

자바1

(반복문)

Chapter 05

반복문

Chapter05의 학습목표

- 조건문/ 반복문에 대해 알아본다.
- while문, for문의 특징과 차이점에 대해 이해하고 구현할 수 있다.
- 자바에서 지원하는 제어종류에 대해 학습한다.

반복문

반복문

- 어떤 작업을 계속해서 수행해야 할 경우에 사용

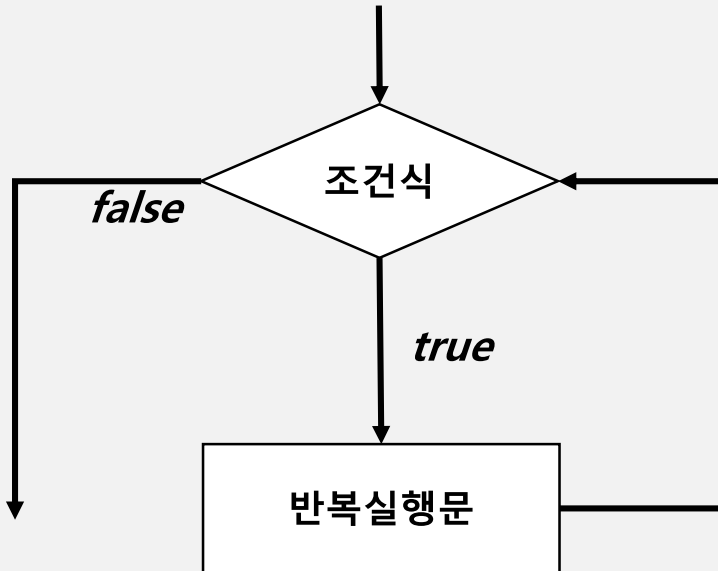
반복문	특징	종료시점	예
for	원하는 반복 횟수 설정	반복횟수가 끝날 때 까지	문자'A'를 10번 출력
while	반복 종료 조건 을 설정	반복 조건이 만족하지 않을 때까지	'A'를 입력 받을 때까지 문자열 'next'를 출력

반복문

1. while문

- 특정 조건을 만족하는 동안 반복해서 명령을 실행
- 조건식이 참이면 반복, 거짓이면 while문 종료

```
while (조건식){  
    반복실행문(명령)}
```



while문 예제)

```
int n = 1;  
while(n <= 10){  
    System.out.println(n);  
    n += 1;  
}
```

```
int n = 10;  
while(n >= 1){  
    System.out.println(n);  
    n -= 1;  
}
```

무한루프에 들어가지 않도록 조심!

반복문

1-1. while문 응용 예제1

문자 'a'를 입력할 때 까지 입력 받은 문자를 출력하는 프로그램을 작성

```
문자를 입력하세요 >>> b
문자가 틀렸습니다.
문자를 입력하세요 >>> c
문자가 틀렸습니다.
문자를 입력하세요 >>> e
문자가 틀렸습니다.
문자를 입력하세요 >>> a
문자가 맞았습니다.
```

```
import java.util.*;

public class Test {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("문자를 입력하세요 >>> ");
        String s = input.next();

        while(!s.equals("a")) {
            System.out.println("문자가 틀렸습니다.");
            System.out.print("문자를 입력하세요 >>> ");
            s = input.next();
        }

        System.out.println("문자가 맞았습니다.");
    }
}
```

반복문

1-2. while문 응용 예제2

문자열 `code` = ""를 선언한 후 `code`에 문자를 'A'씩 추가시켜 문자열이 "AAAAA"가 되면 끝나는 반복문을 작성하시오.

```
code = A
code = AA
code = AAA
code = AAAA
code = AAAAA
```

```
public class Test {
    public static void main(String[] args) {
        String code = "";

        while(!code.equals("AAAAA")) {
            code += 'A';
            System.out.println("code = " + code);
        }
    }
}
```

반복문

2. do-while문

- 일단 while문 안의 명령을 실행한 뒤 조건식을 검사
- 참/거짓 여부에 관계없이 적어도 한번 수행

```
do{  
    반복실행문(명령)  
}  
while (조건식);
```

```
public class Test {  
    public static void main(String[] args) {  
        int i = 100;  
        do {  
            System.out.println("적어도 한번 실행");  
        } while(i < 10);  
    }  
}
```

3. while문 중첩 (다중 while문)

- while문 내부에 또 다른 while문이 나타나는 것
- 반복 처리해야하는 대상이 2개 이상일 때 사용

```
public class Test {  
    public static void main(String[] args) {  
        int i = 0;  
  
        while(i < 10) {  
            int j = 0;  
            i++;  
  
            while(j < 10) {  
                System.out.println("i = " + i);  
                System.out.println("j = " + j);  
                System.out.println();  
                j++;  
            }  
        }  
    }  
}
```

반복문

3-1. while문 중첩 예제

다중 while문을 사용한
타이머 제작

```
int sec = 0, min = 0, hour = 0;

while(hour < 1) {
    while(min < 60) {
        while(sec < 60) {
            System.out.println(hour+"시/"+min+"분"+sec+"초");
            sec += 1;
        }
        sec = 0;
        min += 1;
    }
    min = 0;
    hour += 1;
}

System.out.println(hour+"시/"+min+"분"+sec+"초");
```


반복문 실습

1. while문을 사용하여 오른쪽과 같이
구구단을 출력하는 프로그램을 작성하세요.
hint) 이스케이프 문자, 2중 while

2x1=2	3x1=3	4x1=4	5x1=5	6x1=6	7x1=7	8x1=8	9x1=9
2x2=4	3x2=6	4x2=8	5x2=10	6x2=12	7x2=14	8x2=16	9x2=18
2x3=6	3x3=9	4x3=12	5x3=15	6x3=18	7x3=21	8x3=24	9x3=27
2x4=8	3x4=12	4x4=16	5x4=20	6x4=24	7x4=28	8x4=32	9x4=36
2x5=10	3x5=15	4x5=20	5x5=25	6x5=30	7x5=35	8x5=40	9x5=45
2x6=12	3x6=18	4x6=24	5x6=30	6x6=36	7x6=42	8x6=48	9x6=54
2x7=14	3x7=21	4x7=28	5x7=35	6x7=42	7x7=49	8x7=56	9x7=63
2x8=16	3x8=24	4x8=32	5x8=40	6x8=48	7x8=56	8x8=64	9x8=72
2x9=18	3x9=27	4x9=36	5x9=45	6x9=54	7x9=63	8x9=72	9x9=81

2. 1부터 100사이의 모든 정수 중 3의 배수만
오른쪽 그림과 같이 출력하세요.
한 줄에 5개씩 출력하세요.
hint) 이스케이프 문자, while, if, 나머지연산

3	6	9	12	15
18	21	24	27	30
33	36	39	42	45
48	51	54	57	60
63	66	69	72	75
78	81	84	87	90
93	96	99		

반복문

for문

- 값의 범위나 횟수가 정해져 있을 때 사용
- 정해진 횟수만큼 실행

```
for (초기화; 조건식; 증감식){  
    조건식이 참일때의 반복실행문(명령)}
```

- 1) 초기화 : 반복문을 위해 사용할 변수 설정
- 2) 조건식 : 반복을 계속하기 위한 조건을 설정
(조건식의 값이 true이면 반복)
- 3) 증감식 : 변수의 값을 증가 또는 감소시킴

- 초기화와 증감식은 생략 가능
- 초기화와 증감식은 여러 개 작성 가능

1

```
//초기화, 조건식, 증감식 작성  
for(int i = 0; i < 10; i++){  
    System.out.print(i);  
}
```

0123456789

2

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

0123456789

반복문

3) for문을 사용한 구구단 출력

출력하고자 하는 구구단을 입력 받아 해당 구구단을 출력하는 프로그램

```
import java.util.Scanner;

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

        System.out.print("출력할 구구단을 입력하세요 >>> ");
        int dan = input.nextInt();

        for(int i = 1; i < 10; i++) {
            System.out.println(dan + " * " + i + " = " + dan * i);
        }
    }
}
```

```
출력할 구구단을 입력하세요 >>> 4
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
```

반복문

다중 for문

- 이중 반복

```
for (초기화; 조건식; 증감식){  
    for (초기화; 조건식; 증감식){  
        조건식이 참일때의 반복실행문(명령)  
    }  
}
```

다중 for문 예제)

```
for (int i = 0; i < 10; i++) {  
    for (int j = 0; j < 10; j++) {  
        System.out.printf("i = %d, j = %d", i , j);  
        System.out.println();  
    }  
}
```

```
i = 0, j = 0  
i = 0, j = 1  
...  
i = 0, j = 9  
i = 1, j = 0  
i = 1, j = 1  
...  
i = 9, j = 8  
i = 9, j = 9
```

제어문

1) break문

- while/for과 같은 반복문을 강제로 종료하고자 할 때 사용
- break문을 만나면 반복문을 그 시점에서 종료

```
for (초기화; 조건식; 증감식){  
    반복실행문(명령)  
    break;  
    반복실행문(명령)  
}
```

```
int n = 1;  
while (true){  
    System.out.println(n);  
    if (n == 10)  
        break;  
    n += 1;  
}
```

n이 10이 되는 순간 break문을 만나고 반복문이 종료.
loop를 빠져나감.

2) continue문

- 반복문의 시작 지점으로 제어를 옮김
- continue를 만나면 반복문의 이후 코드 실행x

```
for (초기화; 조건식; 증감식){  
    반복실행문(명령)  
    continue;  
    반복실행문(명령)  
}
```

```
int total = 0;  
for (int i = 0; i < 101; i++){  
    if (i % 3 == 0)  
        continue;  
    total += i;  
}  
System.out.println(total);
```

3의 배수를 만나면 total을 증가시키지 않고 i만 증가

반복문 실습

반복문 실습

1. n과 m 변수에 두 숫자를 입력받아 가로와 세로의 길이가 n, m인 직사각형 모양의 형태를 오른쪽 그림과 같이 출력해보세요.

3
5

2. 정수 number를 입력 받아 해당 정수의 팩토리얼을 구하는 프로그램을 작성하세요.
TMI) factorial : 1부터 어떤 양의 정수 n까지의 정수를 모두 곱한 것
예) number가 5일때, 5! (5팩토리얼) => 1*2*3*4*5 => 120

정수 입력 >>> 5
5! = 120입니다.

3. 정수 n을 입력 받아 n의 약수를 모두 더한 값을 출력하는 프로그램을 작성하세요.
TMI) 약수 : 해당 수를 나눴을 때 딱 떨어지는 수

정수 입력 >>> 5
120의 약수의 합은 360입니다.

수고하셨습니다.