

컴퓨터 프로그래밍 (Computer Programming)

이 선 순



9. 반복문의 심화



목차

1. while문

2. do~while문

3. 기타 제어문

01

while문

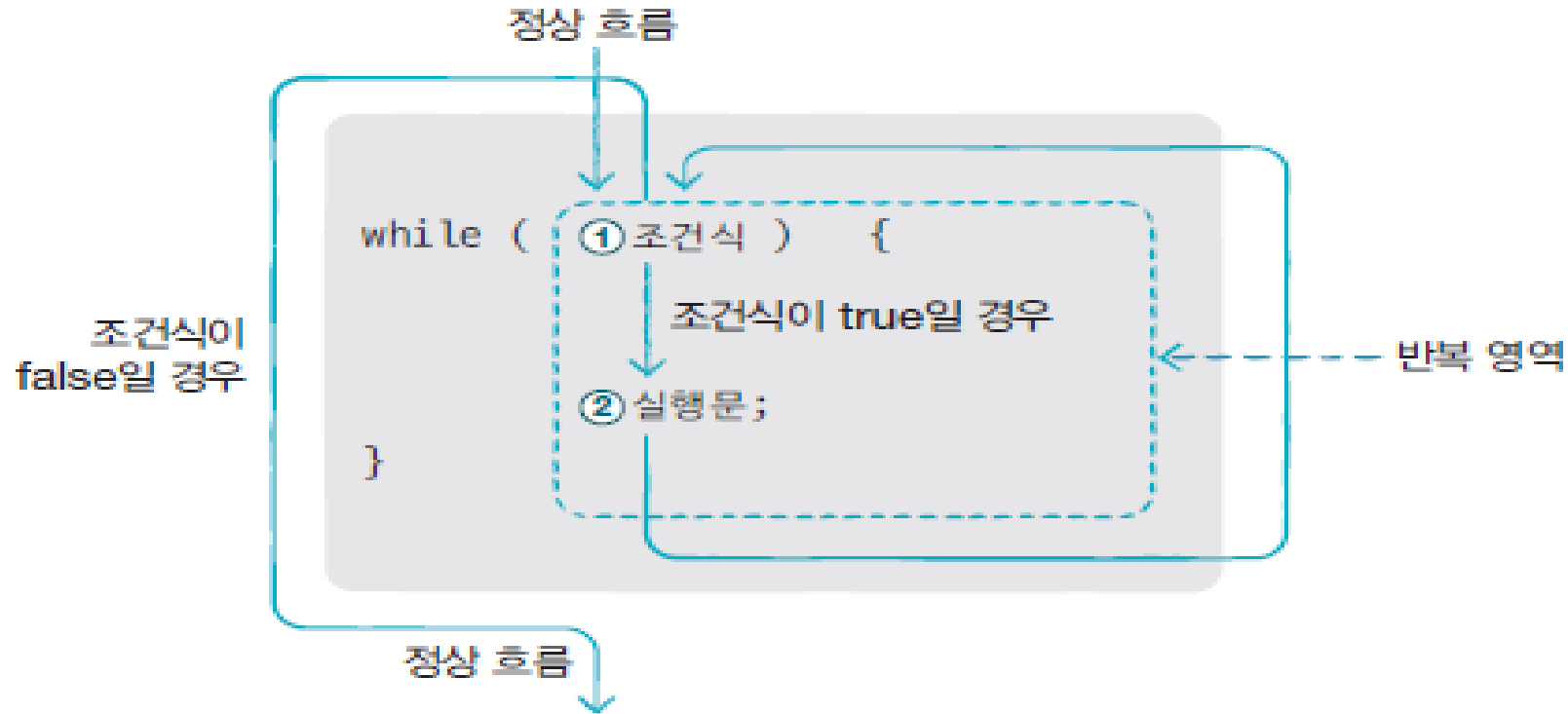
01 while문

■ while문

- while문은 for문 다음으로 활용도가 높은 반복문
- for문이 정해진 횟수만큼 반복한다면, while문은 조건식이 참인 동안 블록안의 내용을 반복해서 수행함
- 조건식에는 비교 또는 논리 연산식이 주로 오는데, 조건식에 따라 반복 여부를 결정함
 - 조건식이 true 일 경우 계속해서 반복
 - 조건식이 false 일 경우 반복을 멈추고 while문을 종료함

01 while문

■ while문



1. While문이 처음 실행될 때 조건식(①)을 평가함
2. 평가결과가 true이면 실행문(②)을 실행함
3. 실행문(②)이 모두 실행되면 조건식으로 되돌아가서 다시 조건식(①)을 평가함
4. 만약 조건식이 true이면 실행문(②)→ 조건식(①)으로 다시 진행함
5. 만약 조건식이 false라면 while문을 종료함

01 while문

■ While문과 for문의 비교

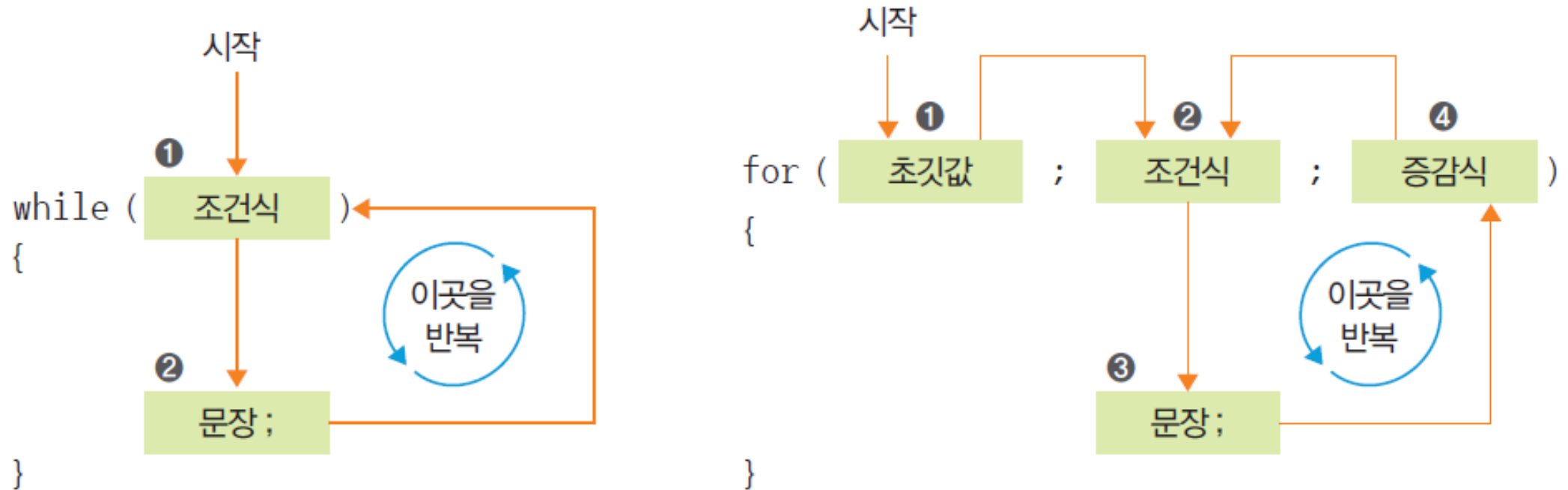


그림 7-1 while 문의 실행 순서와 for 문의 실행 순서 비교

01 while문

■ While문과 for문의 비교

① 원래의 for 문

```
int i;  
for (i=0; i<10 ; i++)  
{  
    출력 ("%d \n", i) ;  
}
```

==

② 초깃값과 증감식의 위치 이동

```
int i;  
i = 0;  
for ( ; i<10 ; )  
{  
    출력 ("%d \n", i) ;  
    i++ ;  
}
```

==

③ while 문으로 변환

```
int i;  
i = 0;  
while (i<10)  
{  
    출력 ("%d \n", i) ;  
    i++ ;  
}
```


01 while문

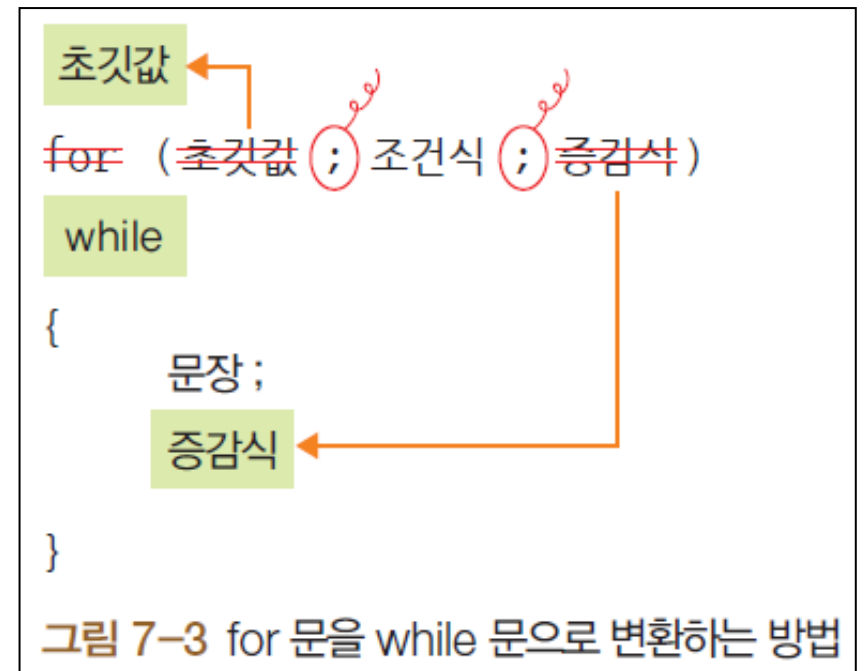
■ while문의 사용

- for문을 while문으로 바꾸기 : [실습7-1]은 [실습 6-2]의 'for (i=0 ; i<5 ; i++)' 문을 while문으로 바꾼 것임.

실습 6-2 기본 for 문 사용 예

```
01 public class Ex06_02 {  
02     public static void main(String[] args) {  
03         int i;  
04  
05         for (i = 0; i < 5; i++)  
06         {  
07             System.out.printf("안녕하세요? 빙글빙글 for 문을 공부중입니다.^w^n");  
08         }  
09     }  
10 }
```

for 문을
사용해서
다섯 번
반복한다.



01 while문

■ While문의 활용

```
1 //실습7-1. for문을 while문으로 바꾸기1
2 public class Ex07_01 {
3     public static void main(String[] args) {
4         int i;
5         i = 0;                //초깃값은 while문 위로 이동한다.
6
7         while (i < 5) {        //조건식은 while문과 함께 놓는다.
8             System.out.printf("while 문을 공부합니다.\n");
9             i++;               // 증감식은 while문 블록({ })안으로 이동한다.
10        }
11    }
12 }
```

For문과 비교해보면 초깃값을 while문 위로 빼고,
증감식은 while문 블록안의 맨 아래에 둔다.

Problems @ Javadoc Declaration Console

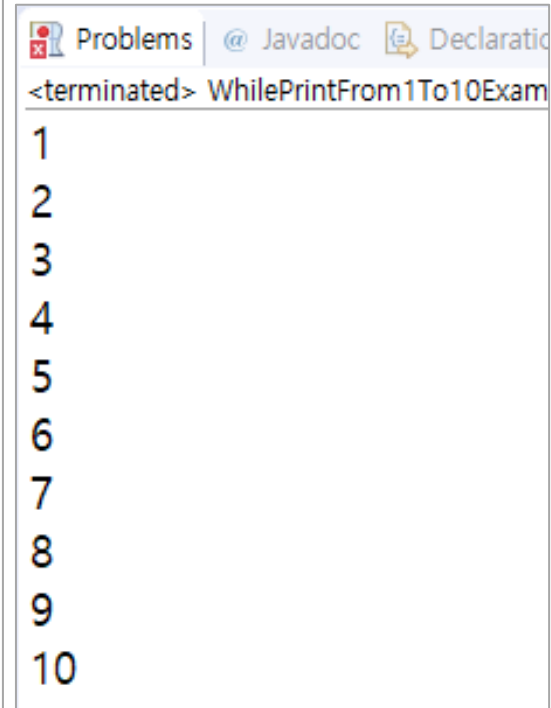
<terminated> Ex07_01 [Java Application] C:\Program Files\Java\jdk-11.

while 문을 공부합니다.
while 문을 공부합니다.
while 문을 공부합니다.
while 문을 공부합니다.
while 문을 공부합니다.

01 while문

■ While문 사용

```
1 //예1. 1부터 10까지 출력
2 package sec02.exam06;
3
4 public class WhilePrintFrom1To10Example {
5     public static void main(String[] args) {
6         int i = 1;
7         while (i <= 10) {
8             System.out.println(i);
9             i++;
10        }
11    }
12 }
```



Problems @ Javadoc Declaratic
<terminated> WhilePrintFrom1To10Exam
1
2
3
4
5
6
7
8
9
10

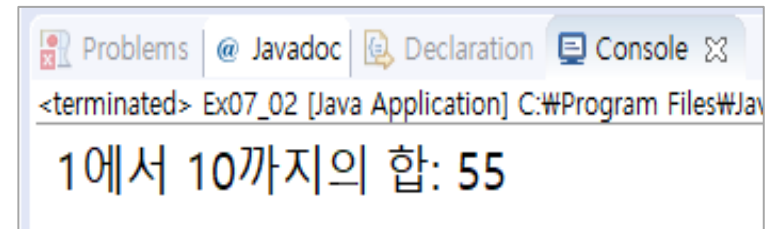
while문은 한번 실행할때 루프 카운터 변수 **i**값을 출력하고 **1** 증가시킵니다. 조건문은 **i**가 **10**이하일때까지 **true**가 되므로 **while**문은 총 **10**번을 반복 실행함

01 while문

■ While문을 사용하여 합계 구하기

```
1 //실습7-2. for문을 while문으로 바꾸기2
2 public class Ex07_02 {
3     public static void main(String[] args) {
4         int hap = 0;
5         int i;
6
7         i = 1;                //초깃값을 지정한다.
8         while (i <= 10) {     //조건식이다.
9             hap = hap + i;
10            i++;              //증감식이다.
11        }
12
13        System.out.printf(" 1에서 10까지의 합: %d \n", hap);
14    }
15 }
```

for(i=1;i<=10;1++)에서 초깃값 **i=1**은 **while**문 밖의 7행으로, 조건식 **i<=10**은 **while**문의 조건식을, 증감식 **i++**는 블록(**{ }**)안의 가장 끝 행으로 옮기면 됨



01 while문

■ 무한 루프를 위한 while 문

- 무한 루프를 적용하려면 while (조건식)이 무조건 참. 즉 while (true)라고 쓰면 이는 for (;;)과 동일한 기능 임

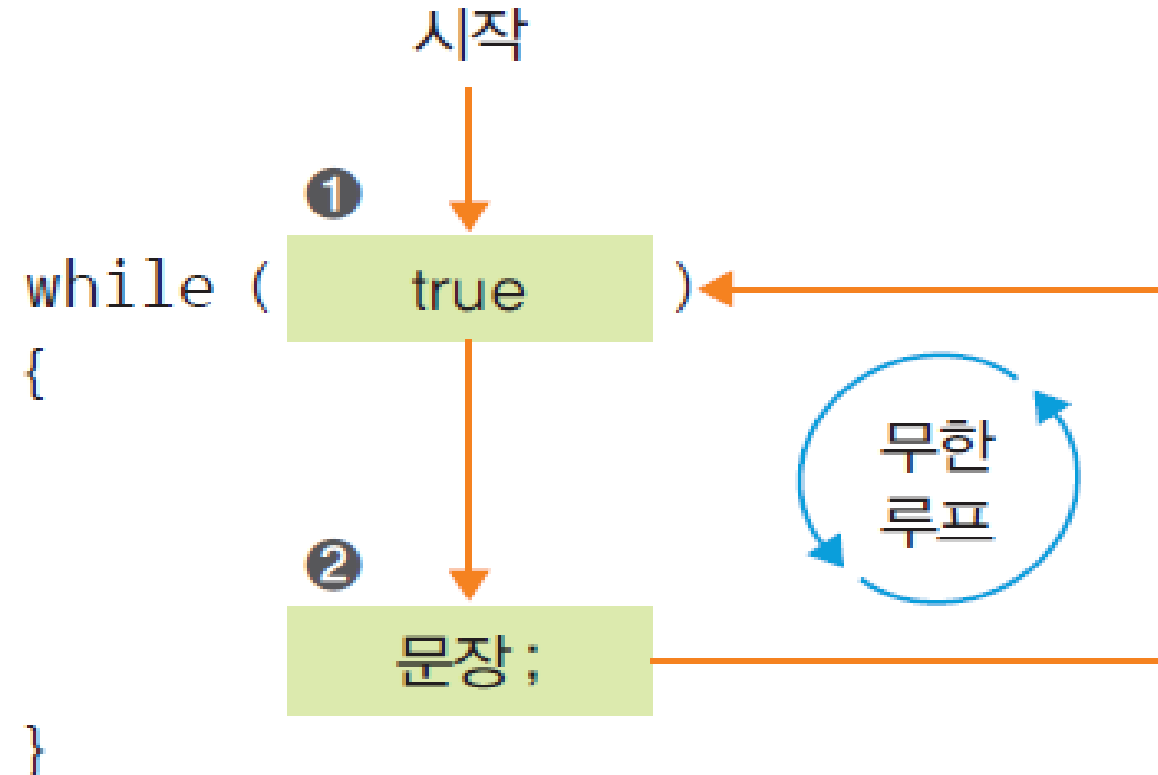


그림 7-5 while을 이용한 무한 루프

01 while문

■ while문의 무한루프

```
1 //실습7-3 예3. while문의 무한루프 만들기
2 import java.util.Scanner;
3
4 public class Ex07_03 {
5     public static void main(String[] args) {
6         Scanner s = new Scanner(System.in);
7         int a, b;
8
9         while (true) {
10             System.out.printf("더할 첫 번째 수 입력 : ");
11             a = s.nextInt(); //값을 입력한다.
12             System.out.printf("더할 두 번째 수 입력 : ");
13             b = s.nextInt(); //값을 입력한다.
14
15             System.out.printf("%d + %d = %d \n", a, b, a + b); //결과를 출력한다.
16         }
17     }
18 }
19 }
20 }
```

TIP. 무한루프에 **for(;;)**이나 **while (true)** 중 어떤 것을 사용해도 되지만 일반적으로 **for**문보다 **while**문을 더 많이 사용한다.



Problems @ Javadoc Declaration Console

Ex07_03 [Java Application] C:\Program Files\Java\jdk-11.0.6\bin

더할 첫 번째 수 입력 : 55
더할 두 번째 수 입력 : 22
55 + 22 = 77
더할 첫 번째 수 입력 : 77
더할 두 번째 수 입력 : 128
77 + 128 = 205
더할 첫 번째 수 입력 :

01 while문

■ while문의 무한루프

```
1 //실습7-4 무한루프를 활용한 계산기
2 import java.io.IOException;
3 import java.util.Scanner;
4
5 public class Ex07_04 {
6     public static void main(String[] args) throws IOException {
7         Scanner s = new Scanner(System.in);
8         int a, b;
9         char ch;
10
11         while (true) {
12             System.out.printf("계산할 첫번째 수를 입력 : ");
13             a = s.nextInt();
14             System.out.printf("계산할 두번째 수를 입력 : ");
15             b = s.nextInt();
16             System.out.printf("계산할 연산자를 입력하세요 : ");
17             ch = (char) System.in.read();
```

01 while문

■ while문의 무한루프

```
18
19     switch (ch) {
20     case '+':
21         System.out.printf("%d + %d = %d 입니다. \n", a, b, a + b);
22         break;
23     case '-':
24         System.out.printf("%d - %d = %d 입니다. \n", a, b, a - b);
25         break;
26     case '*':
27         System.out.printf("%d * %d = %d 입니다. \n", a, b, a * b);
28         break;
29     case '/':
30         System.out.printf("%d / %d = %f 입니다. \n", a, b, a / (float) b);
31         break;
32     case '%':
33         System.out.printf("%d %% %d = %d 입니다. \n", a, b, a % b);
34         break;
35     default:
36         System.out.printf("연산자를 잘못 입력했습니다. \n");
37     }
38 }
39
40 }
41 }
```

Problems @ Javadoc Declaration Console

Ex07_04 [Java Application] C:\Program Files\Java\jdk-11.0.6\

계산할 첫번째 수를 입력 : 10
계산할 두번째 수를 입력 : 2
계산할 연산자를 입력하세요 : +
10 + 2 = 12 입니다.
계산할 첫번째 수를 입력 : 10
계산할 두번째 수를 입력 : 2
계산할 연산자를 입력하세요 : *
10 * 2 = 20 입니다.
계산할 첫번째 수를 입력 : 10
계산할 두번째 수를 입력 : 2
계산할 연산자를 입력하세요 : /
10 / 2 = 5.000000 입니다.
계산할 첫번째 수를 입력 : 10
계산할 두번째 수를 입력 : 2
계산할 연산자를 입력하세요 : %
10 % 2 = 0 입니다.
계산할 첫번째 수를 입력 :

02

do~while문

02 do~while문

■ do~while문

- while 문이나 for 문은 조건식이 처음부터 거짓이면 한 번도 수행하지 않고 종료
- do~while 문은 어떤 경우라도 한번은 수행하는데 자주 사용하지 않지만 유용할 때가 있음
- do~while 문은 조건식에 의해 반복 실행하는 점에서 while문과 동일하나 조건식이 아래쪽에 있다는 것이 다름
- while문은 처음에 조건식이 거짓일때 '문장'을 한번도 수행하지 않음
- do~while 문에서는 조건식을 확인하기 전에 일단 '문장'을 수행하므로 조건식이 거짓이든 참이든 무조건 한번은 수행함

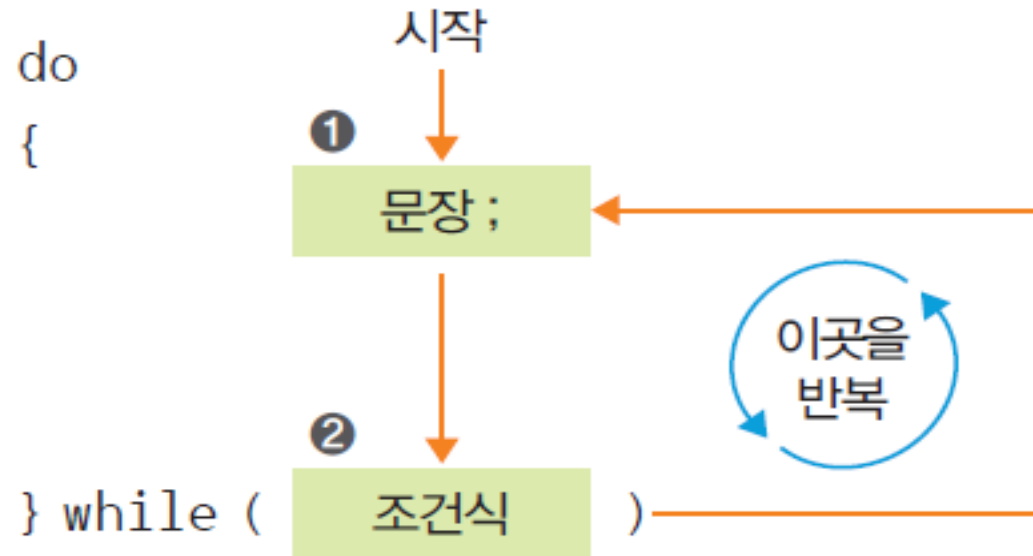
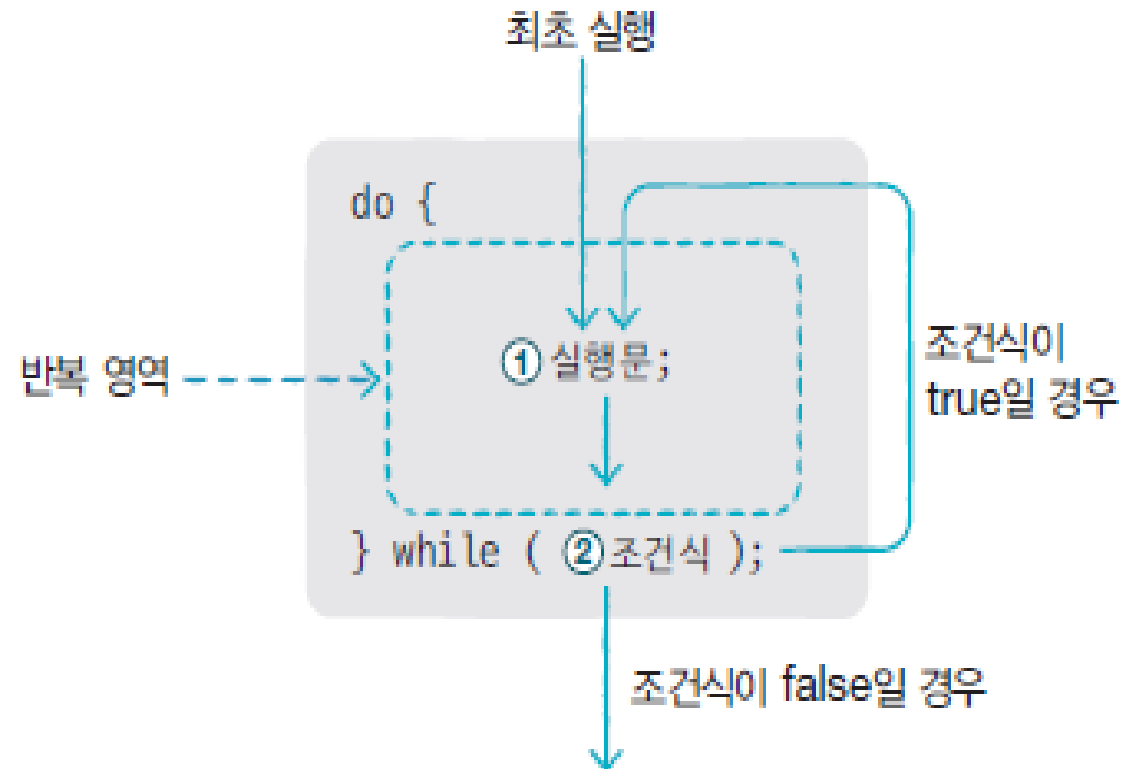


그림 7-8 do~while 문의 형식과 실행 순서

02 do~while문

■ do~while문

1. do~while 문이 처음 실행될 때 실행문(①)을 우선 실행함
2. 실행문(①)이 모두 실행되면 조건식(②)을 평가함
3. 조건식의 결과가 true이면 실행문(①)→ 조건식(②)과 같이 반복 실행함
4. 조건식의 결과가 false이면 do~while문을 종료함



02 do~while문

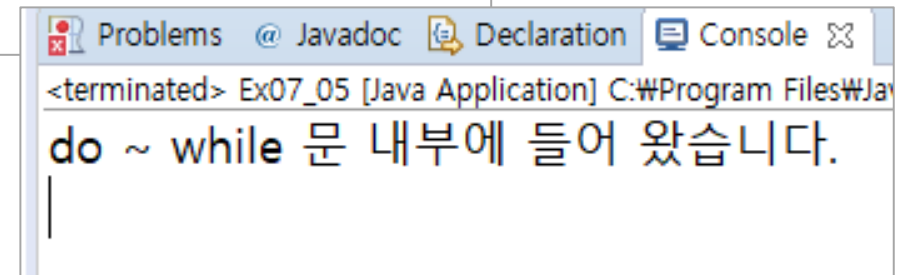
■ do~while문의 활용

```
1 //실습7-5 do~while문 사용 예1
2 public class Ex07_05 {
3     public static void main(String[] args) {
4         int a = 100;
5
6         while (a == 200) {                // while문 실행 : 먼저 조건식을 판단
7             System.out.printf("while 문 내부에 들어 왔습니다.\n");
8         }
9
10        do {                             //do~while문 실행 :
11            System.out.printf("do ~ while 문 내부에 들어 왔습니다.\n"); //먼저 내용을 실행한 다음
12        } while (a == 200);               //조건식을 판단한다.
13    }
14 }
```

4행을 보면 현재 a가 100임.

6행의 while문은 조건식을 먼저 판단하기 위해 7행을 실행하지 않지만,

10행의 do~while문은 11행을 수행한 다음 12행의 조건식을 판단함



02 do~while문

■ do~while문의 활용

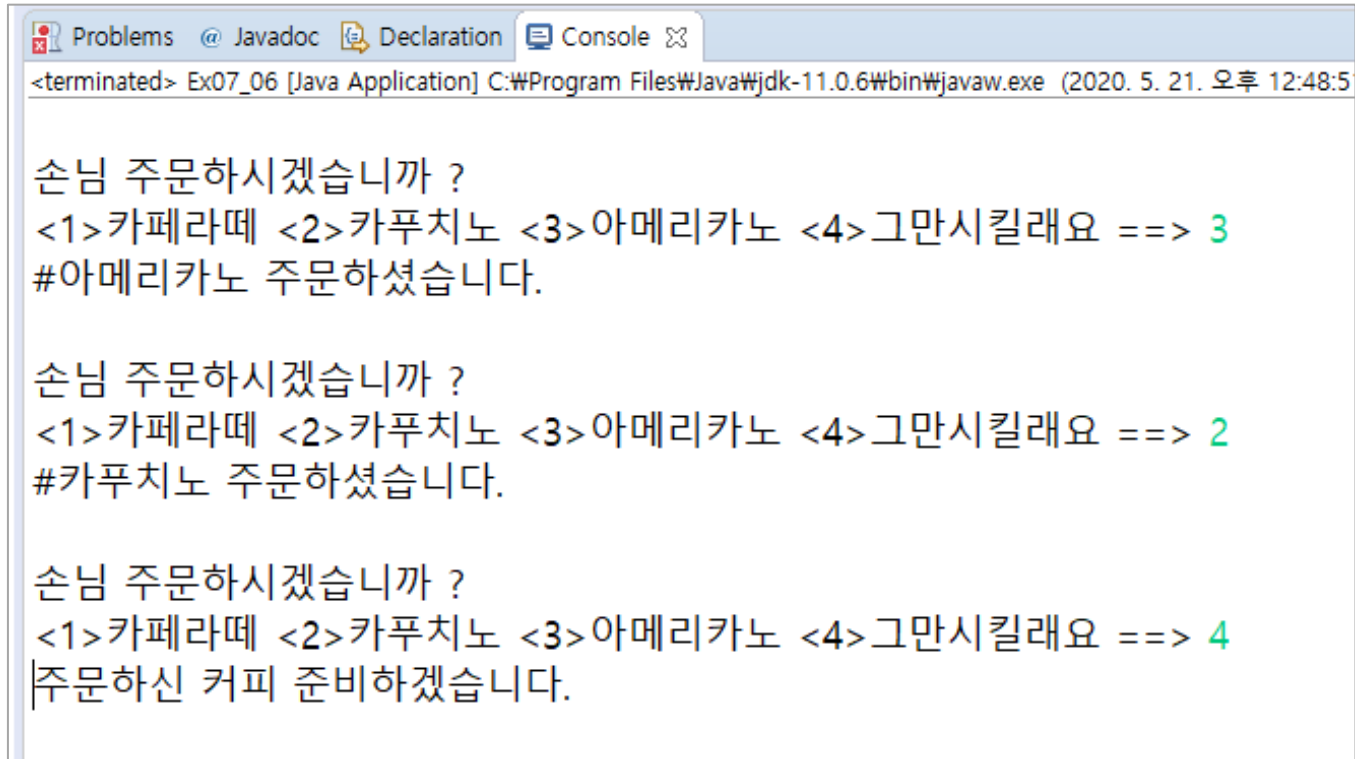
9~23행은 do~while문이므로
적어도 한번 이상 실행됨

10행~12행은 종업원이 주문을
받는 과정이며,
손님이 마음에 드는 커피가 없다면
바로 4번을 선택하고 나감

```
1 //실습7-6 do~while문 사용 예2
2 import java.util.Scanner;
3
4 public class Ex07_06 {
5     public static void main(String[] args) {
6         Scanner s = new Scanner(System.in);
7         int menu;
8
9         do { //do~while문이므로 반드시 한번은 수행
10            System.out.printf("₩n손님 주문하시겠습니까 ? ₩n");
11            System.out.printf("<1>카페라떼 <2>카푸치노 <3>아메리카노 <4>그만시킬래요 ==> "); //메뉴출력
12            menu = s.nextInt(); //커피 종류선택
13            switch (menu) { //선택한 커피에 따라 주문 접수
14                case 1:
15                    System.out.printf("#카페라떼 주문하셨습니다.₩n"); break;
16                case 2:
17                    System.out.printf("#카푸치노 주문하셨습니다.₩n"); break;
18                case 3:
19                    System.out.printf("#아메리카노 주문하셨습니다.₩n"); break;
20                case 4:
21                    System.out.printf("주문하신 커피 준비하겠습니다.₩n"); break;
22                default:
23                    System.out.printf("잘못 주문하셨습니다.₩n");
24            }
25        } while (menu != 4); //선택한 메뉴가 4번이 아니라면 계속 반복해서 주문을 받는다.
26
27        s.close();
28    }
29 }
```

02 do~while문

■ do~while문의 활용



```
Problems @ Javadoc Declaration Console
<terminated> Ex07_06 [Java Application] C:\Program Files\Java\jdk-11.0.6\bin\javaw.exe (2020. 5. 21. 오후 12:48:5
손님 주문하시겠습니까 ?
<1>카페라떼 <2>카푸치노 <3>아메리카노 <4>그만시킬래요 ==> 3
#아메리카노 주문하셨습니다.

손님 주문하시겠습니까 ?
<1>카페라떼 <2>카푸치노 <3>아메리카노 <4>그만시킬래요 ==> 2
#카푸치노 주문하셨습니다.

손님 주문하시겠습니까 ?
<1>카페라떼 <2>카푸치노 <3>아메리카노 <4>그만시킬래요 ==> 4
주문하신 커피 준비하겠습니다.
```

Tip. 실제로 프로그래밍을 할 때는 **do~while**문을 자주 사용하지 않는다. **While**문을 조금만 변형하면 충분히 구현될 수 있기 때문임

Self study 9-1

1. [실습6-11]을 while문으로 바꿔보자.
2. [실습7-2]를 do~while문으로 바꿔보자.
3. [실습7-6]에서 사용한 switch~case문을 중첩 if 문으로 바꿔보자.
4. 다음 빈칸에 알맞은 말을 넣으시오.

for문은 for(초깃값;조건식;증감값) 형식이지만 while문은 while ()으로 표현한다. 그러므로 for문을 while 문으로 변환하려면 (),
() 이 없기 때문에 별도로 처리해야 한다.

질문은 이메일을 이용해주세요.
ds.june2@gmail.com

감사합니다