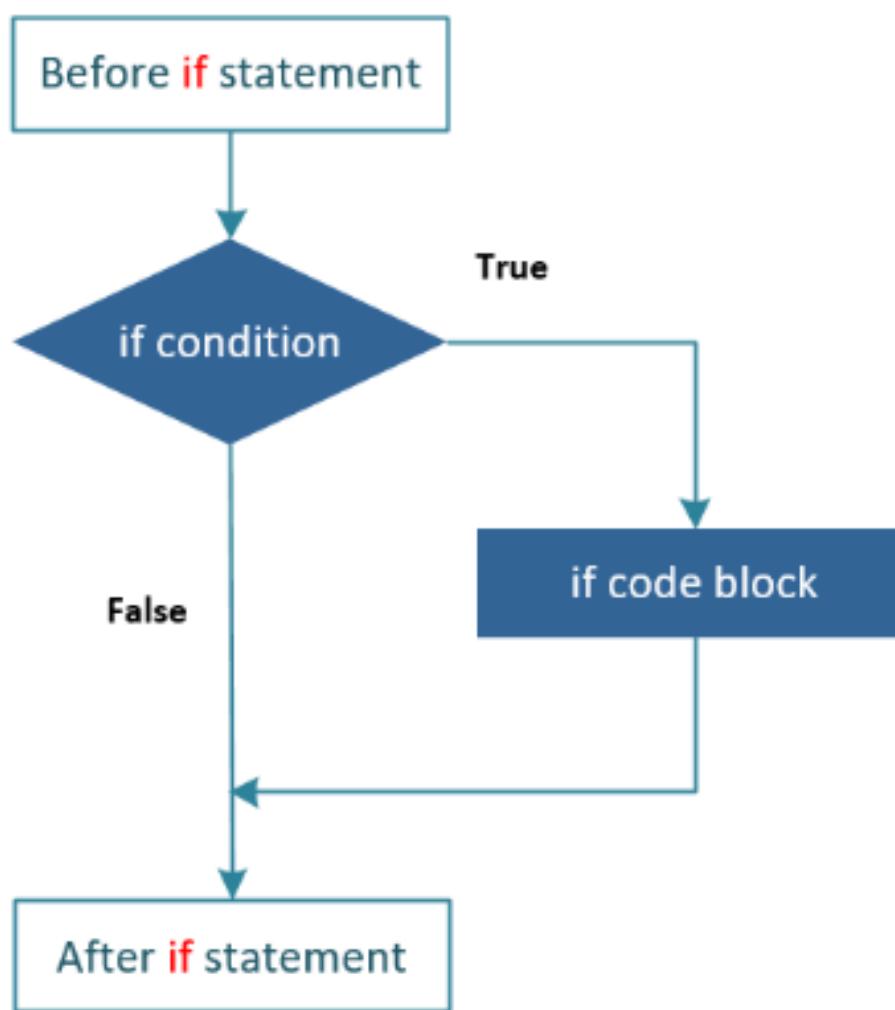


# IF ELSE STATEMENT

## CẤU TRÚC RẼ NHÁNH

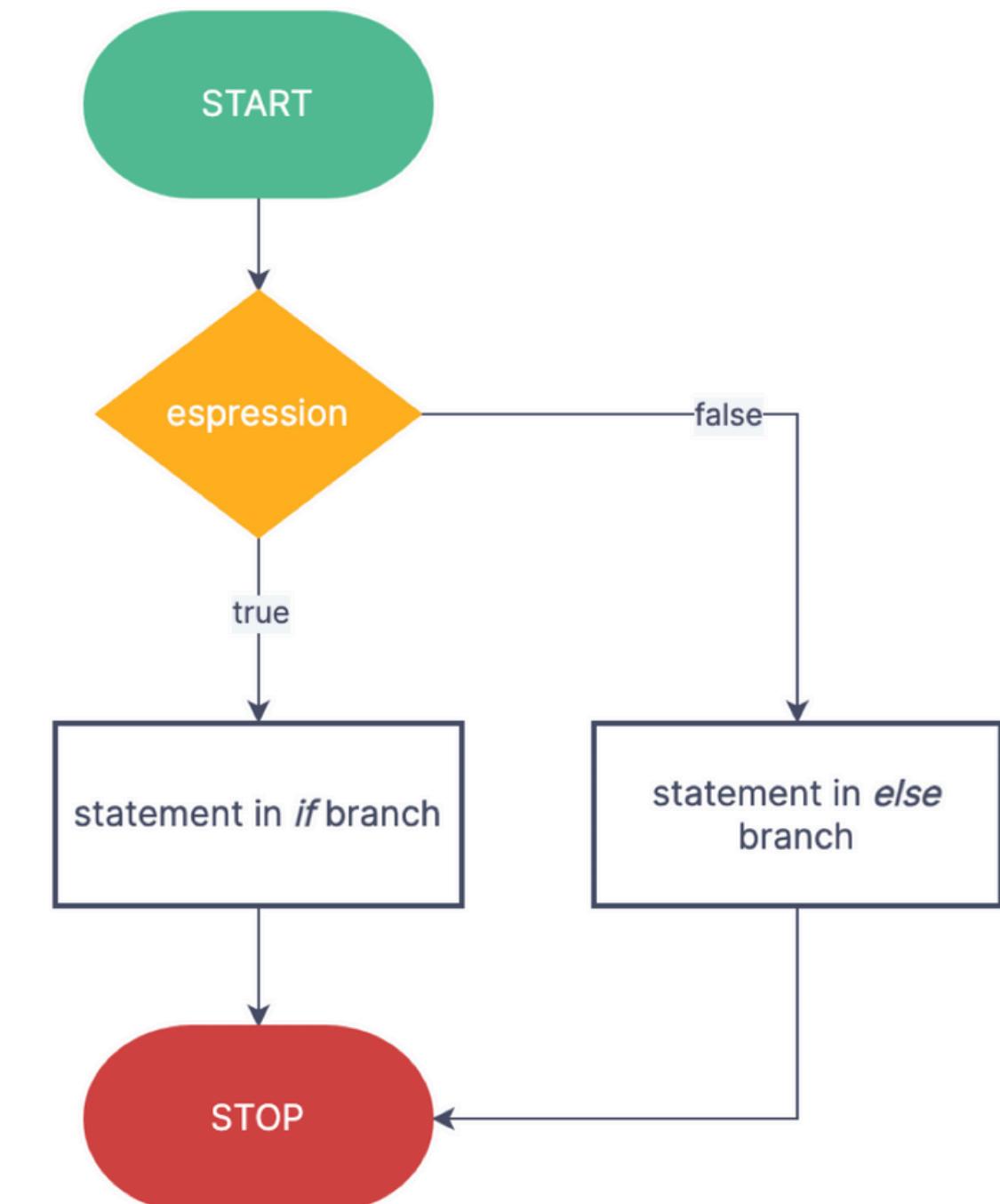


.NET Core App Types. Images courtesy Microsoft.

# NỘI DUNG

- Kiểu dữ liệu boolean (or bit) và các giá trị truthey falsy trong lập trình
- Toán tử so sánh
- Cấu trúc rẽ nhánh if else, switch case, ...
- Ôn tập Hướng dẫn viết console app với câu lệnh điều kiện

```
C# Program.cs ×  
app > C# Program.cs  
1 // Khai báo biến age và gán giá trị ban đầu là 20  
2 int age = 20;  
3  
4 // Kiểm tra nếu age lớn hơn hoặc bằng 18  
5 if (age >= 18)  
6 {  
7     // Nếu điều kiện đúng, in ra "You are an adult."  
8     Console.WriteLine("You are an adult.");  
9 }  
10 else  
11 {  
12     // Nếu điều kiện sai, in ra "You are not an adult."  
13     Console.WriteLine("You are not an adult.");  
14 }
```





# TOÁN TỬ SO SÁNH

## TOÁN TỬ SO SÁNH (COMPARISON OPERATORS).

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

- Trong máy tính nói chung hay lập trình nói riêng để xử lý nhiều trường hợp có thể xảy ra người ta thường dùng 1 kiểu dữ liệu đặc biệt đó là boolean (viết là bool). Nếu như tính toán số học cho ra kết quả là những con số thì các phép tính **so sánh (COMPARISON)**: > , <, >= , <= , == , != sẽ cho ta chỉ 2 giá trị là đúng hoặc sai (Truthy or Falsy).

*Nếu kết quả phép tính số học là trả về 1 giá trị ví dụ:*

**1 + 1 => 2**

*Thì đổi với phép tính so sánh thì kết quả lại trả về là true hoặc false*

**1 == 1 => true**



# TOÁN TỬ SO SÁNH

## MỘT SỐ VÍ DỤ

1

2

3

4

5

6

7

8

9

10

Toán tử so sánh trong C# được sử dụng để so sánh hai giá trị và kết quả trả về sẽ là một giá trị boolean (true hoặc false). Các toán tử này bao gồm:

- **==**: So sánh bằng nhau
- **!=**: So sánh khác nhau
- **>**: So sánh lớn hơn
- **<**: So sánh nhỏ hơn
- **>=**: So sánh lớn hơn hoặc bằng
- **<=**: So sánh nhỏ hơn hoặc bằng

The screenshot shows a code editor with a dark theme. A file named 'Program.cs' is open. The code defines three integers: a = 5, b = 3, and c = 5. It then uses various comparison operators to print out boolean results to the terminal. The code is as follows:

```
C# Program.cs ×
app > C# Program.cs
1 int a = 5;
2 int b = 3;
3 int c = 5;
4 // Phép so sánh bằng (==)
5 Console.WriteLine($"a == c: {a == c}"); // true, vì a bằng c (5 == 5)
6 // Phép so sánh khác (!=)
7 Console.WriteLine($"a != b: {a != b}"); // true, vì a khác b (5 != 3)
8 // Phép so sánh lớn hơn (>)
9 Console.WriteLine($"a > b: {a > b}"); // true, vì a lớn hơn b (5 > 3)
10 // Phép so sánh nhỏ hơn (<)
11 Console.WriteLine($"b < c: {b < c}"); // true, vì b nhỏ hơn c (3 < 5)
12 // Phép so sánh lớn hơn hoặc bằng (>=)
13 Console.WriteLine($"a >= c: {a >= c}"); // true, vì a lớn hơn hoặc bằng c (5 >= 5)
14 // Phép so sánh nhỏ hơn hoặc bằng (<=)
15 Console.WriteLine($"b <= a: {b <= a}"); // true, vì b nhỏ hơn hoặc bằng a (3 <= 5)
```

Below the code editor, there is a terminal window showing the execution of the program. The output is:

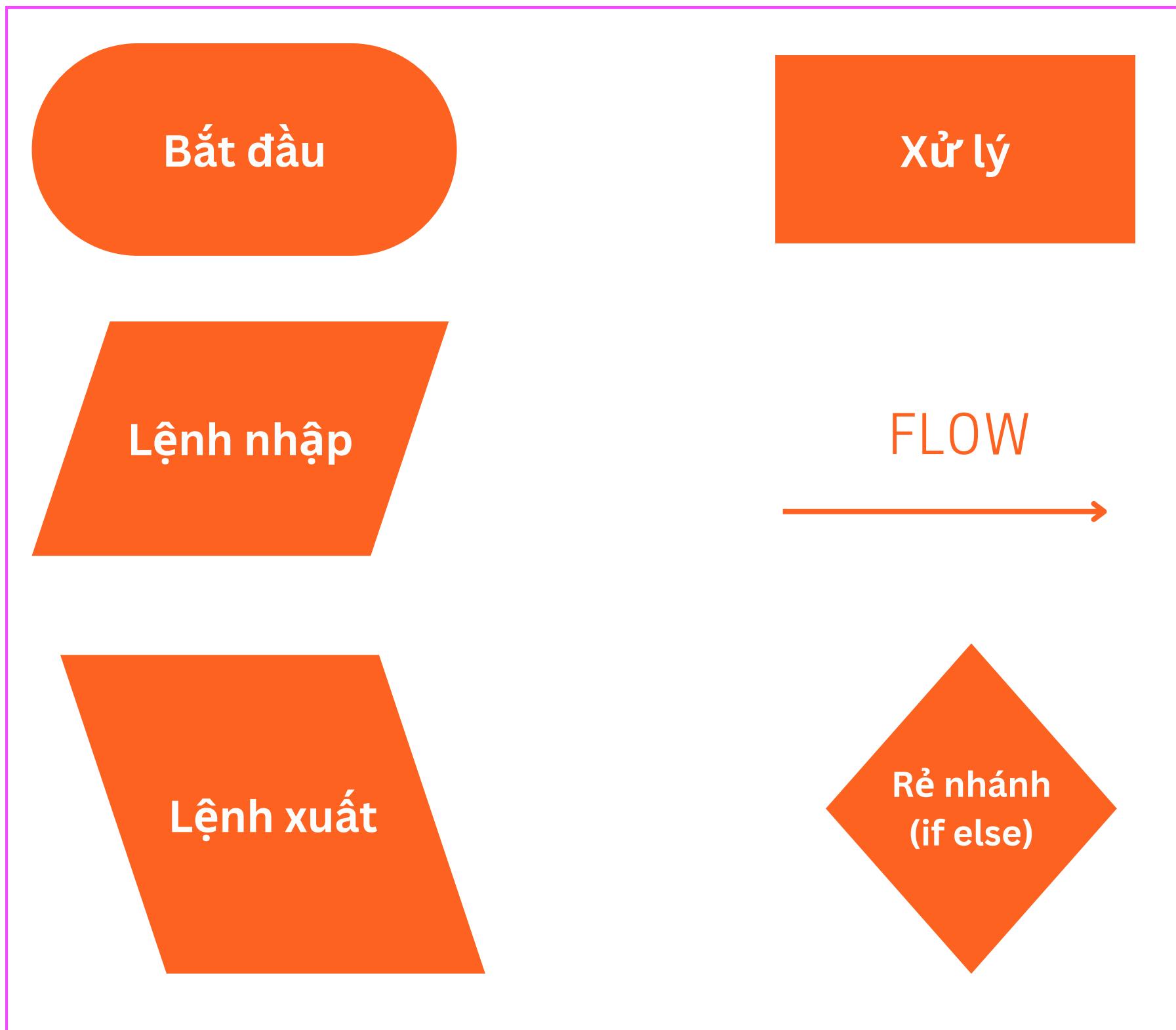
```
khai@MacBook-Pro demo % /Users/khai/.vscode/extensions/ms-dotnettools.csharp-2.39.29-darwin-arm64/.debugger/arm64/vsdbg --interpret
r=vscode --connection=/var/folders/vx/3xbhx751gsclx3l9drmk14h0000gn/T/CoreFxPipe_vsdbg-ui-af28d1950cac4638ae113505c4fc744
a == c: True
a != b: True
a > b: True
b < c: True
a >= c: True
b <= a: True
○ khai@MacBook-Pro demo %
```



# CẤU TRÚC RẺ NHÁNH IF ELSE, SWITCH CASE, ...

## CÙNG LÀM QUEN VỚI LƯU ĐỒ THUẬT TOÁN

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10



Trước khi đi vào áp dụng toán tử so sánh trong cấu trúc rẻ nhánh. Cùng Làm quen với lưu đồ thuật toán.

Lưu đồ `thuật toán giúp bạn hiểu và giải thích quy trình mà không cần quan tâm đến ngôn ngữ lập trình cụ thể, dễ dàng chia sẻ ý tưởng với người khác và chuyển đổi thuật toán sang bất kỳ ngôn ngữ nào sau này.



# CẤU TRÚC RẺ NHÁNH IF ELSE, SWITCH CASE, ...

## CÚ PHÁP BIỂU THỨC IF

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

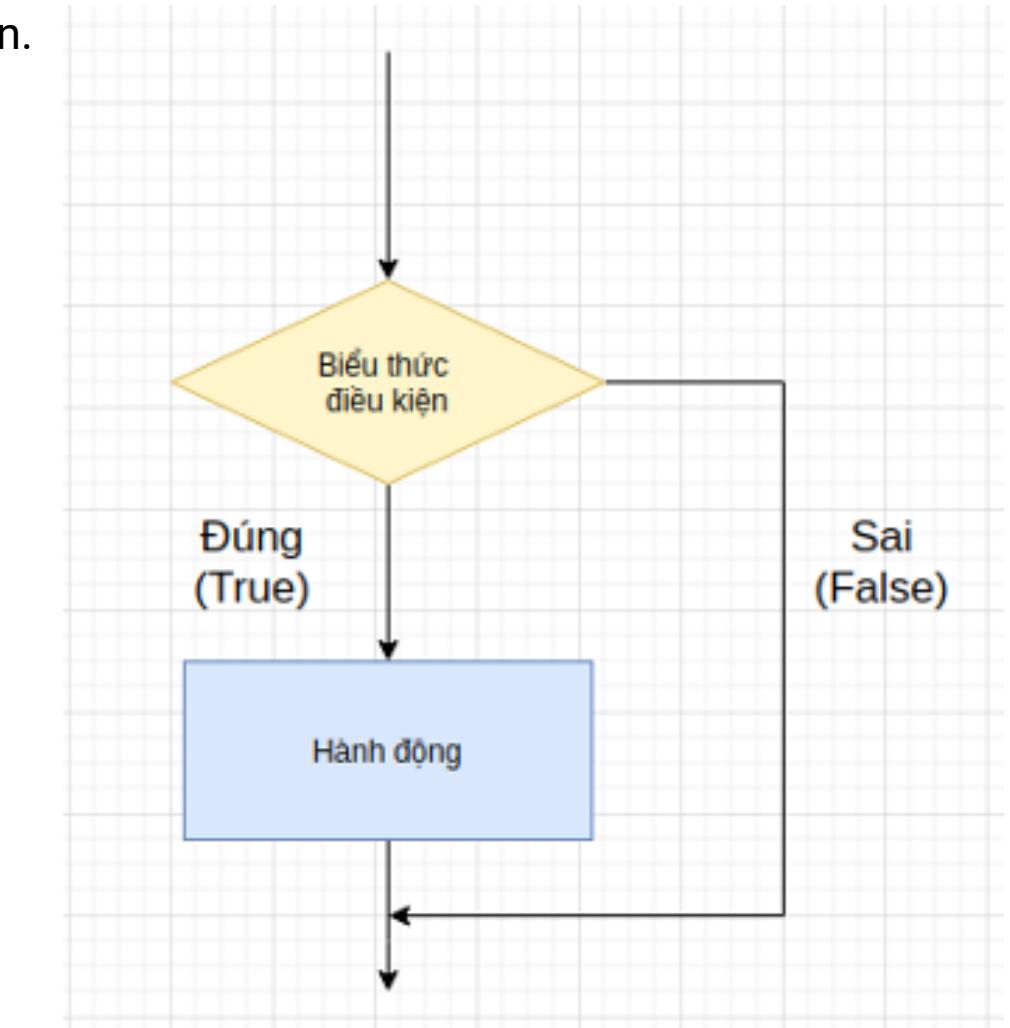
Trước khi đi vào áp dụng toán tử so sánh trong cấu trúc rẻ nhánh. Cùng Làm quen với lưu đồ thuật toán.

### Cú pháp

**if (Biểu thức điều kiện)**

**Hành động**

- Hành động có thể là một dòng hoặc nhóm các xử lý gồm nhiều dòng lệnh trong cặp {}
- Nếu biểu thức điều kiện là đúng (True) thì sẽ thực hiện hành động
- Ngược lại, chương trình sẽ không thực hiện hành động và sẽ chạy các dòng lệnh tiếp theo ngoài if



Flow chart(Lưu đồ)



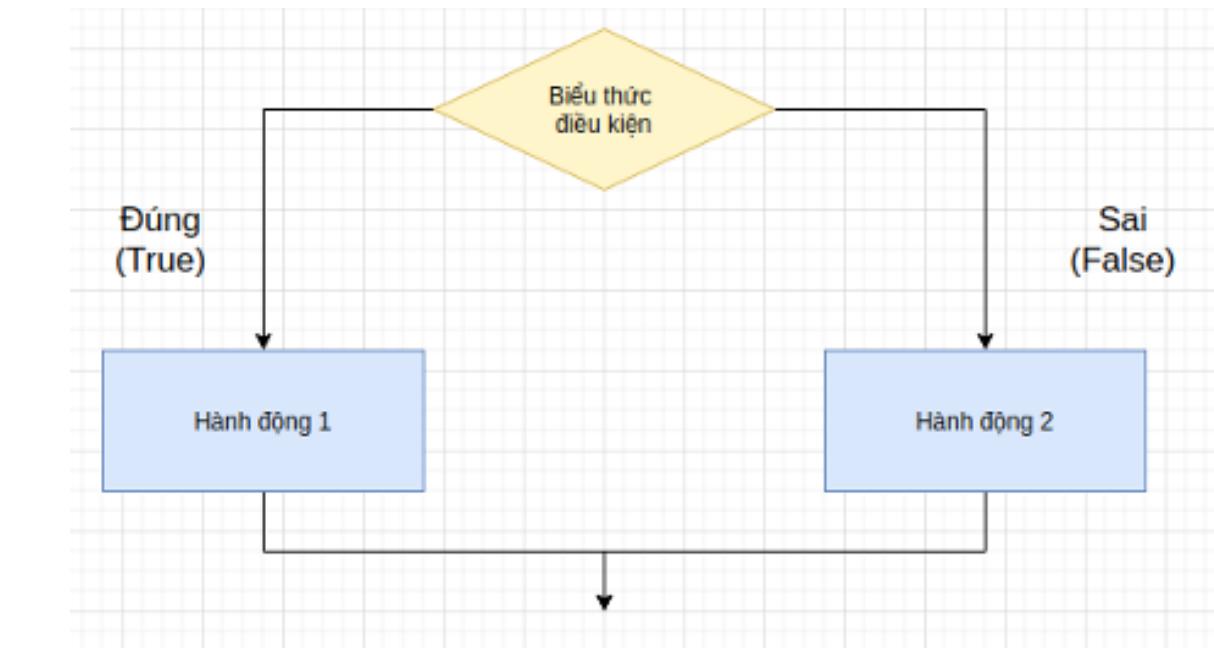
# CẤU TRÚC RẺ NHÁNH IF ELSE, SWITCH CASE, ...

## CÚ PHÁP BIỂU THỨC IF

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

- Cú pháp  
if (*biểu thức điều kiện*)  
*Hành động 1*  
else  
*Hành động 2*

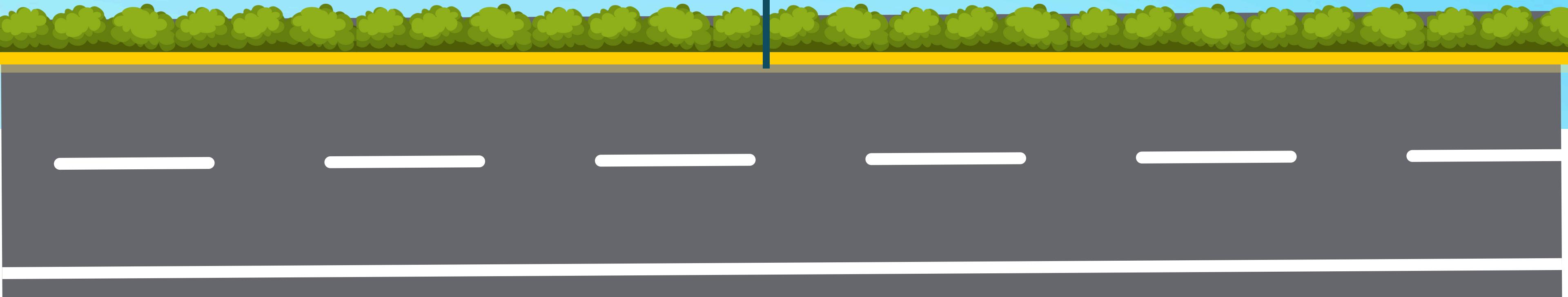
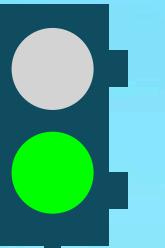
- Nếu biểu thức điều kiện là đúng (True) thì sẽ thực hiện hành động 1 ngược lại sẽ thực hiện hành động 2
- Hành động có thể là một dòng hoặc nhóm các xử lý gồm nhiều dòng lệnh trong cặp {}



Flow chart(Lưu đồ)

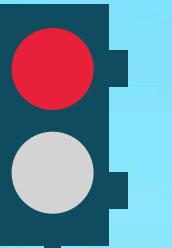


```
if (red signal)  
    car.stop()  
else  
    car.move()
```



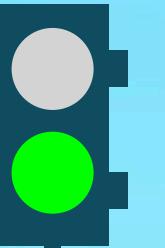


```
if (red signal)  
    car.stop()  
else  
    car.move()
```





```
if (red signal)
    car.stop()
else
    car.move()
```



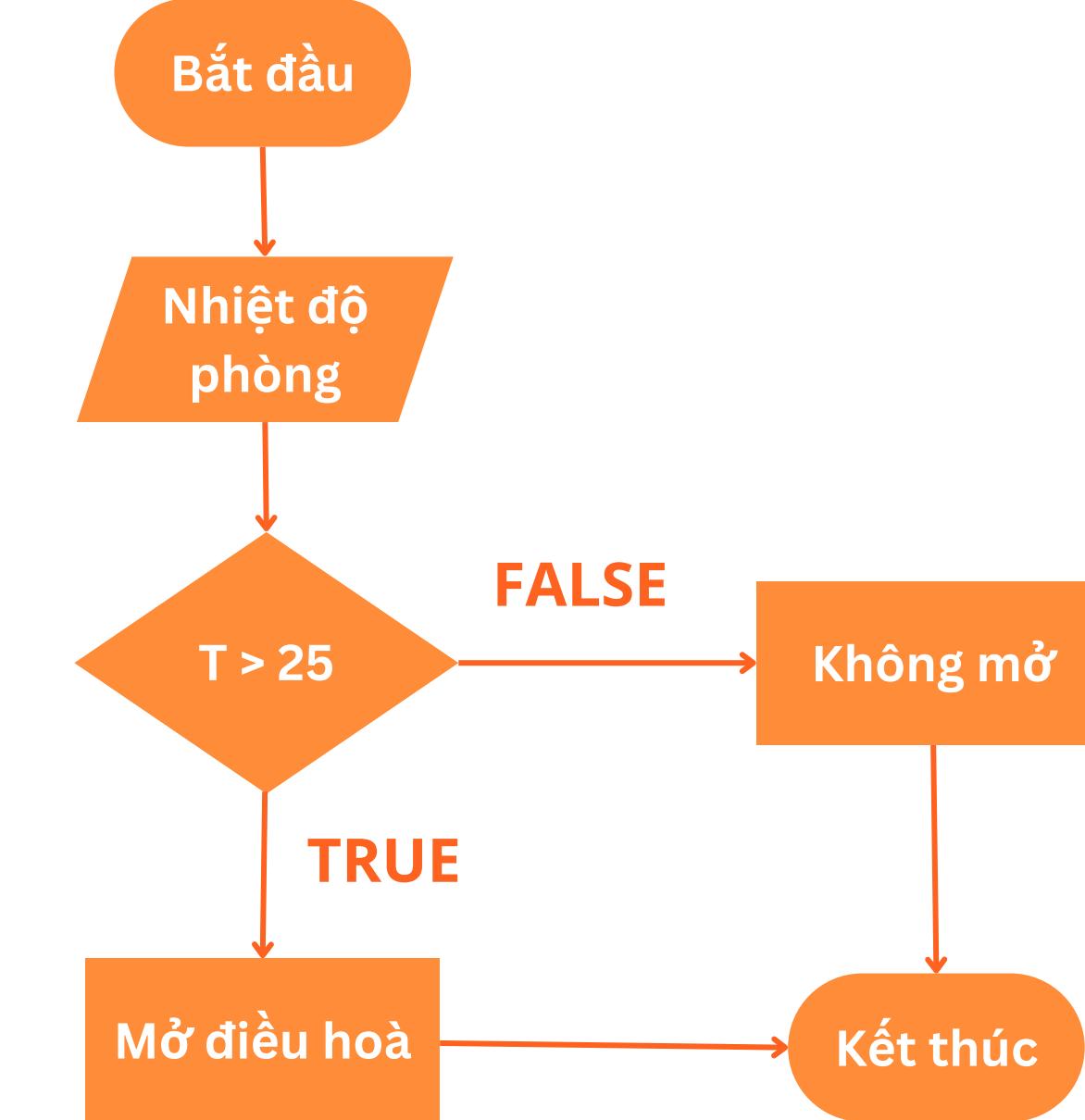


# CẤU TRÚC RẺ NHÁNH IF ELSE, SWITCH CASE, ...

## VÍ DỤ VỀ IF ELSE QUA BÀI TOÁN NHIỆT ĐỘ

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

Viết chương trình nếu  
nhiệt độ phòng dưới  
25 độ thì không cần mở  
điều hòa và ngược lại

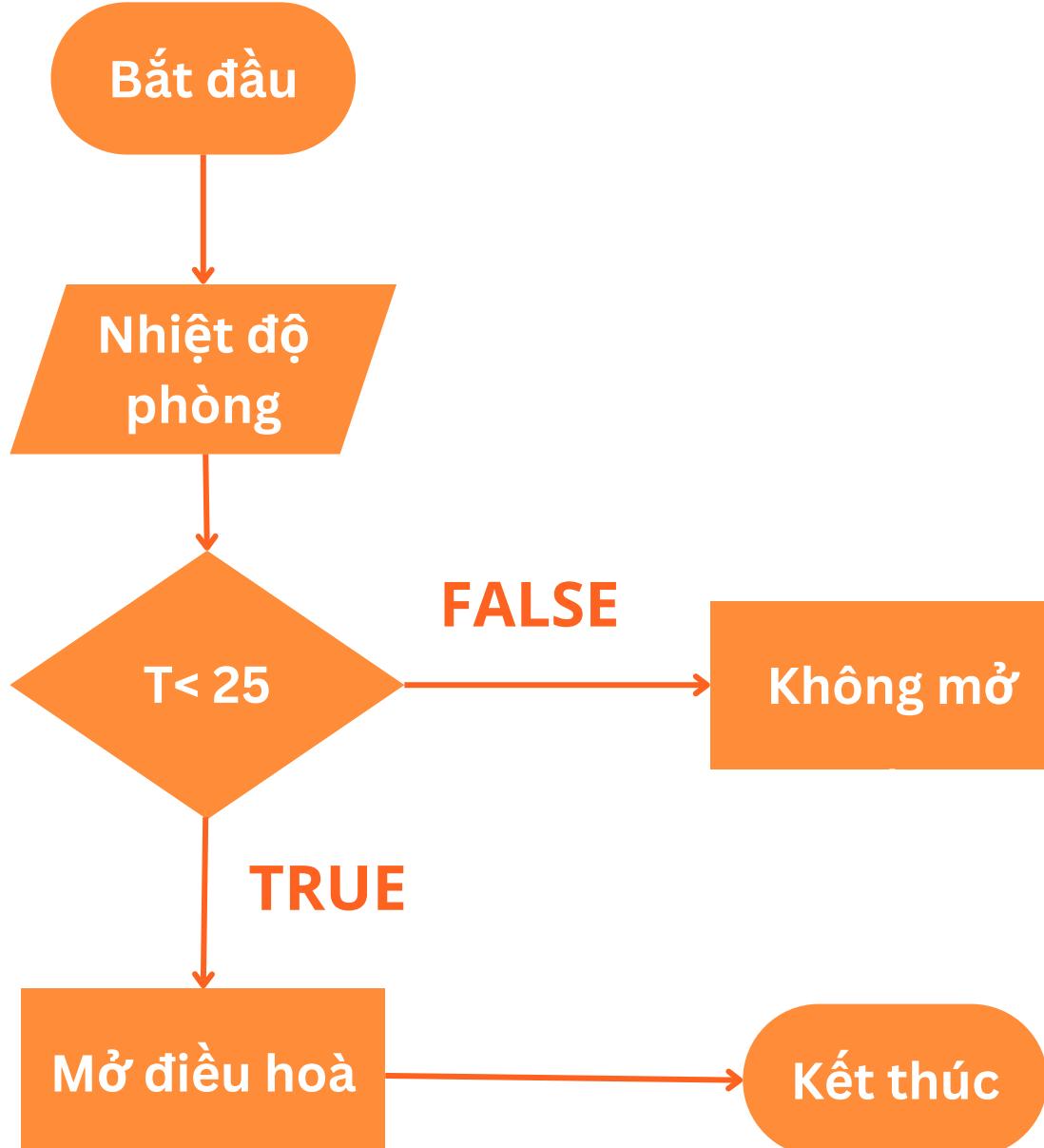




# CẤU TRÚC RẺ NHÁNH IF ELSE, SWITCH CASE, ...

## VÍ DỤ VỀ IF ELSE QUA BÀI TOÁN NHIỆT ĐỘ

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10



```
C# Program.cs ×  
app > C# Program.cs  
1 // Bắt đầu chương trình  
2 Console.WriteLine("Nhập nhiệt độ phòng:");  
3  
4 // Đọc nhiệt độ từ người dùng  
5 double temperature = Convert.ToDouble(Console.ReadLine());  
6  
7 // Kiểm tra nhiệt độ có nhỏ hơn 25 không  
8 if (temperature < 25)  
9 {  
10    Console.WriteLine("Không mở điều hòa.");  
11 }  
12 else  
13 {  
14    Console.WriteLine("Mở điều hòa.");  
15 }  
16  
17 // Kết thúc chương trình  
18 Console.WriteLine("Chương trình kết thúc.");
```



# CẤU TRÚC RẺ NHÁNH IF ELSE, SWITCH CASE, ...

## BÀI TẬP IF ELSE

1

2

3

4

5

6

7

8

9

10

# IF ELSE - LUYỆN TẬP 1

**Yêu cầu:** Hãy lập trình một mini-game đơn giản, trong đó người chơi cần đoán xem con số bất kỳ là chẵn hay lẻ để ghi điểm.

**Hãy viết một chương trình C# cho phép người dùng nhập vào một số nguyên bất kỳ. Chương trình sẽ kiểm tra và in ra kết quả:**

- "Đây là số chẵn" nếu số chia hết cho 2.
- "Đây là số lẻ" nếu không chia hết cho 2.

Viết chương trình kiểm tra xem một số nguyên nhập từ người dùng có phải là số chẵn hay số lẻ không.

1 2 3  
0 5 4  
9 7 6  
8 10



# CẤU TRÚC RẺ NHÁNH IF ELSE, SWITCH CASE, ...

## BÀI TẬP IF ELSE

1

2

3

4

5

6

7

8

9

10

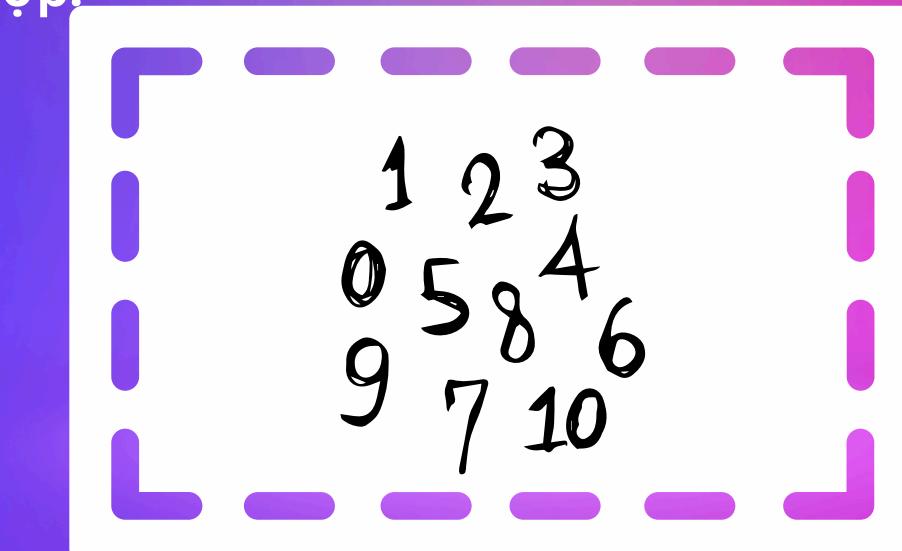
# IF ELSE LUYỆN TẬP 2

**Yêu cầu:** Bạn đang xây dựng một trợ lý ảo đơn giản giúp người dùng chọn trang phục phù hợp theo thời tiết. Người dùng sẽ nhập vào một mã số thời tiết như sau:

- 1: Nắng
- 2: Mưa
- 3: Lạnh
- 4: Tuyết
- Các giá trị khác: Không xác định

**Viết chương trình C# nhận vào mã thời tiết (số nguyên từ bàn phím), sau đó in ra gợi ý trang phục phù hợp.**

Viết chương trình C# nhận vào mã thời tiết (số nguyên từ bàn phím), sau đó in ra gợi ý trang phục phù hợp.





# CẤU TRÚC RẺ NHÁNH IF ELSE, SWITCH CASE, ...

## BÀI TẬP IF ELSE

# IF ELSE LUYỆN TẬP 3

**Yêu cầu:** Một công ty trả lương cho nhân viên theo quy định sau:

- Lương cơ bản được tính theo giờ làm việc.
- Nếu nhân viên làm trên 40 giờ trong tuần, phần vượt quá 40 giờ sẽ được tính là tăng ca, với mức lương gấp 1.5 lần lương cơ bản.

**Bạn hãy viết chương trình C# cho phép người dùng nhập vào:**

- **Lương cơ bản theo giờ (kiểu double)**
- **Số giờ làm việc trong tuần (kiểu double)**

**Sau đó, chương trình sẽ tính và hiển thị ra lương thực tế của nhân viên.**

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10





# CẤU TRÚC RẺ NHÁNH IF ELSE, SWITCH CASE, ...

## KHÔI LỆNH IF ELSE IF MỞ RỘNG

1

2

3

4

5

6

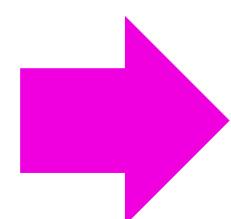
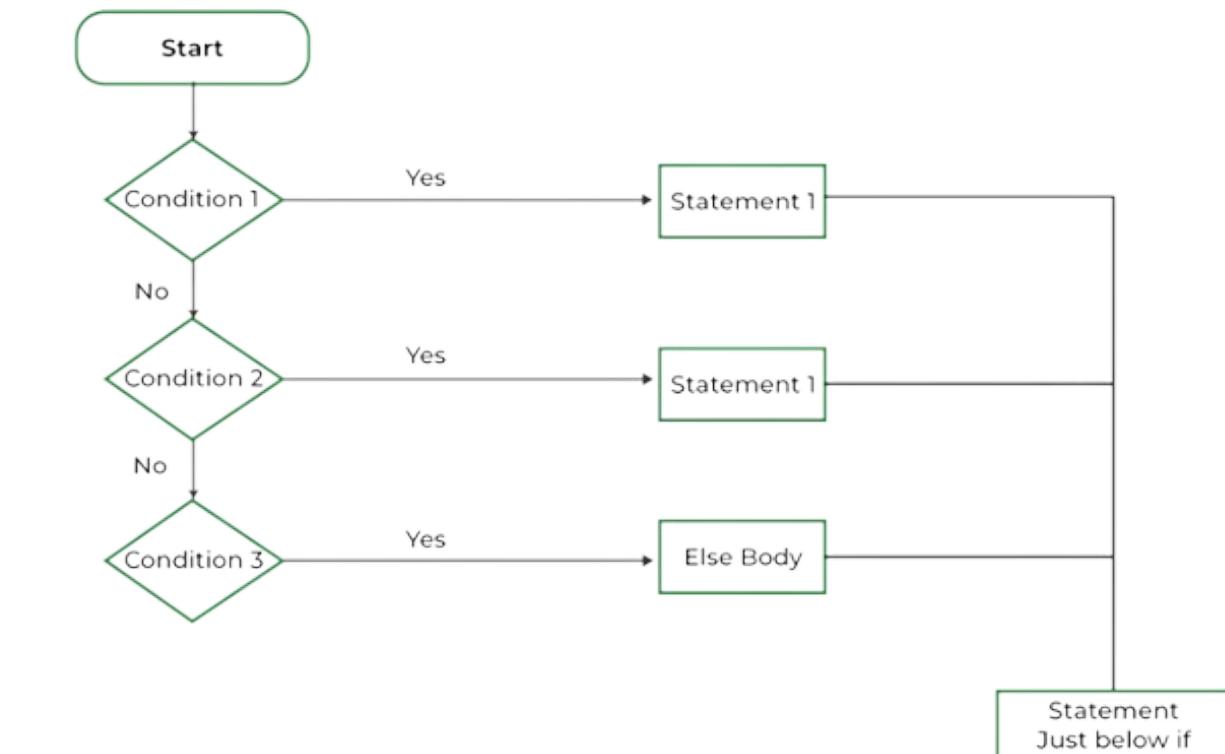
7

8

9

10

```
javascript  
  
if (điềuKiện1) {  
    // mã sẽ được thực thi nếu điềuKiện1 là đúng  
} else if (điềuKiện2) {  
    // mã sẽ được thực thi nếu điềuKiện1 là sai và điềuKiện2 là đúng  
} else {  
    // mã sẽ được thực thi nếu cả điềuKiện1 và điềuKiện2 đều sai  
}
```

[Sao chép mã](#)

- if:** Sử dụng để kiểm tra một điều kiện. Nếu điều kiện đúng (true), khối lệnh trong if sẽ được thực thi.
- else if:** Sử dụng để kiểm tra điều kiện khác nếu điều kiện trong if không đúng.
- else:** Sử dụng để thực thi một khối lệnh nếu tất cả các điều kiện trên không đúng.



# CẤU TRÚC RẺ NHÁNH IF ELSE, SWITCH CASE, ...

## KHÔI LỆNH IF ELSE IF MỞ RỘNG

1

2

3

4

5

6

7

8

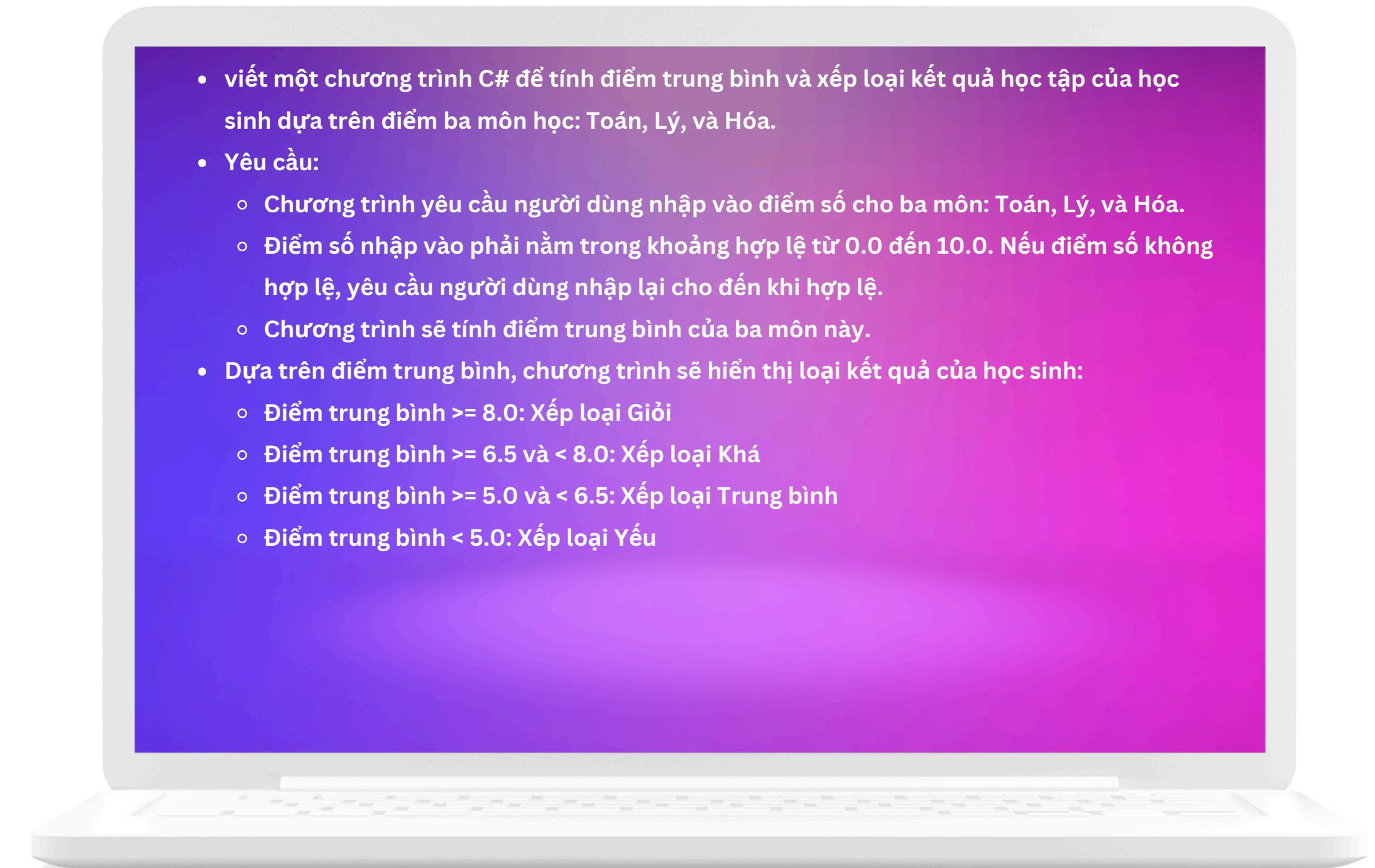
9

10

## IF ELSE

## LUYỆN TẬP 4

- viết một chương trình C# để tính điểm trung bình và xếp loại kết quả học tập của học sinh dựa trên điểm ba môn học: Toán, Lý, và Hóa.
- Yêu cầu:**
  - Chương trình yêu cầu người dùng nhập vào điểm số cho ba môn: Toán, Lý, và Hóa.
  - Điểm số nhập vào phải nằm trong khoảng hợp lệ từ 0.0 đến 10.0. Nếu điểm số không hợp lệ, yêu cầu người dùng nhập lại cho đến khi hợp lệ.
  - Chương trình sẽ tính điểm trung bình của ba môn này.
- Dựa trên điểm trung bình, chương trình sẽ hiển thị loại kết quả của học sinh:
  - Điểm trung bình  $\geq 8.0$ : Xếp loại Giỏi
  - Điểm trung bình  $\geq 6.5$  và  $< 8.0$ : Xếp loại Khá
  - Điểm trung bình  $\geq 5.0$  và  $< 6.5$ : Xếp loại Trung bình
  - Điểm trung bình  $< 5.0$ : Xếp loại Yếu





# CẤU TRÚC RẺ NHÁNH IF ELSE, SWITCH CASE, ...

KHÔI LỆNH IF ELSE IF MỞ RỘNG

## IF ELSE LUYỆN TẬP 5

**Yêu cầu:** Bạn đang xây dựng một mini game tính nhanh dành cho trẻ em. Luật chơi yêu cầu người chơi nhập vào một số có đúng 3 chữ số (ví dụ: 123, 456, 987...). Hệ thống sẽ tự động tách và tính tổng từng chữ số trong số đó.

**Viết chương trình C# thực hiện các yêu cầu sau:**

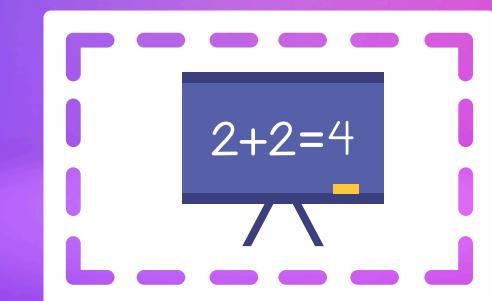
1. Nhập vào một số nguyên từ bàn phím.
2. Nếu số không có đúng 3 chữ số (tức  $< 100$  hoặc  $> 999$ ), in ra: "Số không hợp lệ. Vui lòng nhập số có đúng 3 chữ số."
3. Nếu hợp lệ, tách từng chữ số và tính tổng, sau đó in kết quả theo định dạng: Kết quả:  $9 + 8 + 7 = 24$

Nhập vào 1 số có 3 chữ số tính tổng 3 ký số đó.

Ví dụ:

Nhập vào 987 =>  $9 + 8 + 7 = 24$

Nhập vào số có 2 chữ số hoặc 1 chữ số => Không hợp lệ



- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10



# CẤU TRÚC RẺ NHÁNH IF ELSE, SWITCH CASE, ...

## CẤU TRÚC SWITCH CASE

1

2

3

4

5

6

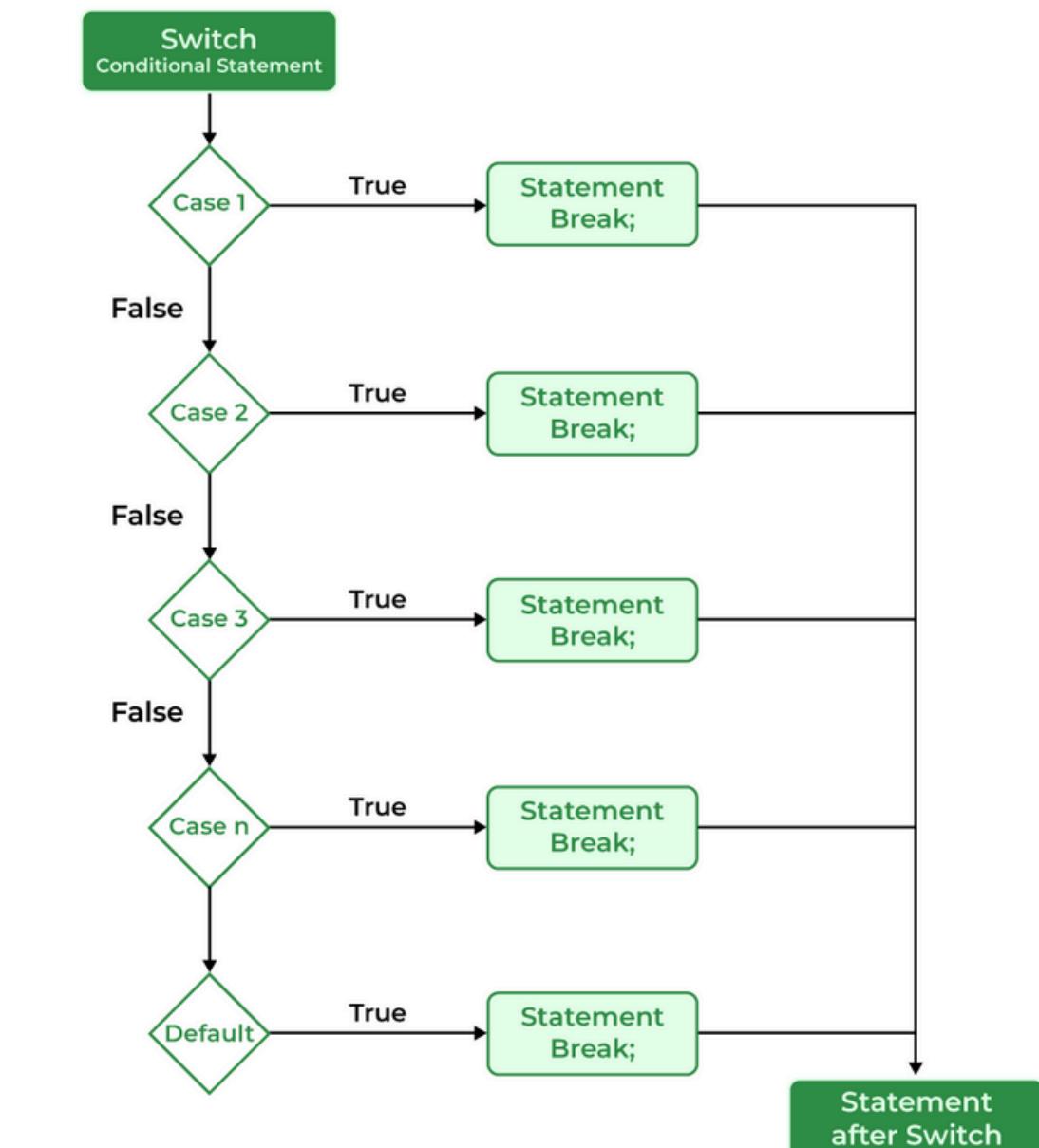
7

8

9

10

```
switch(expression) {  
    case value1:  
        // Thực thi khi expression === value1  
        break;  
    case value2:  
        // Thực thi khi expression === value2  
        break;  
    // Có thể có nhiều case khác  
    default:  
        // Thực thi khi không có case nào khớp  
}
```



## Sử dụng switch expression:

```
var result = expression switch  
{  
    value1 => result1, // Thực thi khi expression == value1  
    value2 => result2, // Thực thi khi expression == value2  
    // Có thể thêm nhiều case khác  
    _ => defaultResult // Thực thi khi không có case nào khớp  
};  
  
// Thực thi kết quả  
Console.WriteLine(result);
```



# CẤU TRÚC RẺ NHÁNH IF ELSE, SWITCH CASE, ...

## BÀI TẬP SWITCH CASE

1

2

3

4

5

6

7

8

9

10

**Yêu cầu:** Bạn đang lập trình một tính năng nhỏ cho một ứng dụng quản lý công việc cá nhân. Khi người dùng nhập vào một số từ 1 đến 7, hệ thống sẽ hiển thị tên ngày trong tuần tương ứng để phục vụ sắp xếp lịch.

**Hãy viết chương trình C# cho phép người dùng:**

1. **Nhập vào một số nguyên từ 1 đến 7.**
2. **In ra tên ngày trong tuần tương ứng**
3. **Nếu người dùng nhập số ngoài phạm vi 1-7, hiển thị thông báo: "Giá trị không hợp lệ. Vui lòng nhập số từ 1 đến 7."**



# CẤU TRÚC RẺ NHÁNH IF ELSE, SWITCH CASE, ...

## SWITCH EXPRESSION

1

Bài tập: Xây dựng chương trình quản lý điểm học sinh

2

Mô tả: Bạn cần xây dựng một chương trình nhận điểm số của học sinh và in ra xếp loại tương ứng dựa trên thang điểm chữ. Cụ thể:

3

- Điểm A: Từ 90 đến 100
- Điểm B: Từ 80 đến 89
- Điểm C: Từ 70 đến 79
- Điểm D: Từ 60 đến 69
- Điểm F: Dưới 60

4

Yêu cầu:

5

- Chương trình sẽ nhận vào một điểm số từ 0 đến 100.
- Sử dụng switch...case để xác định xếp loại (A, B, C, D, F) dựa trên điểm số.
- In ra kết quả xếp loại của học sinh.

6

7

8

9

10



# CẤU TRÚC RẺ NHÁNH IF ELSE, SWITCH CASE, ...

## SWITCH EXPRESSION

1

Bài tập: Xây dựng chương trình quản lý điểm học sinh

2

Mô tả: Bạn cần xây dựng một chương trình nhận điểm số của học sinh và in ra xếp loại tương ứng dựa trên thang điểm chữ. Cụ thể:

3

- Điểm A: Từ 90 đến 100
- Điểm B: Từ 80 đến 89
- Điểm C: Từ 70 đến 79
- Điểm D: Từ 60 đến 69
- Điểm F: Dưới 60

4

Yêu cầu:

5

- Chương trình sẽ nhận vào một điểm số từ 0 đến 100.
- Sử dụng switch...case để xác định xếp loại (A, B, C, D, F) dựa trên điểm số.
- In ra kết quả xếp loại của học sinh.

6

7

8

9

10



# CẤU TRÚC RẺ NHÁNH IF ELSE, SWITCH CASE, ...

## LOGICAL OPERATORS

1

- A&&B đúng khi tất cả đều đúng
- A||B đúng khi một trong 2 đúng

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

HTTP://CYBERSOFT.EDU.VN

- a == b : So sánh bằng. Trả về True Nếu a và b cùng giá trị
- a!=b : So sánh khác . Trả về True Nếu a và b khác giá trị

2

3

4

5

6

7

8

9

10

```
C# Program.cs X
app > C# Program.cs
1 // Toán tử logic
2 // Toán tử AND logic (&&): Kết quả là true nếu cả hai biểu thức đều true
3 bool andResult = (5 > 3) && (8 > 6); // true && true => true
4 // Toán tử OR logic (||): Kết quả là true nếu ít nhất một trong hai biểu thức là true
5 bool orResult = (5 > 3) || (8 < 6); // true || false => true
6 // Toán tử NOT logic (!): Đảo ngược giá trị logic của biểu thức
7 bool notResult = !(5 > 3); // !true => false
8
9
10 // Ngoài ra còn có các toán tử mở rộng khác: Toán tử bitwise
11 // Toán tử AND bitwise (&): Thực hiện phép AND trên từng bit của các toán hạng
12 int bitwiseAndResult = 5 & 3; // 0101 & 0011 => 0001 (kết quả: 1)
13 // Toán tử OR bitwise (|): Thực hiện phép OR trên từng bit của các toán hạng
14 int bitwiseOrResult = 5 | 3; // 0101 | 0011 => 0111 (kết quả: 7)
15 // Toán tử XOR bitwise (^): Thực hiện phép XOR trên từng bit của các toán hạng
16 int bitwiseXorResult = 5 ^ 3; // 0101 ^ 0011 => 0110 (kết quả: 6)
17 // Toán tử NOT bitwise (~): Thực hiện phép NOT trên từng bit của toán hạng (đảo ngược các bit)
18 int bitwiseNotResult = ~5; // ~0101 => 1010 (kết quả: -6 trong hệ nhị phân bù 2)
```



# CẤU TRÚC RẺ NHÁNH IF ELSE, SWITCH CASE, ...

## OTHER OPERATORS

1

2

3

4

5

6

7

8

9

10

### 1. Biểu thức ba ngôi (Ternary Operator)

Biểu thức ba ngôi là một cách viết gọn của câu lệnh if...else.

Cú pháp: **condition ? expression1 : expression2;**

Nếu condition đúng (true), expression1 sẽ được thực thi; nếu không, expression2 sẽ được thực thi.

```
int a = 10;
int b = 20;

// Sử dụng biểu thức ba ngôi để xác định giá trị lớn hơn
int max = (a > b) ? a : b;
Console.WriteLine("Giá trị lớn hơn là: " + max); // Output: 20
```

Ví dụ:

```
1 // Khai báo biến nhiệt độ
2 let nhietDo = 30;
3 // Kiểm tra nhiệt độ và xác định trạng thái thời tiết
4 let trangThai = (nhietDo > 25) ? "Nóng" : "Mát"; // (điều kiện) ? nếu đúng : nếu sai
5 // In ra kết quả
6 console.log(trangThai); // Kết quả sẽ là "Nóng"
```



# CẤU TRÚC RẺ NHÁNH IF ELSE, SWITCH CASE, ...

## OTHER OPERATORS

1

2

3

4

5

6

7

8

9

10

### 2. Toán tử ?? (Null-coalescing Operator)

- Toán tử ?? được sử dụng để trả về toán hạng đầu tiên nếu nó không phải là null; nếu không, nó sẽ trả về toán hạng thứ hai.
- Cú pháp: **expression1 ?? expression2**

```
string name = null;
string displayName = name ?? "Guest";
Console.WriteLine("Hello, " + displayName); // Output: Hello, Guest
```

### 3. Toán tử ??= (Null-coalescing Assignment)

Toán tử ??= gán giá trị cho biến chỉ khi biến đó đang là null.

Cú pháp: **variable ??= value**

```
string name = null;
name ??= "Default Name";
Console.WriteLine(name); // Output: Default Name
```



# CẤU TRÚC RẺ NHÁNH IF ELSE, SWITCH CASE, ...

## OTHER OPERATORS

1

2

3

4

5

6

7

8

9

10

### 2. Toán tử ?? (Null-coalescing Operator)

Toán tử ?? được sử dụng để trả về toán hạng đầu tiên nếu nó không phải là null; nếu không, nó sẽ trả về toán hạng thứ hai.

Cú pháp: expression1 ?? expression2

```
string name = null;
string displayName = name ?? "Guest";
Console.WriteLine("Hello, " + displayName); // Output: Hello, Guest
```