

2부 자바 기본 다루기

-8장 제어문을 보조하는 보조 제어문

최문환



08장 제어문을 보조하는 보조 제어문

1. 실행을 중지하는 break 보조 제어문
2. continue 보조 제어문
3. 실행이 종료되지 않고 끝없이 수행되는
무한 루프

1. break 보조 제어문

break문을 만나면 반복문 밖으로 제어가 이동

반복문 {

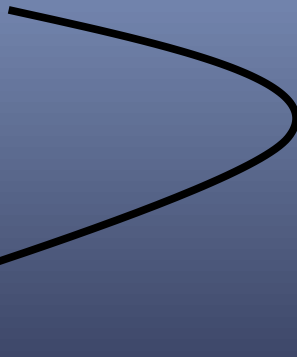
:

break;

:

:

}




<예제> break 제어문 사용 예

```
01: class F01 {
02:     public static void main(String[] args) {
03:         int n;
04:         for(n=1; n<=10; n++)    //1부터 10까지 자연수를 구함
05:             System.out.print(" " + n);
06:
07:
08:         System.out.println("");
09:
10:         for(n=1; n<=10; n++){ //1부터 10까지 자연수를 구함
11:             if(n%3==0)        //제어변수 n이 3으로 나누어 떨어지면
12:                 break;        //반복문을 벗어남
13:             System.out.print(" " + n);
14:         }
15:         System.out.println("");
16:     } //main 함수의
17: }
```

1. break 보조 제어문

다중 반복문에서 break문이 사용되었을 경우에는 가장 근접한 반복문에서만 벗어남

```
바깥쪽 반복문{  
    안쪽 반복문 {  
        break;  
        :  
    }  
    :  
}
```

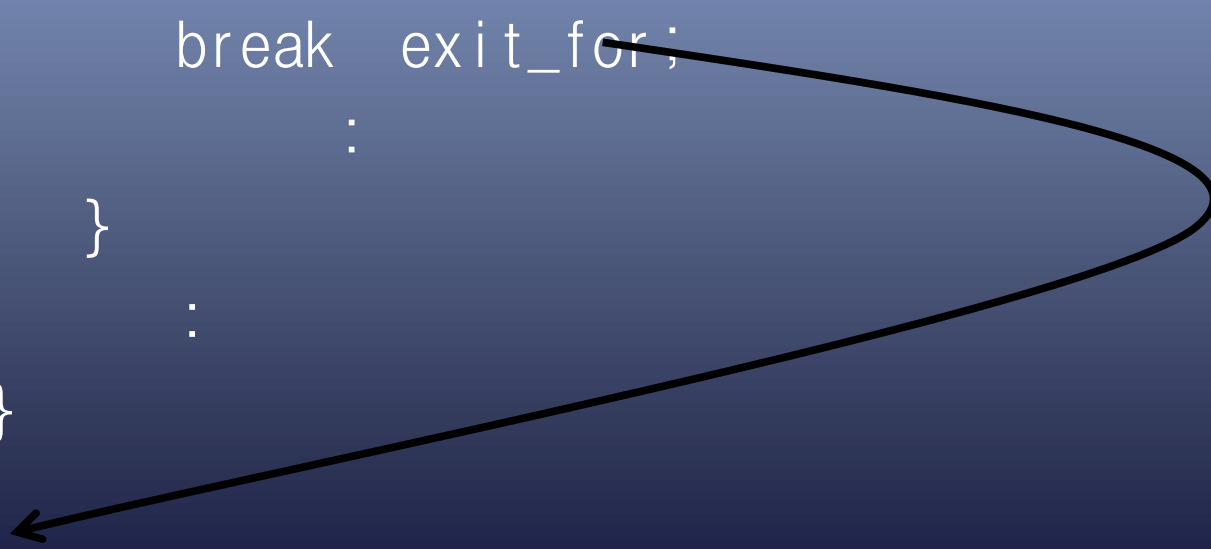


1. break 보조 제어문

만일 바깥쪽 반복문 밖으로 벗어나려면 어떻게 해야 할까?
레이블을 사용해야 한다.

break 다음에 레이블명을 기술하여 완전히 반복문에서 벗어날 수 있다.

```
exit_for:
바깥쪽 반복문{
    안쪽 반복문 {
        break exit_for;
    }
}
```



<예제> break 제어문 사용 예

```
01: class F10{
02:     public static void main(String[] args)
03:     {
04:         int i;
05:         int a;
07:
08:         for(a=1; a<10; a++){
09:             for(i=1; i<=10; i++){
10:                 if(i%3==0)
11:                     break;
12:                 System.out.print(" i->" + i );
13:             }
14:             System.out.println("\n a -> " + a);
15:         }
16:         System.out.println("-----\n");
```

<예제> break 제어문 사용 예

```
18: exit_for:
19:   for(a=1; a<10; a++){
20:     for(i=1; i<=10; i++){
21:       if(i%3==0)
22:         break exit_for;
23:       System.out.print(" i->" + i );
24:     }
25:     System.out.println("\n a -> " + a);
26:   }
27:   System.out.println("\n-----
");
28: }
29: }
```


2. continue 보조 제어문

continue문의 반복을 계속하기 위해서 증감식으로 제어가 옮겨진다.

```
반복문 {  
    :  
    :  
    continue;  
    :  
    :  
}
```



<예제> 1부터 10사이의 정수 중 3의 배수만 제외하고 출력하기

```
01: class F02 {  
02:     public static void main(String[] args) {  
03:         int n;  
04:         for(n=1; n<=10; n++){ //1부터 10까지 자연수를 구함  
05:             if(n%3==0) //제어변수 n이 3으로 나누어떨어지면  
07:                 continue; //증감식으로 제어가 옮겨져서 반복은 계속  
08:             System.out.print(" " + n);  
09:         }  
10:         System.out.println("");  
11:     } //main 함수의 끝  
12: }
```

3. 무한 루프

- 무한 루프는 프로그램의 실행이 종료되지 않고 끝없이 수행되어지고 있는 상태

3.1 while문을 이용한 무한 루프

- 조건식의 결과가 참이면 반복을 계속하고 거짓이면 반복문을 벗어난다.
- while 문을 이용한 무한 루프는 일반적으로 조건식에 true를 기술하여 표현

```
while( true ) {  
    :  
}
```

<예제> while문을 이용한 무한 루프

```
001: class F03 {  
002:     public static void main(String[] args) {  
003:         int n=0;  
004:         while( true ) {  
005:             System.out.print("Fall Wonderland ");  
007:             if(++n >=10)  
008:                 break;  
009:         }  
010:         System.out.println("The End ");  
011:     }  
012: }
```

<문제>

1. 1부터 100사이의 자연수중에서 제일 큰 7의 배수를 구하는 프로그램을 작성하시오. (Ex08_01.java-> for,if,break)

2. 다음 프로그램의 수행 결과값을 예측해 보시오.

```
class Ex08_02{  
    public static void main(String[] args) {  
        int i, s=0;  
        for(i=1; i<=5; ++i){  
            if(i==3) continue;  
            s+=i;  
        }  
        System.out.println(" sum = " + s);  
    }  
}
```