Java Fundamental

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.

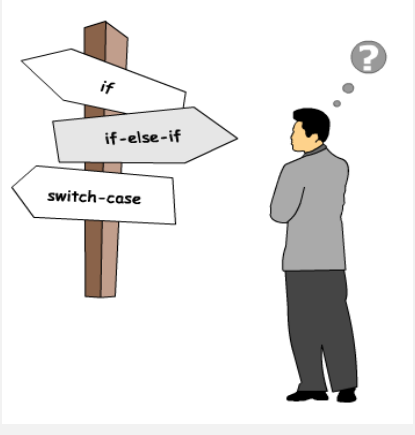
Design by Minh An

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.

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*

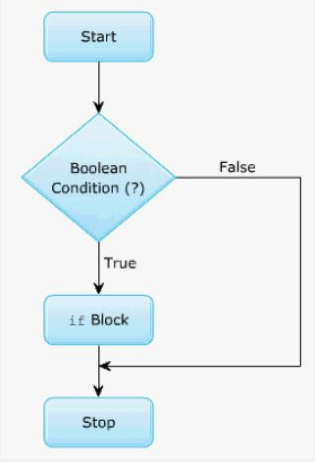


Cấu trúc rẽ nhánh if

- 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.



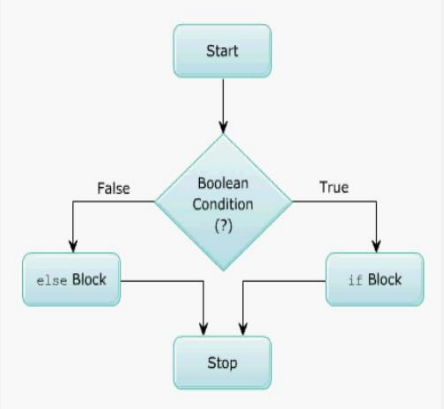
Design by Minh An

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

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

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 valuel:
    //statement sequence
    break;
case value2:
    //statement sequence
    break;
.....
.....
case valueN:
    //statement sequence
    break;
default:
   //default statement sequence
}
```

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);
```


Cấu trúc lặp

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

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 (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 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 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 (condition);
```

- Ví dụ: Kiểm tra tính đúng đắn của dữ liệu nhập vào.

Cấu trúc lặp for

- Câu lệnh *for* giống như câu *while* ở chức năng của nó.
- Khi số lần lặp được biết trước, câu lệnh for được sử dụng.

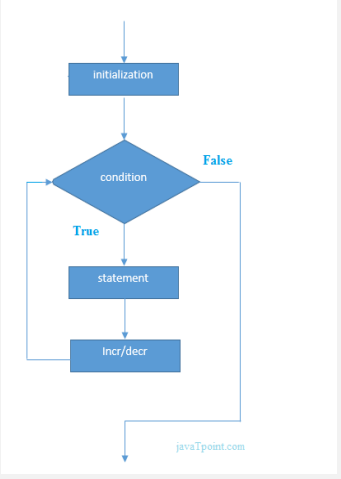
- Cú pháp:

```
for (exp1; exp2; exp3){
     statetments;
}
```

- Trong đó:
  - exp1 là biểu thức khởi tạo giá trị của biến điều khiển vòng lăp.
  - 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.

Cấu trúc lặp for

- Lệnh khởi tạo sẽ được chạy trước tiên và chỉ 1 lần duy nhất
- Sau đó biểu thức điều kiện 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, chạy lệnh tăng giảm
- Sau đó lại ktra đk và chạy tiếp thân vòng lặp nếu dk đúng. Thoát nếu dk sai



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("Tong cua 5 so le dau tien la : " + sum);
    }
}
```

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*.

Các lệnh chuyển điều khiển

- break;
- continue;
- return;