



Bài 2

Vòng lặp trong Java

Module: ADVANCED PROGRAMMING WITH JAVA

- Sử dụng được cú pháp Java để thao tác với cấu trúc lặp
- Sử dụng được vòng lặp for
- Sử dụng được vòng lặp while
- Sử dụng được vòng lặp do-while
- Sử dụng được break, continue

Thảo luận

Cấu trúc lặp for

Cấu trúc lặp while

Cấu trúc lặp do..while

Cấu trúc lặp for-each

Lệnh break

Lệnh continue

Vòng lặp for



- Cú pháp:

```
for (initial-action; loop-continuation-condition; action-after-each-iteration) {  
    statement(s);  
}
```

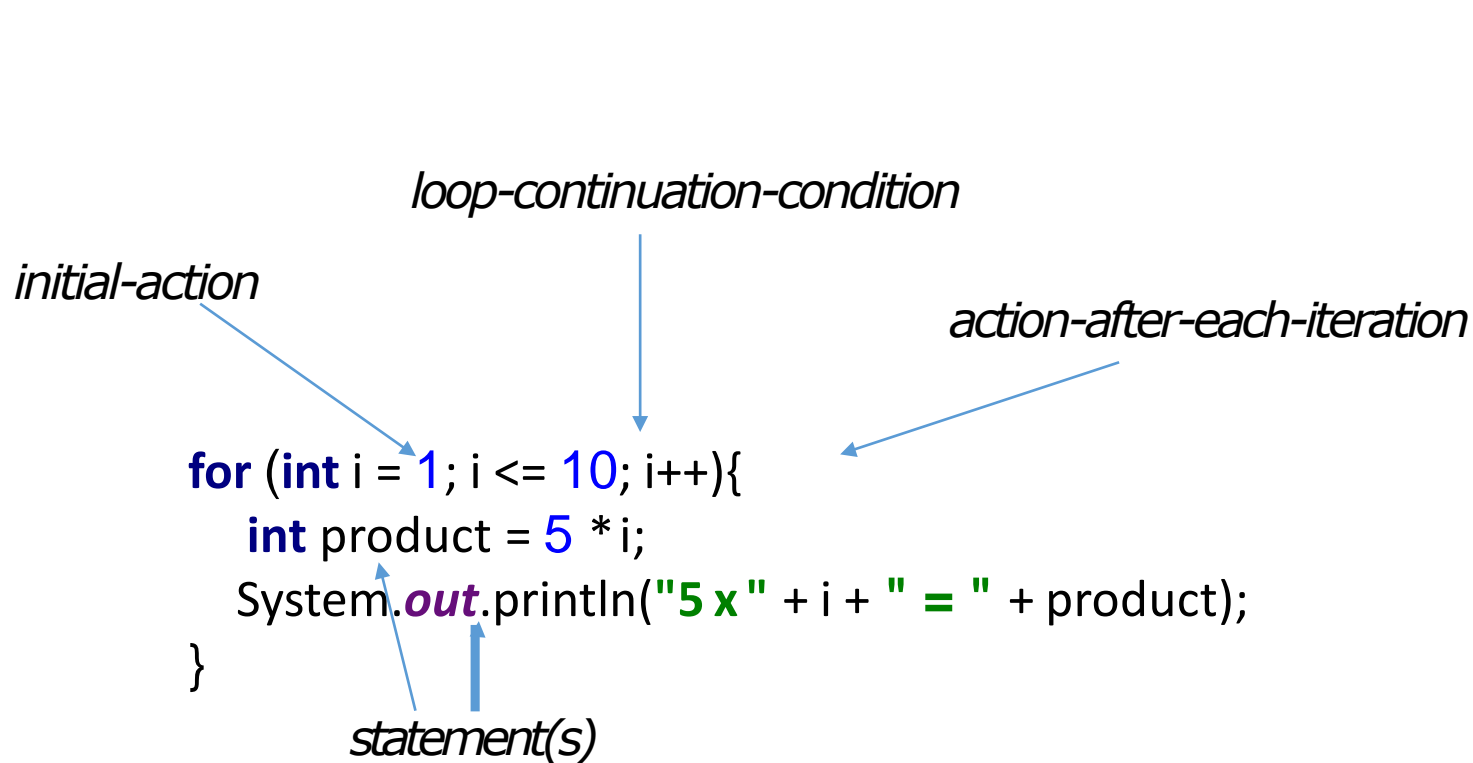
Trong đó:

- *initial-action*: là các câu lệnh được thực thi một lần duy nhất khi vòng lặp bắt đầu chạy
- *loop-continuation-condition*: là biểu thức điều kiện để xác định xem vòng lặp có được tiếp tục hay không
- *statement(s)*: là khối lệnh sẽ được thực thi trong mỗi lần lặp
- *action-after-each-iteration*: là các câu lệnh được thực thi sau mỗi lần lặp

Vòng lặp for: Ví dụ



- Hiển thị bảng cửu chương của 5



$$5 \times 1 = 5$$

$$5 \times 2 = 10$$

$$5 \times 3 = 15$$

$$5 \times 4 = 20$$

$$5 \times 5 = 25$$

$$5 \times 6 = 30$$

$$5 \times 7 = 35$$

$$5 \times 8 = 40$$

$$5 \times 9 = 45$$

$$5 \times 10 = 50$$

Lưu ý: Biến `i` ở vòng lặp trên còn được gọi là biến điều khiển (control variable). Phạm vi của biến `i` là bên trong vòng lặp `for`.

Vòng lặp for-each



- Vòng lặp for-each (còn gọi là *enhanced for*) được sử dụng để duyệt qua các phần tử của một collection, chẳng hạn như mảng, ArrayList, LinkedList, HashSet...
- Cú pháp:

```
for (type var: collection){  
  
}
```

Trong đó:

- type: Kiểu dữ liệu của các đối tượng của collection
- var: Biến đại diện lần lượt cho từng phần tử của collection trong mỗi lần lặp
- collection: đối tượng cần lặp

for-each: Ví dụ



- Duyệt qua các phần tử của một mảng:

```
int[] array = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
```

```
for (int element: array) {  
    System.out.println("element = " + element);  
}
```

- Tương đương với câu lệnh for:

```
for(int i = 0; i < array.length; i++){  
    System.out.println("element = " + array[i]);  
}
```

Vòng lặp *while*



- Vòng lặp *while* thực thi lặp lại một khối lệnh nếu biểu thức điều kiện trả về giá trị đúng
- Cú pháp:

```
while (loop-continuation-condition) {  
    statement(s);  
}
```

Trong đó:

- *loop-continuation-condition* :là biểu thức điều kiện
- *statement(s)*: là các câu lệnh được thực thi trong mỗi lần lặp

Vòng lặp while: Ví dụ



- Tìm *đáp án của phép cộng*:

```
int number1 = input.nextInt();  
int number2 = input.nextInt();  
int expectedAnswer = input.nextInt();  
while (number1 + number2 != expectedAnswer) {  
    System.out.print("Wrong answer");  
    expectedAnswer = input.nextInt();  
}  
System.out.println("You got it!");
```

Vòng lặp do-while



- Cú pháp:

```
do {  
    statements(s);  
} while (loop-continuation-condition);
```

Trong đó:

- statement(s): Các câu lệnh được thực thi trong mỗi lần lặp
- loop-continuation-condition: Biểu thức điều kiện. Nếu biểu thức điều kiện trả về giá trị true thì vòng lặp sẽ tiếp tục thực thi. Nếu biểu thức điều kiện trả về false thì vòng lặp kết thúc

do-while: Ví dụ



- Tính tổng các số nguyên:
 - Tính tổng của các số được nhập vào từ bàn phím
 - Nếu nhập vào số 0 thì kết thúc chương trình

```
int sum = 0;
int number;
Scanner scanner = new Scanner(System.in);
do{
    System.out.print("Enter a number: ");
    number = scanner.nextInt();
    sum += number;
} while (number != 0);
System.out.println("The sum is: " + sum);
```

break



- Câu lệnh `break` được sử dụng để kết thúc một vòng lặp
- Ví dụ:

```
for (int i = 0; i < 10; i++){  
    if(i == 5)  
        break;  
    System.out.println("i = " + i);  
}  
System.out.println("End of loop");
```

i = 0
i = 1
i = 2
i = 3
i = 4
End of loop

continue



- Câu lệnh `continue` được sử dụng để bỏ qua vòng lặp hiện tại
- Ví dụ:

```
for (int i = 0; i < 10; i++){  
    if(i == 5)  
        continue;  
    System.out.println("i = " + i);  
}  
System.out.println("End of loop");
```

i = 0
i = 1
i = 2
i = 3
i = 4
i = 6
i = 7
i = 8
i = 9
End of loop

Demo

for
while
do...while
for-each

Hướng dẫn

- Hướng dẫn làm bài thực hành và bài tập
- Chuẩn bị bài tiếp: Mạng và phương thức