

Bài 3

CẤU TRÚC ĐIỀU KHIỂN

Company: DEVPRO VIỆT NAM

Website: devpro.edu.vn

Design by Minh An

Nội dung

- **Cấu trúc rẽ nhánh**
if..., if...else..., switch...case...
- **Cấu trúc lặp**
for..., while..., do...while...
- **Các lệnh chuyển điều khiển**
break, continue, return.

Design by Minh An

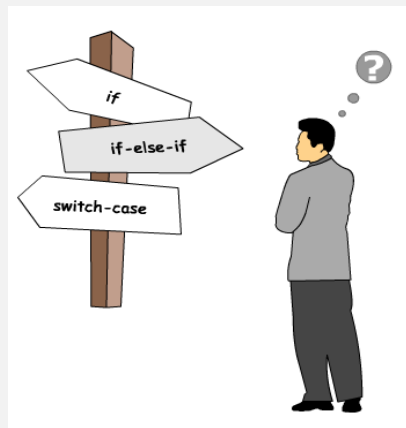
Cấu trúc điều khiển

- Ra quyết định thực hiện một khối lệnh khi có một điều kiện đúng (rẽ nhánh).
- Lặp lại một khối lệnh khi một điều kiện còn đúng (vòng lặp).
- Tất cả các môi trường phát triển ứng dụng đều cung cấp một cách thức ra quyết định (decision - making) được gọi là các câu lệnh điều khiển luồng mà nó chỉ đạo thực thi ứng dụng.

Design by Minh An

1. Các cấu trúc điều khiển rẽ nhánh

- Java hỗ trợ các loại câu lệnh điều khiển rẽ nhánh sau:
 - Câu lệnh *if*
 - Câu lệnh *if-else-if*
 - Câu lệnh *switch-case*



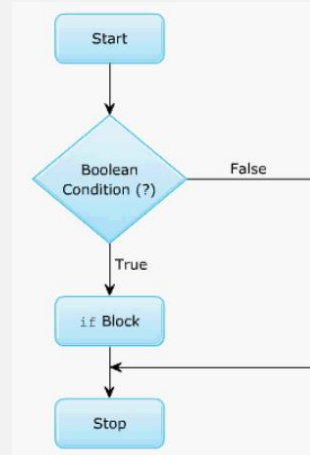
Design by Minh An

1.1. Cấu trúc rẽ nhánh if

- Cú pháp:

```
if (condition) {  
    statements;  
}
```

- Ví dụ:
 - Thực hiện phép chia số a cho số b.
 - Số b khác 0 là đúng: quyết định chia số a cho số b.



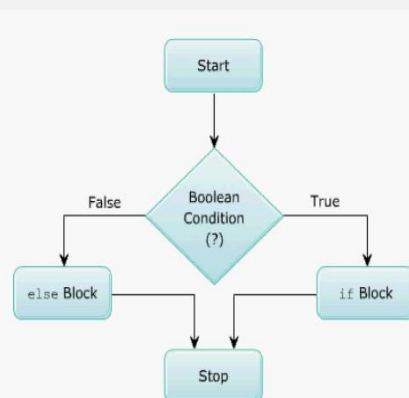
Design by Minh An

1.1.2. Cấu trúc rẽ nhánh if...else...

- Câu lệnh if-else kiểm tra kết quả của một điều kiện và thực thi hành động thích hợp dựa trên kết quả đó.
- Cấu trúc của câu lệnh if-else

```
if (condition) {  
    statement_T;  
} else {  
    statement_F;  
}
```

- Ví dụ:** Kiểm tra một số nguyên là số chẵn hay số lẻ.



Design by Minh An

1.1.3. Cấu trúc rẽ nhánh if...else...if...else...

- Cấu trúc đa câu lệnh *if* được biết đến như là *if-else-if* từng bậc
- Khi điều kiện đúng được tìm thấy, câu lệnh được liên kết với điều kiện đúng được thực thi.

```
if (condition_1) {  
    statement_1;  
} else if(condition_2) {  
    statement_2;  
} else {  
    statement_F;  
}
```

- Ví dụ: bài toán tính tiền điện sinh hoạt của một hộ gia đình trong một tháng.

Design by Minh An

TT	Nhóm đối tượng khách hàng	Giá bán điện (đồng/kWh)
1	Giá bán lẻ điện sinh hoạt	
	Bậc 1: Cho kWh từ 0 - 50	1.678
	Bậc 2: Cho kWh từ 51 - 100	1.734
	Bậc 3: Cho kWh từ 101 - 200	2.014
	Bậc 4: Cho kWh từ 201 - 300	2.536
	Bậc 5: Cho kWh từ 301 - 400	2.834
	Bậc 6: Cho kWh từ 401 trở lên	2.927
2	Giá bán lẻ điện sinh hoạt dùng công tơ thẻ trả trước	2.461

Translated from APTECH WORLD
WIDE book by Duong Thanh
Minh/ 3/ 8 of 17

1.1.4. Cấu trúc lựa chọn switch...case...



- Câu lệnh **switch** – **case** được sử dụng khi một biến cần phải được so sánh trở lại với các giá trị khác.
- Câu lệnh **switch** thực thi **case** tương ứng với giá trị của biểu thức.

```
switch (expression){  
    case value_1:  
        statement_1; break;  
    case value_2:  
        statement_2; break;  
    ...  
    case value_n:  
        statement_n; break;  
    default:  
        statement_F;  
}
```

Design by Minh An

Cấu trúc lựa chọn switch...case...



```
int day = 4;  
String str;  
switch (day){  
    case 0: str = "Sunday"; break;  
    case 1: str = "Monday"; break;  
    case 2: str = "Tuesday"; break;  
    case 3: str = "Wednesday"; break;  
    case 4: str = "Thursday"; break;  
    case 5: str = "Friday"; break;  
    case 6: str = "Saturday"; break;  
    default: str = "Invalid day";  
}  
System.out.println(str);
```

Design by Minh An

2. Cấu trúc lặp



- Các câu lệnh lặp được hỗ trợ bởi ngôn ngữ lập trình Java là:
 - `while ...`
 - `do ... while ...`
 - `for ...`

Design by Minh An

2.1. Cấu trúc lặp while



- Câu lệnh `while` được sử dụng để thực thi một hay nhiều câu lệnh trong khi điều kiện liên quan là “Đúng”(true).
- Điều kiện được kiểm tra trước khi các câu lệnh được thực thi.
- Cú pháp:

```
while (bool_expression)
{
    statements;
}
```
- Ví dụ: **Đếm số chữ số của một số nguyên dương.**

Design by Minh An

Cấu trúc lặp while



- Các luật:
 - Các biến được sử dụng trong biểu thức điều kiện (**bool_expression**) cần phải được khởi tạo ở trước vòng lặp.
 - Thân của vòng lặp phải có một lệnh mà nó làm thay đổi giá trị của biến trong biểu thức điều kiện.

Design by Minh An

Cấu trúc lặp do ... while



- Câu lệnh *do-while* kiểm tra điều kiện ở cuối của vòng lặp thay vì ở phía đầu để chắc chắn rằng vòng lặp được thực thi ít nhất một lần.
- Cú pháp:

```
do{  
    statements;  
}while (boolean condition);
```
- Ví dụ: Kiểm tra tính đúng đắn của dữ liệu nhập vào.

Design by Minh An

Cấu trúc lặp for



- Cấu trúc lặp **for** giống như cấu trúc lặp **while** ở chức năng của nó.
- Khi số lần lặp được biết trước, câu lệnh for được sử dụng.
- Cú pháp:**

```
for ([exp1] ; [exp2] ; [exp3]) {  
    statements ;  
}
```

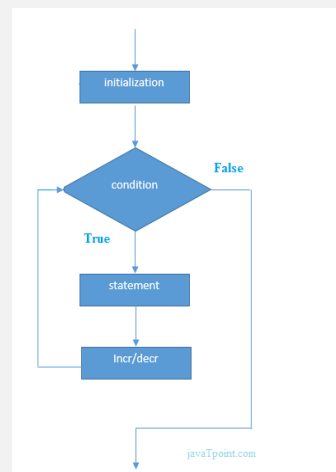
- Trong đó:
 - exp1** là biểu thức khởi tạo giá trị của biến điều khiển vòng lặp.
 - exp2** là biểu thức kiểm tra điều kiện dừng vòng lặp.
 - exp3** là biểu thức định nghĩa cách thức thay đổi giá trị của biến điều khiển vòng lặp.

Design by Minh An

Cấu trúc lặp for



- Lệnh khởi tạo (**exp1**) sẽ được chạy trước tiên và chỉ 1 lần duy nhất
- Sau đó biểu thức điều kiện (**exp2**) sẽ được xem xét, nếu điều kiện đúng thì khối lệnh trong thân vòng lặp sẽ được thực thi. Nếu điều kiện sai thì thoát vòng lặp
- Sau khi thực thi xong khối lệnh trong thân vòng lặp, vòng for quay trở lên, thực hiện biểu thức thay đổi giá trị biến điều khiển vòng lặp (**exp3**)
- Lặp lại bước 2.



Design by Minh An

Cấu trúc lặp for



- Ví dụ

```
class ForDemo {  
    public static void main(String [] args){  
        int count = 1, sum = 0;  
        for (count = 1; count <= 10; count += 2){  
            sum += count;  
        }  
        System.out.println("Tổng của 5 số lẻ đầu tiên là : " + sum);  
    }  
}
```

Design by Minh An

3. Các cấu trúc lặp lồng nhau



- Đặt một vòng lặp bên trong thân của một vòng lặp khác được gọi là lồng nhau (nested loops).
- Khi lồng hai vòng lặp, vòng lặp bên ngoài sẽ điều khiển số lần thực thi vòng lặp bên trong.
- Các vòng lặp lồng nhau hay dùng nhất là vòng lặp *for*.

Design by Minh An

4. Các lệnh chuyển điều khiển



- `break;`
- `continue;`
- `return;`

Design by Minh An

5. Bài tập



Bài tập 1: Cho các bài toán

Bài toán 1: Giải phương trình bậc 2 $ax^2 + bx + c = 0$ ($a \neq 0$)

Bài toán 2: Xếp loại học tập của học sinh dựa vào điểm tổng kết theo cách sau:

- Xếp loại giỏi nếu điểm tổng kết từ 8.00 trở lên.
- Xếp loại khá nếu điểm tổng kết từ 7.00 tới cận 8.00.
- Xếp loại trung bình nếu điểm tổng kết từ 5.00 tới cận 7.00.
- Còn lại, xếp loại yếu.

Bài toán 3: Tính tiền lương thực lĩnh của nhân viên gồm lương cứng (LC) và thưởng (phạt) theo doanh số bán hàng (DS). Thưởng (phạt) được tính như dưới đây:

- Nếu $DS < 50000000$ thì bị trừ 10% LC, không được thưởng.
- Nếu $50000000 \leq DS \leq 100000000$ thì không bị trừ LC, không được thưởng.
- Nếu $100000000 \leq DS \leq 150000000$ thì không bị trừ LC, được thưởng 5%DS.
- Nếu $DS > 150000000$ thì không bị trừ LC, được thưởng 10%DS.

Bài toán 4: Tính số ngày trong tháng của 1 tháng theo giá trị tháng và năm (dương lịch) nhập vào từ bàn phím.

Viết chương trình cho phép người sử dụng lựa chọn để thực hiện 1 trong 4 bài toán trên.

- Chọn 1: Thực hiện bài toán 1.
- Chọn 2: Thực hiện bài toán 2.
- Chọn 3: Thực hiện bài toán 3.
- Chọn 4: Thực hiện bài toán 4.
- Chọn 0: Thoát khỏi chương trình.

Nếu người sử dụng không chọn 1 trong 5 lựa chọn trên thì thông báo chọn sai và thoát chương trình.

Design by Minh An

Bài tập



Bài tập 2: Cho các bài toán

Bài toán 1: Nhập 1 chữ số và đọc chữ số đó. Ví dụ 1 đọc là “số một”.

Bài toán 2: Tính giá trị biểu thức

$$y = \begin{cases} 5\cos(3x+2) - \ln(x+2) & \text{khi } 2 < x < 3 \\ (x+3)^2 & \text{khi } x \geq 3 \\ 1 & \text{với các trường hợp còn lại} \end{cases}$$

Bài toán 3: Tính tiền mua gạo

Một cửa hàng bán gạo với đơn giá bán được tính như sau: Nếu mua từ 1kg đến 50kg thì đơn giá là 14000 đồng/1kg, nếu mua nhiều hơn 50kg thì đơn giá cho phần mua thêm (ngoài 50kg) được giảm như sau:

- Phần thêm từ 01kg đến 25 kg được giảm 500 đồng/1kg.
- Phần thêm từ 26kg đến 40 kg được giảm 750 đồng/1kg.
- Phần thêm từ 41kg đến 55 kg được giảm 1000 đồng/1kg.
- Trên 55 kg được giảm 1500 đồng/1kg.

Tính số tiền một khách hàng phải trả cho m kg gạo mà mình mua ($m \geq 1$).

Bài toán 4: Tam giác

Cho tọa độ 3 đỉnh A(ax, ay), B(bx, by) và C(cx, cy) của tam giác ABC. Tính và hiển thị ra màn hình độ dài các cạnh AB, BC, AC của tam giác. Cho biết ba điểm A, B, C có thẳng hàng hay không, nếu không hãy tính và in ra màn hình chu vi (cv), diện tích (S) và bán kính đường tròn ngoại tiếp ($R = \frac{abc}{4S}$ với a, b, c là độ dài 3 cạnh) của tam giác tạo bởi 3 điểm đó.

Viết chương trình cho phép người sử dụng lựa chọn để thực hiện 1 trong 4 bài toán trên.

- Chọn 1: Thực hiện bài toán 1.
- Chọn 2: Thực hiện bài toán 2.
- Chọn 3: Thực hiện bài toán 3.
- Chọn 4: Thực hiện bài toán 4.
- Chọn 0: Thoát khỏi chương trình.

Nếu người sử dụng không chọn 1 trong 5 lựa chọn trên thì thông báo chọn sai và thoát chương trình.

Design by Minh An

Bài tập



Bài tập 3: Cho các bài toán

Bài toán 1: Tính $n!$, n là số nguyên không âm.

Bài toán 2: Tìm số đảo ngược

Cho n là số nguyên dương, số m là số đảo ngược của số n nếu các chữ số của m theo thứ tự ngược lại các chữ số của n . Ví dụ: $n = 23456$, thì $m = 65432$.

Bài toán 3: Tính gần đúng e^x với sai số epsilon theo công thức:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^i}{i!} + \dots$$

Bài toán 4: Tìm phân số tối giản của một phân số nếu nó chưa tối giản.

Viết chương trình cho phép người sử dụng lựa chọn để thực hiện 1 trong 4 bài toán trên.

- Chọn 1: Thực hiện bài toán 1.
- Chọn 2: Thực hiện bài toán 2.
- Chọn 3: Thực hiện bài toán 3.
- Chọn 4: Thực hiện bài toán 4.
- Chọn 0: Thoát khỏi chương trình.

Nếu người sử dụng không chọn 1 trong 5 lựa chọn trên thì thông báo chọn sai và thoát chương trình.

Chương trình cũng cho phép người sử dụng lặp lại lựa chọn để thực hiện bài toán mong muốn.

Design by Minh An

Bài tập



Bài tập 4: Cho các bài toán

Bài toán 1: Kiểm tra số nguyên tố

Bài toán 2: In ra màn hình tổng của n số nguyên tố đầu tiên

Bài toán 3: In ra màn hình các bảng cửu chương từ bảng 2 đến bảng 10

Bài toán 4: Tính gần đúng $\sin(x)$ với sai số epsilon theo công thức:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^i \frac{x^{2i+1}}{(2i+1)!} + \dots$$

Bài toán 5: In ra màn hình dãy n số fibonacci dạng: 1, 1, 2, 3, 5, 8, 13, ...

$$f_1 = f_2 = 1$$

$$f_n = f_{n-1} + f_{n-2}$$

Viết chương trình cho phép người sử dụng lựa chọn để thực hiện 1 trong 5 bài toán trên.

- Chọn 1: Thực hiện bài toán 1.
- Chọn 2: Thực hiện bài toán 2.
- Chọn 3: Thực hiện bài toán 3.
- Chọn 4: Thực hiện bài toán 4.
- Chọn 5: Thực hiện bài toán 5.
- Chọn 0: Thoát khỏi chương trình.

Nếu người sử dụng không chọn 1 trong 6 lựa chọn trên thì thông báo chọn sai.

Chương trình cũng cho phép người sử dụng lặp lại lựa chọn để thực hiện bài toán mong muốn.