



반복문과 배열 그리고 예외 처리

반복문의 특징

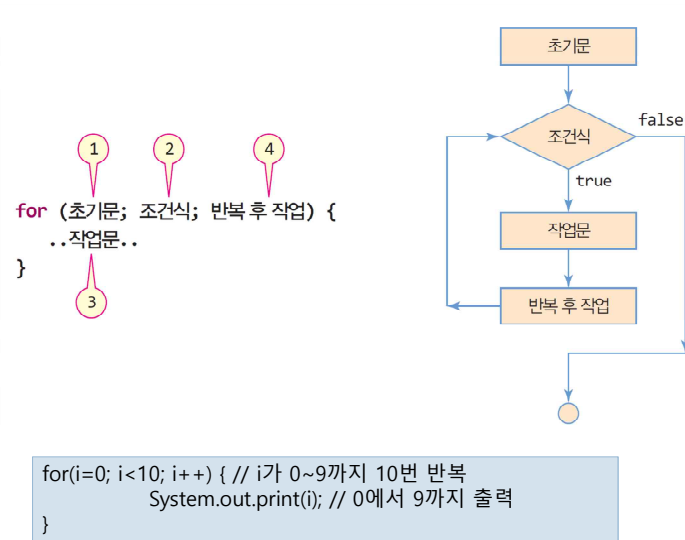
2

- 자바 반복문의 종류
 - ▣ for 문
 - ▣ while 문
 - ▣ do while 문



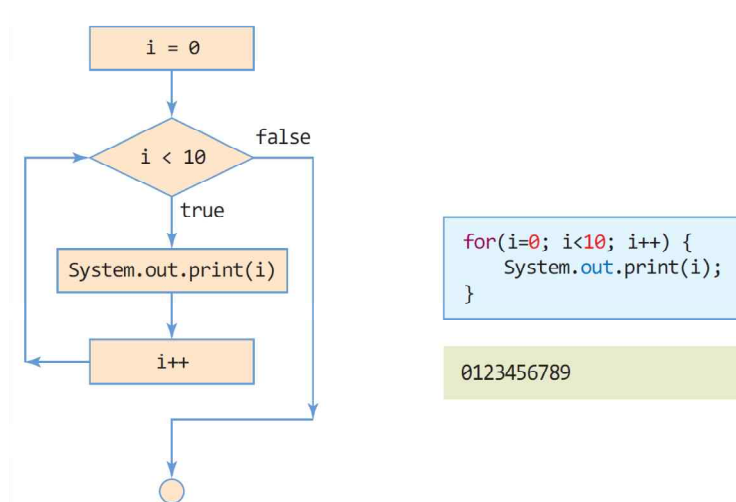
for 문의 구성

3



for 문의 실행 과정을 나타내는 순서도

4



for문의 예시

5

- 0에서 9까지 정수 출력

```
int i;
for(i = 0; i < 10; i++) {
    System.out.print(i);
}
```

```
int i;
for(i = 0; i < 10; i++)
    System.out.print(i);
```

- for문에서 변수 선언 가능

```
for(int i = 0; i < 10; i++) // 변수 i는 for문을 벗어나서 사용할 수 없음
    System.out.print(i);
```

- 0에서 100까지의 합 구하기

```
int sum = 0;
for(int i = 0; i <= 100; i++)
    sum += i;
```

```
int i, sum;
for(i = 0, sum=0; i <= 100; i++)
    sum += i;
```

```
int sum = 0;
for(int i = 100; i >= 0; i--)
    sum += i;
```

for문의 특이한 형태

6

```
for(초기작업; true; 반복후작업) { // 반복 조건이 true이면 무한 반복
    .....
}
```

```
for(초기작업; ; 반복후작업) { // 반복조건이 비어 있으면 true로 간주, 무한 반복
    .....
}
```

```
// 초기 작업과 반복후작업은 ';'로 분리하여 여러 문장 나열 가능
for(i=0; i<10; i++, System.out.println(i)) {
    .....
}
```

```
// for문 내에 변수 선언
for(int i=0; i<10; i++) { // 변수 i는 for문 내에서만 사용 가능
    .....
}
```

예제 3-1 : for 문을 이용하여 1부터 10까지 합 출력

7

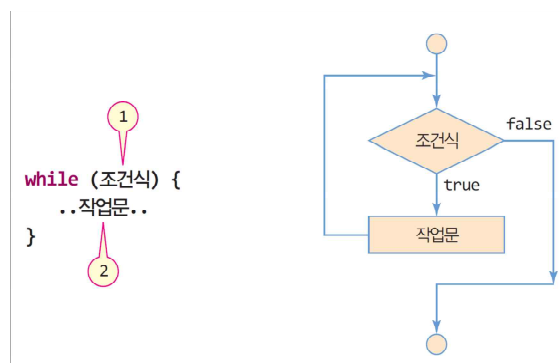
for문을 이용하여 1부터 10까지 덧셈을 표시하고 합을 구하시오.

```
public class ForSample {  
    public static void main(String[] args) {  
        int sum=0;  
  
        for(int i=1; i<=10; i++) { // 1~10까지 반복  
            sum += i;  
            System.out.print(i); // 더하는 수 출력  
  
            if(i<=9) // 1~9까지는 '+' 출력  
                System.out.print("+");  
            else { // i가 10인 경우  
                System.out.print("="); // '=' 출력하고  
                System.out.print(sum); // 덧셈 결과 출력  
            }  
        }  
    }  
}
```

1+2+3+4+5+6+7+8+9+10=55

while 문의 구성

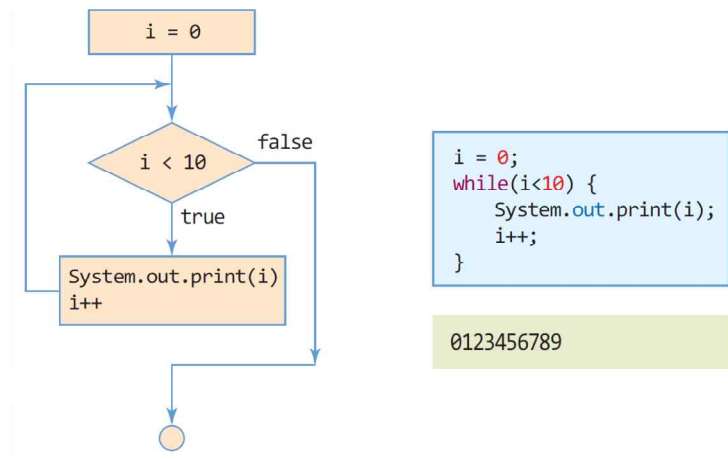
8



- 반복 조건이 true이면 반복, false이면 반복 종료
- 반복 조건이 없으면 컴파일 오류
- 처음부터 반복조건을 통과한 후 작업문 수행

while문의 실행 과정을 나타내는 순서도

9



예제 3-2 : -1이 입력될 때까지 입력된 수의 평균 구하기

10

while문을 이용하여 정수를 여러 개 입력 받고 평균을 출력하라. -1이 입력되면 입력을 종료한다.

```
import java.util.Scanner;
public class WhileSample {
    public static void main(String[] args) {
        int count=0; // count는 입력된 정수의 개수
        int sum=0; // sum은 합
        Scanner scanner = new Scanner(System.in);
        System.out.println("정수를 입력하고 마지막에 -1을 입력하세요.");

        int n = scanner.nextInt(); // 정수 입력
        while(n != -1) { // -1이 입력되면 while 문 벗어남
            sum += n;
            count++;
            n = scanner.nextInt(); // 정수 입력
        }
        if(count == 0) System.out.println("입력된 수가 없습니다.");
        else {
            System.out.print("정수의 개수는 " + count + "개이며 ");
            System.out.println("평균은 " + (double)sum/count + "입니다.");
        }
        scanner.close();
    }
}
```

정수를 입력하고 마지막에 -1을 입력하세요.

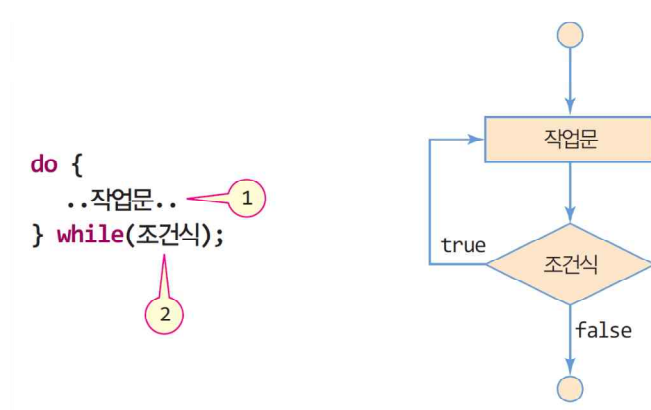
10 30 -20 40 -1

정수의 개수는 4개이며 평균은 15.0입니다.

-1은 마지막 입력을 뜻함

do-while 문의 구성

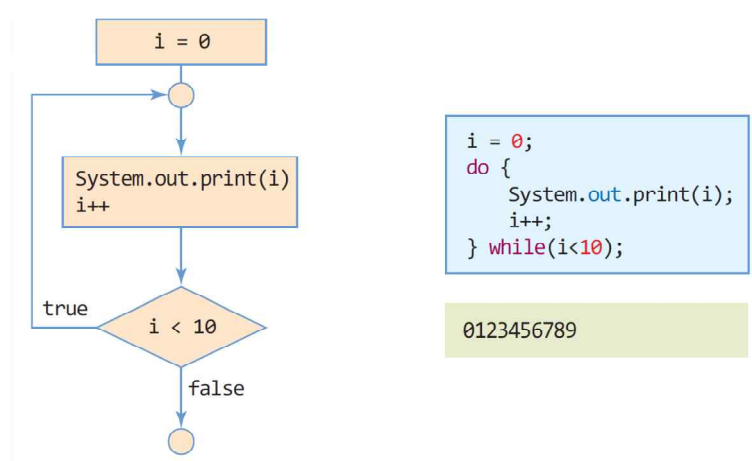
11



- 무조건 최소 한번 작업문 실행
- 반복 조건이 true이면 반복, false이면 반복 종료
- 반복 조건이 없으면 컴파일 오류

do-while문의 실행 과정을 나타내는 순서도

12



예제 3-3 : a-z까지 출력

13

do-while 문을 이용하여 'a'부터 'z'까지 출력하는 프로그램을 작성하시오.

```
public class DoWhileSample {  
    public static void main (String[] args) {  
        char c = 'a';  
  
        do {  
            System.out.print(c);  
            c = (char) (c + 1);  
        } while (c <= 'z');  
    }  
}
```

abcdefghijklmnopqrstuvwxyz

중첩 반복

14

- 중첩 반복
 - ▣ 반복문이 다른 반복문을 내포하는 구조
 - ▣ 이론적으로는 몇 번이고 중첩 반복 가능
 - ▣ 너무 많은 중첩 반복은 프로그램 구조를 복잡하게 하므로 2중 또는 3중 반복이 적당

```
for(int i=0; i<100; i++) { // 100개의 학교 성적을 모두 더한다.
```

```
.....
```

```
    for(int j=0; j<10000; j++) { // 10000명의 학생 성적을 모두 더한다.
```

```
        .....
```

```
        .....
```

```
    }  
    .....
```

```
}
```

10000명의 학생이 있는 100개 대학의 모든 학생 성적의 합을 구할 때,
for 문을 이용한 이중 중첩 구조

예제 3-4 : 2중 중첩을 이용한 구구단

15

2중 중첩 for문을 사용하여 구구단을 출력하는 프로그램을 작성하시오. 한 줄에 한 단씩 출력한다.

```
public class NestedLoop {
    public static void main(String[] args) {
        for(int i=1; i<10; i++) { // 1단에서 9단
            for(int j=1; j<10; j++) { // 각 단의 구구셈 출력
                System.out.print(i + "*" + j + "=" + i*j); // 구구셈 출력
                System.out.print("\t"); // 하나씩 탭으로 띄기
            }
            System.out.println(); // 한 단이 끝나면 다음 줄로 커서 이동
        }
    }
}
```

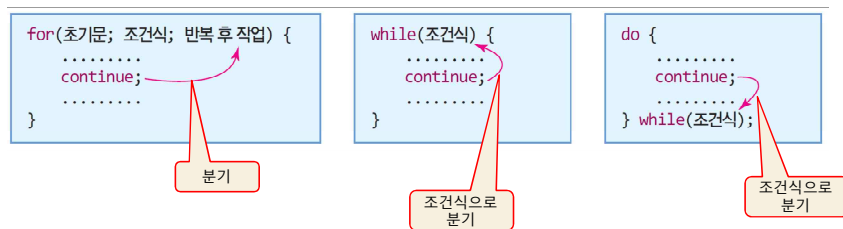
1*1=1	1*2=2	1*3=3	1*4=4	1*5=5	1*6=6	1*7=7	1*8=8	1*9=9
2*1=2	2*2=4	2*3=6	2*4=8	2*5=10	2*6=12	2*7=14	2*8=16	2*9=18
3*1=3	3*2=6	3*3=9	3*4=12	3*5=15	3*6=18	3*7=21	3*8=24	3*9=27
4*1=4	4*2=8	4*3=12	4*4=16	4*5=20	4*6=24	4*7=28	4*8=32	4*9=36
5*1=5	5*2=10	5*3=15	5*4=20	5*5=25	5*6=30	5*7=35	5*8=40	5*9=45
6*1=6	6*2=12	6*3=18	6*4=24	6*5=30	6*6=36	6*7=42	6*8=48	6*9=54
7*1=7	7*2=14	7*3=21	7*4=28	7*5=35	7*6=42	7*7=49	7*8=56	7*9=63
8*1=8	8*2=16	8*3=24	8*4=32	8*5=40	8*6=48	8*7=56	8*8=64	8*9=72
9*1=9	9*2=18	9*3=27	9*4=36	9*5=45	9*6=54	9*7=63	9*8=72	9*9=81

continue문

16

□ continue 문

- 반복문을 빠져 나가지 않으면서 다음 반복으로 진행



예제 3-5 : continue 문을 이용하여 양수 합 구하기

17

5개의 정수를 입력 받고 그 중 양수들만 합하여 출력하는 프로그램을 작성하라.

```
import java.util.Scanner;

public class ContinueExample {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("정수를 5개 입력하세요.");
        int sum=0;
        for(int i=0; i<5; i++) {
            int n = scanner.nextInt(); // 키보드에서 정수 입력
            if(n<=0)
                continue; // 양수가 아닌 경우 다음 반복으로 진행
            else
                sum += n; // 양수인 경우 덧셈
        }
        System.out.println("양수의 합은 " + sum);

        scanner.close();
    }
}
```

정수를 5개 입력하세요.

5

-2

6

10

-4

양수의 합은 21

break문

18

□ break 문

□ 반복문 하나를 완전히 빠져 나갈 때 사용

- 하나의 반복문만 벗어남
- 중첩 반복의 경우 안쪽 반복문의 break 문이 실행되면 안쪽 반복문만 벗어남

```
for(초기문; 조건식; 반복 후 작업) {
    .....
    break;
    .....
}
```

(a) 현재 반복문 벗어나기

```
for(초기문; 조건식; 반복 후 작업) {
    while(조건식) {
        .....
        break;
        .....
    }
    .....
}
```

(b) 중첩 반복에서 안쪽 반복문만 벗어나는 경우

예제 3-6 : break 문을 이용하여 while 문 벗어나기

19

"exit"이 입력되면 while 문을 벗어나도록 break 문을 활용하는 프로그램을 작성하라.

```
import java.util.Scanner;

public class BreakExample {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("exit을 입력하면 종료합니다.");
        while(true) {
            System.out.print("> >");
            String text = scanner.nextLine();
            if(text.equals("exit")) // "exit"이 입력되면 반복 종료
                break; // while 문을 벗어남
        }
        System.out.println("종료합니다...");
        scanner.close();
    }
}
```

문자열 비교 시 equals() 사용

```
exit을 입력하면 종료합니다.
>>edit
>>exit
종료합니다...
```