



반복문 (for, while, do-while)

자바의 반복문 - for, while, do-while 알아보기

이번 시간에는 자바의 반복문에 대해 알아보겠습니다. 반복문은 특정 코드를 여러 번 실행할 때 사용하는 중요한 프로그래밍 구조입니다. 자바에서는 대표적으로 `for`, `while`, `do-while` 반복문이 있습니다. 이 세 가지는 모두 반복의 기능을 제공하지만, 사용되는 상황과 구조가 조금씩 다릅니다. 이번 글에서는 이 세 가지 반복문의 특징과 사용 방법을 자세히 알아보겠습니다.

1. 반복문이란?

반복문은 특정 작업을 여러 번 반복할 때 사용되는 코드 구조입니다. 즉, 동일한 코드를 여러 번 실행해야 할 때 유용하게 사용됩니다. 자바에서는 `for`, `while`, `do-while` 이 세 가지의 반복문을 제공합니다.

2. for 반복문

특징

- 반복 횟수가 명확하게 정해져 있을 때 사용합니다.
- 초기식, 조건식, 증감식으로 구성되어 있어 반복의 시작, 끝, 변화를 한눈에 볼 수 있습니다.

구조

```
for (초기식; 조건식; 증감식) {  
    // 반복할 코드  
}
```

- 초기식: 반복을 시작할 때 한 번만 실행되는 부분입니다.
- 조건식: 반복을 계속할지 결정하는 조건입니다. `true` 일 때 반복이 실행됩니다.
- 증감식: 각 반복이 끝날 때마다 실행되며, 반복 변수의 값을 변경합니다.

예제: 1부터 5까지 출력하기

```
for (int i = 1; i <= 5; i++) {  
    System.out.println("현재 숫자: " + i);  
}
```

- **설명:** `i` 는 1부터 시작하여 5보다 작거나 같은 동안 반복됩니다. 매번 `i++` 을 통해 `i` 가 1씩 증가합니다.

결과:

```
현재 숫자: 1  
현재 숫자: 2  
현재 숫자: 3  
현재 숫자: 4  
현재 숫자: 5
```

언제 사용하는가?

- 반복 횟수가 명확할 때, 예를 들어 "1부터 100까지 출력하라" 같은 경우.

3. while 반복문

특징

- 반복 횟수가 명확하지 않을 때 사용합니다.
- 조건식이 `true` 인 동안 계속해서 반복합니다.

구조

```
while (조건식) {  
    // 반복할 코드  
}
```

- 조건식이 `true` 일 때 반복이 실행됩니다. 조건식이 `false` 가 되면 반복을 종료합니다.

예제: 사용자가 "exit"를 입력할 때까지 반복하기

```
import java.util.Scanner;

Scanner scanner = new Scanner(System.in);
String input = "";

while (!input.equals("exit")) {
    System.out.print("문자를 입력하세요 ('exit' 입력 시 종료):");
    input = scanner.nextLine();
    System.out.println("입력한 내용: " + input);
}
System.out.println("프로그램 종료");
```

- **설명:** 사용자가 "exit"을 입력할 때까지 반복합니다.

언제 사용하는가?

- 반복 횟수가 정해져 있지 않거나 언제 끝날지 모를 때, 예를 들어 사용자가 원하는 만큼 입력을 받을 때.

4. do-while 반복문

특징

- 최소 한 번은 반드시 실행되어야 할 때 사용합니다.
- **do** 블록을 먼저 실행한 후 조건을 검사합니다.

구조

```
do {
    // 반복할 코드
} while (조건식);
```

- **do** 블록의 코드를 먼저 실행하고, 그 후에 조건식을 검사합니다. 조건식이 **true** 이면 다시 반복을 실행합니다.

예제: 사용자가 1부터 10 사이의 숫자를 입력할 때까지 반복

```
import java.util.Scanner;

Scanner scanner = new Scanner(System.in);
int number;

do {
    System.out.print("1부터 10 사이의 숫자를 입력하세요: ");
    number = scanner.nextInt();
} while (number < 1 || number > 10);

System.out.println("올바른 숫자를 입력하셨습니다: " + number);
```

- **설명:** `do` 블록이 먼저 실행되므로, 사용자가 최소 한 번은 숫자를 입력하게 됩니다.

언제 사용하는가?

- **최소 한 번은 실행되어야 하는 작업**이 있을 때, 예를 들어 네트워크 통신에서 서버에 연결을 시도할 때와 같이.

정리

- **for:** 반복 횟수가 명확할 때 사용
- **while:** 반복 횟수가 불확실하거나 조건이 언제 `false` 가 될지 모를 때 사용
- **do-while:** 최소 한 번은 실행되어야 할 때 사용

반복문은 프로그램에서 매우 중요한 역할을 합니다. 세 가지 반복문을 잘 이해하고 상황에 맞게 사용하면 효율적인 코드를 작성할 수 있습니다.