



12주 3강

기본 자바 소개 5



송실사이버대학교

송실사이버대학교의 강의콘텐츠는
저작권법에 의하여 보호를 받는바, 무단
전재, 배포, 전송, 대여 등을 금합니다.

*사용서체 : 나눔글꼴

이번 주차에는...

기본 자바 소개 5

- 반복문의 기본, for 문
- 반복문의 심화, while 문

1. 단순 for 문(1)

■ for 문의 개념

안녕하세요? 빙글빙글 for 문을 공부중입니다. ^^
안녕하세요? 빙글빙글 for 문을 공부중입니다. ^^
안녕하세요? 빙글빙글 for 문을 공부중입니다. ^^
안녕하세요? 빙글빙글 for 문을 공부중입니다. ^^
안녕하세요? 빙글빙글 for 문을 공부중입니다. ^^

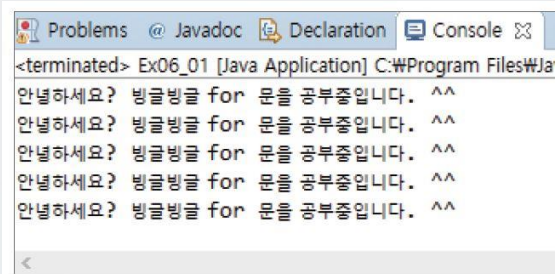


그림 6-1 실행 결과

실습 6-1 같은 문장을 반복해서 출력

```
01 public class Ex06_01 {  
02     public static void main(String[] args) {  
03         System.out.printf("안녕하세요? 빙글빙글 for 문을 공부중입니다. ^^n");  
04         System.out.printf("안녕하세요? 빙글빙글 for 문을 공부중입니다. ^^n");  
05         System.out.printf("안녕하세요? 빙글빙글 for 문을 공부중입니다. ^^n");  
06         System.out.printf("안녕하세요? 빙글빙글 for 문을 공부중입니다. ^^n");  
07         System.out.printf("안녕하세요? 빙글빙글 for 문을 공부중입니다. ^^n");  
08     }  
09 }
```

내용을
출력한다.

2. 단순 for 문(2)

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.^^\n");  
08         }  
09     }  
10 }
```

for 문을
사용해서
다섯 번
반복한다.

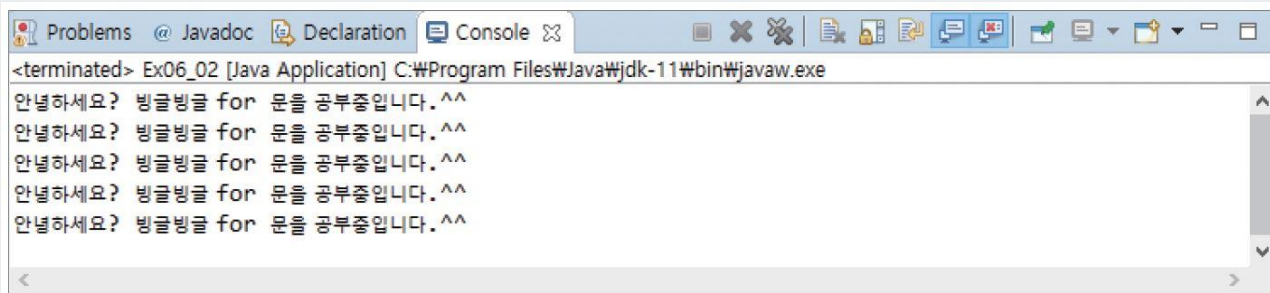
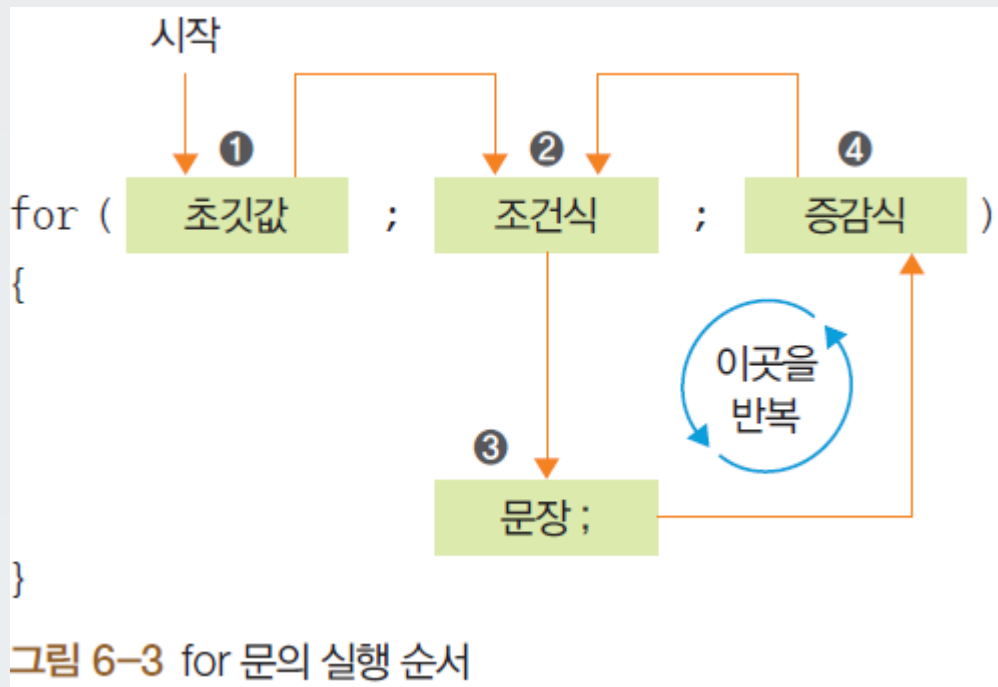


그림 6-2 실행 결과

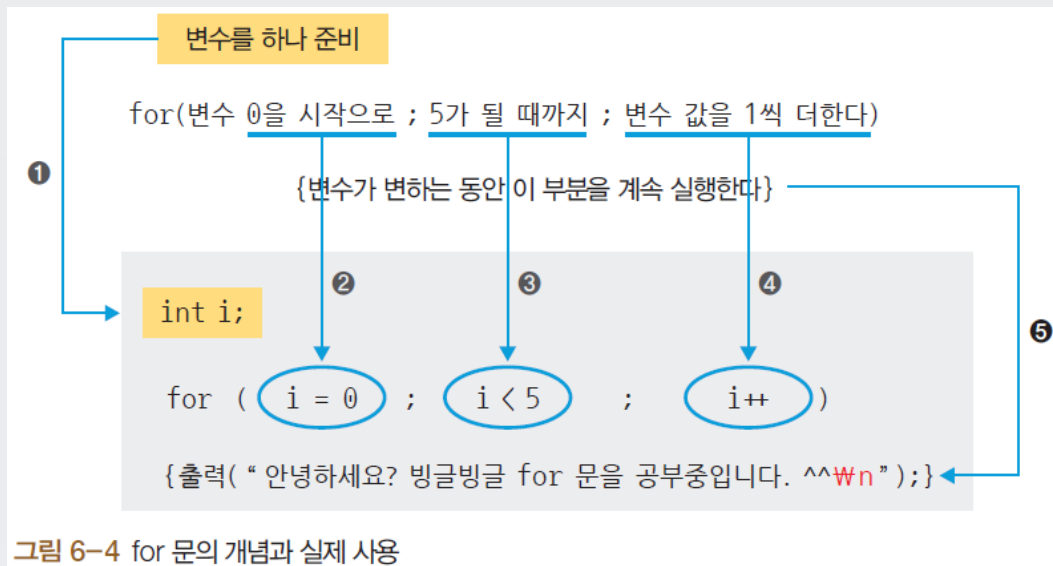
3. 단순 for 문(3)

- for 문의 활용



4. 단순 for 문(4)

- [실습 6-2]의 5~8행에서 사용한 for 문의 기본 구조



초깃값 → 조건식 → 반복할 문장 → 증감식 → 조건식 → 반복할 문장 → 증감식 → 조건식 → 반복할 문장
→ 증감식 → 조건식 ...

5. 단순 for 문(5)

❶ 초기값 수행

```
int i;  
for ( i=0 ; i < 3 ; i++ )  
{  
    출력 ("안녕하세요? IT CookBook.\n");  
}
```



❷ 조건식 확인

```
int i;  
for( i=0 ; i < 3 ; i++ )  
{  
    출력 ("안녕하세요? IT CookBook.\n");  
}
```

조건이
거짓이면

❸ 반복문 탈출



조건이 참이면



6. 단순 for 문(6)

③ 반복할 문장 실행

```
int i;  
for( i=0 ; i < 3 ; i++ )  
{  
    출력 ("안녕하세요? IT CookBook.\n");  
}
```



④ 증감식 실행

```
int i;  
for( i=0 ; i < 3 ; i++ )  
{  
    출력 ("안녕하세요? IT CookBook.\n");  
}
```

그림 6-5 for 문이 반복되는 순서

7. 단순 for 문(7)

- 1회 : ❶ 초깃값을 수행한다(현재 $i=0$).
- 제2회 : ❷ 조건식을 확인한다. 현재 i 값이 0이므로 $i<3$ 는 참이다.
- 제3회 : ❸ `System.out.printf` 문을 수행한다('안녕하세요? ...' 출력).
- 제4회 : ❹ 증감식 $i++$ 를 수행하여 i 값을 1 증가시킨다(현재 $i=1$).
- 제5회 : 다시 ❷ 조건식을 확인한다. 현재 i 값이 1이므로 $i<3$ 는 참이다.
- 제6회 : 다시 ❸ `System.out.printf` 문을 수행한다('안녕하세요? ...' 출력).
- 제7회 : 다시 ❹ 증감식 $i++$ 를 수행하여 i 값을 1 증가시킨다(현재 $i=2$).
- 제8회 : 다시 ❷ 조건식을 확인한다. 현재 i 값이 2이므로 $i<3$ 는 참이다.
- 제9회 : 다시 ❸ `System.out.printf` 문을 수행한다('안녕하세요? ...' 출력).
- 제10회 : 다시 ❹ 증감식 $i++$ 를 수행하여 i 값을 1 증가시킨다(현재 $i=3$).
- 제11회 : 다시 ❷ 조건식을 확인한다. 현재 i 값이 3이므로 드디어 $i<3$ 가 거짓이다.
- 제12회 : 조건이 거짓이므로 ❺ 반복문을 탈출하고 반복문 블록({ }) 밖의 내용을 수행한다.

8. 단순 for 문(8)

- 실행할 문장이 하나이면 중괄호로 묶지 않아도 되지만, 반복할 문장이 2개 이상이면 반드시 중괄호로 묶어야 함

```
int i;  
for (i=0 ; i < 3 ; i++)  
{  
    System.out.printf("안녕하세요? 빙글  
    빙글 for 문을 공부중입니다. ^ ^\n");  
}
```

==

```
int i;  
for (i=0 ; i < 3 ; i++)  
    System.out.printf("안녕하세요? 빙글  
    빙글 for 문을 공부중입니다. ^ ^\n");
```

9. 단순 for 문(9)

실습 6-3 for 문과 중괄호 사용 예

```
01 public class Ex06_03 {  
02     public static void main(String[] args) {  
03         int i;  
04         for (i = 0; i < 3; i++)  
05         {  
06             System.out.printf("안녕하세요? Wn");  
07             System.out.printf("##또 안녕하세요?## Wn");  
08         }  
09  
10         System.out.printf("WnWn");  
11  
12         for (i = 0; i < 3; i++)  
13             System.out.printf("안녕하세요? Wn");  
14             System.out.printf("##또 안녕하세요?## Wn");  
15  
16     }  
17 }
```

for 문에 중괄호를 사용했다.

for 문에 중괄호를 사용하지 않았다.

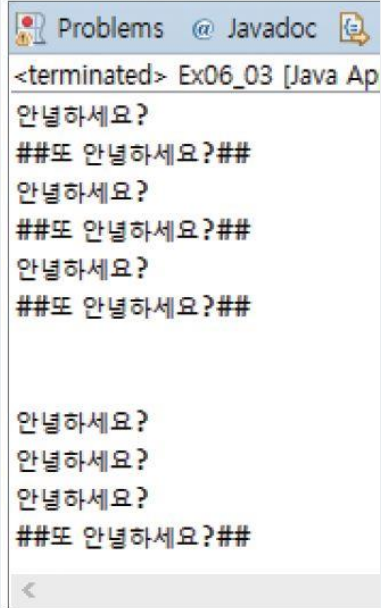


그림 6-6 실행 결과

10. 단순 for 문(10)

■ for 문 연습

실습 6-4 for 문 사용 예 1

```
01 public class Ex06_04 {  
02     public static void main(String[] args) {  
03         int i;  
04  
05         for (  ) { ----- 초깃값, 조건식, 증감식을 수정하여 5에서 1씩 줄여간다.  
06             System.out.printf("%d : 안녕하세요? 빙글빙글 for 문을 공부중입니다. ^w^n", i);  
07         }  
08     }  
09 }
```

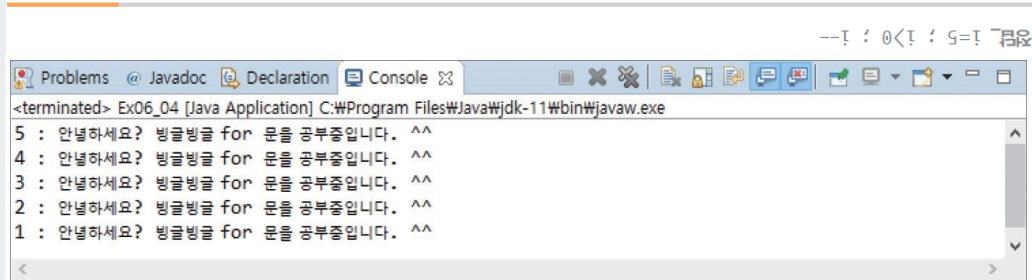


그림 6-7 실행 결과

11. 단순 for 문(11)

실습 6-5 for 문 사용 예 2

```
01 public class Ex06_05 {  
02     public static void main(String[] args) {  
03         int i;  
04  
05         for (i = 1; i <= 5; i++) { ----- i 값이 1부터 5까지 변경된다.  
06             System.out.printf("%d \n", i);  
07         }  
08     }  
09 }
```

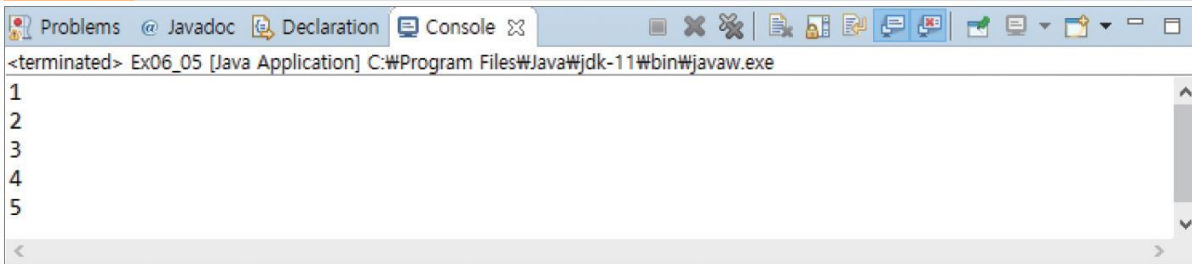


그림 6-8 실행 결과

12. 단순 for 문(12)

■ for 문을 활용하여 합계 구하기

실습 6-6 for 문을 사용하지 않고 합계 구하기

```
01 public class Ex06_06 {  
02     public static void main(String[] args) {  
03         int hap;  
04  
05         hap = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10; ----- hap에 1부터 10까지 더해 입력한다.  
06  
07         System.out.printf(" 1에서 10까지의 합: %d \n", hap);  
08     }  
09 }
```

합계가 들어갈 변수 준비(hap)

1부터 10까지 변할 변수 준비(i)

```
for (i가 1을 시작으로 ; 10보다 작거나 같을 때까지 ; i가 1씩 증가)  
{  
    hap 값에 i 값을 더해줌  
}
```



그림 6-9 실행 결과

13. 단순 for 문(13)

실습 6-7 for 문을 사용하여 합계 구하기 1

```
01 public class Ex06_07 {
02     public static void main(String[] args) {
03         int hap; ----- 합계를 누적할 변수를 선언한다.
04         int i; ----- 1부터 10까지 변하는 변수를 선언한다.
05
06         for (i = 1; i <= 10; i++) {
07             hap = hap + i; ----- hap 변수에 1부터 10까지 반복해서 누적한다.
08         }
09
10         System.out.printf(" 1에서 10까지의 합: %d \n", hap);
11     }
12 }
```

for 문에 의해 1부터 10까지 10회 반복된다.

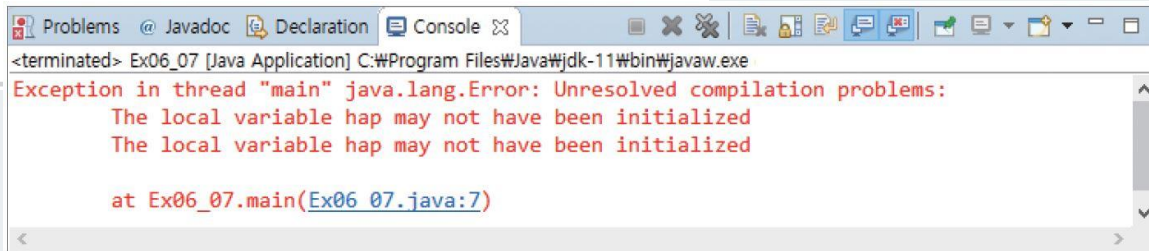


그림 6-10 실행 결과

14. 단순 for 문(14)

실습 6-8 for 문을 사용하여 합계 구하기 2

```
01 public class Ex06_08 {  
02     public static void main(String[] args) {  
03         int hap = 0;        ----- 합계를 누적할 변수를 선언하고 0으로 초기화한다.  
04         int i;  
05  
06         for (i = 1; i <= 10; i++) {  
07             hap += i;        ----- hap 변수에 1부터 10까지 반복해서 누적한다. hap=hap+i와 동일하다.  
08         }  
09  
10         System.out.printf(" 1에서 10까지의 합: %d \n", hap);  
11     }  
12 }
```

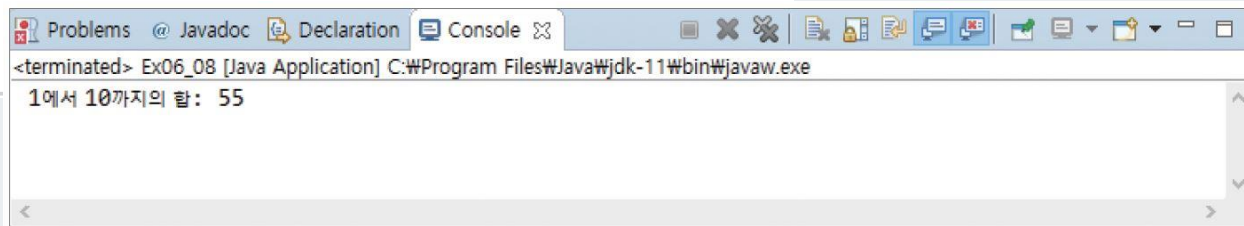
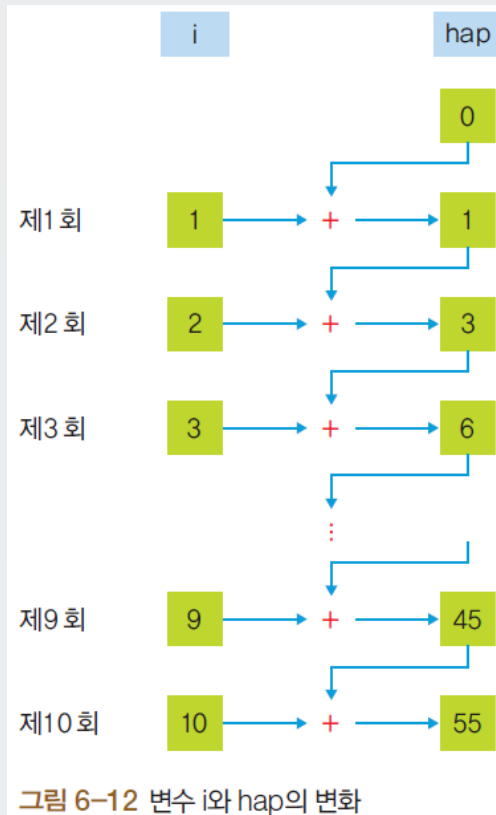


그림 6-11 실행 결과

15. 단순 for 문(15)

- [실습 6-8]의 변수 i와 hap의 변화



16. 단순 for 문(16)

실습 6-9 for 문을 사용하여 합계 구하기 3

```
01 public class Ex06_09 {  
02     public static void main(String[] args) {  
03         int hap = 0;  
04         int i;  
05  
06         for (  ; i <= 1000;  ) { ----- i를 501부터 2씩 증가시킨다.  
07             hap += i;  
08         }  
09  
10         System.out.printf(" 500에서 1000까지의 홀수의 합: %d \n", hap);  
11     }  
12 }
```

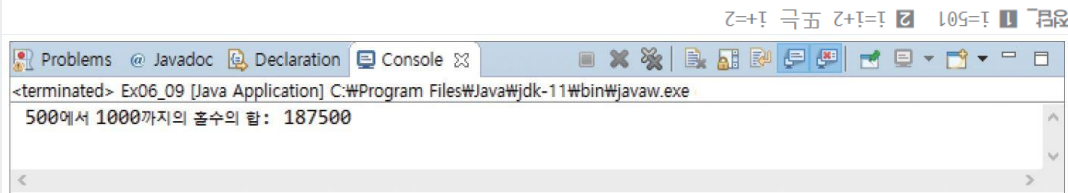


그림 6-13 실행 결과

17. 단순 for 문(17)

실습 6-10 for 문을 사용하여 합계 구하기 4

```
01 import java.util.Scanner;
02
03 public class Ex06_10 {
04     public static void main(String[] args) {
05         Scanner s = new Scanner(System.in);
06         int hap = 0;          ----- 합계를 누적할 변수를 선언하고 0으로 초기화한다.
07         int i;                ----- 1씩 증가할 변수를 선언한다.
08         int num;              ----- 입력받을 최종 값을 선언한다.
09
10         System.out.printf(" 값 입력 : ");
11         num = s.nextInt();     ----- 최종 값을 입력한다.
12
13         for (i = 1; i <= num; i++) { ----- 1부터 최종 값까지 1씩 증가시키며 반복한다.
14             hap = hap + i;
15         }
16
17         System.out.printf(" 1에서 %d까지의 합: %d \n", num, hap);
18     }
19 }
```



그림 6-14 실행 결과

18. 단순 for 문(18)

실습 6-11 for 문을 사용하여 합계 구하기 5

```
01 import java.util.Scanner;
02
03 public class Ex06_11 {
04     public static void main(String[] args) {
05         Scanner s = new Scanner(System.in);
06         int hap = 0;
07         int i;
08         int num1, num2, num3; ----- 입력받을 세 변수를 선언한다.
09     }
```

19. 단순 for 문(19)

```
10    System.out.printf(" 시작값 입력 : ");
11    num1 = s.nextInt();
12    System.out.printf(" 끝값 입력 : ");
13    num2 = s.nextInt();
14    System.out.printf(" 증가값 입력 : ");
15    num3 = s.nextInt();
16
17    for (  ) { ----- 시작 값은 num1, 최종 값은 num2, 증가 값은 num3을 사용한다.
18        hap = hap + i;
19    }
20
21    System.out.printf(" %d에서 %d까지 %d씩 증가한 값의 합: %d \n", num1, num2, num3, hap);
22 }
23 }
```

εwnu+I=I : 7W



그림 6-15 실행 결과

20. 단순 for 문(20)

실습 6-12 for 문을 사용한 구구단 프로그램

```
01 import java.util.Scanner;
02
03 public class Ex06_12 {
04     public static void main(String[] args) {
05         Scanner s = new Scanner(System.in);
06         int i;
07         int dan; ----- 계산할 단을 입력받은 변수를 선언한다.
08
09         System.out.printf(" 몇 단 ? ");
10         dan = s.nextInt(); ----- 계산할 단을 입력받는다.
11
12         for (i = 1; i <= 9; i++) {
13             System.out.printf(" %d X %d = %d \n", dan, i, dan * i);
14         }
15     }
16 }
```

입력한 단에 대한
구구단을 1부터 9까지
반복해서 출력한다.

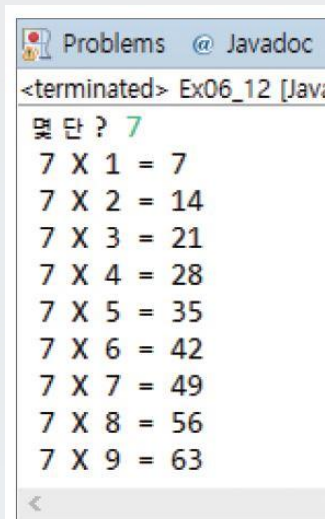
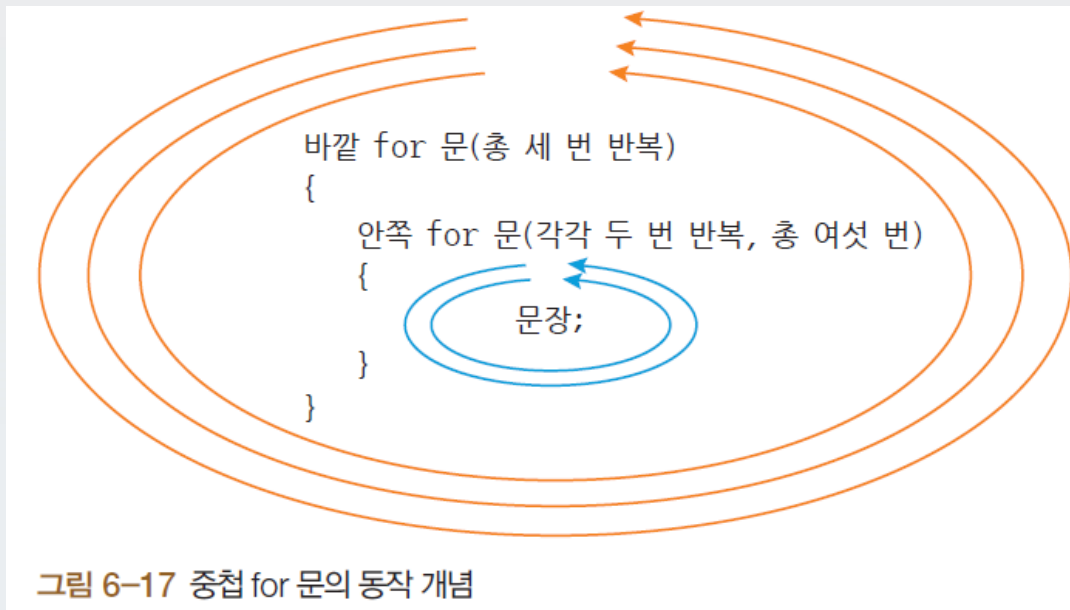


그림 6-16 실행 결과

21. 중첩 for 문(1)

- 중첩 for 문의 개념
 - for 문 내부에 또 다른 for 문이 들어 있는 형태



22. 중첩 for 문(2)

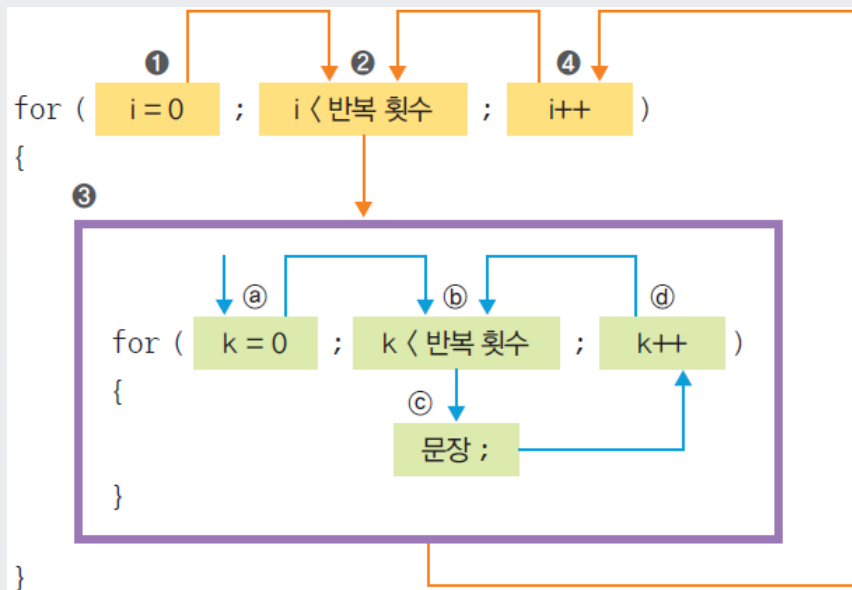


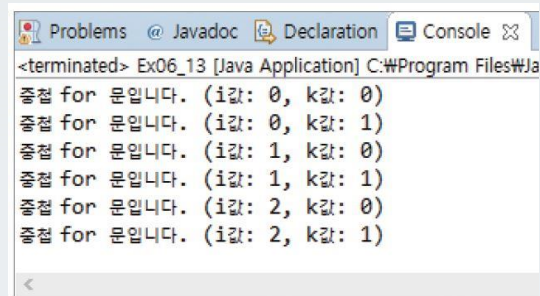
그림 6-18 중첩 for 문의 작동 방식

① → ② → ③ → (a → b → c → d → b → c → d → b → 안쪽 for 문을 빠져나감) → ④ → ② → ③
 → (a → b → c → d → b → c → d → b → 안쪽 for 문을 빠져나감) → ④ → ② → ③ → (a → b → c
 → d → b → c → d → b → 안쪽 for 문을 빠져나감) → ④ → ② → 바깥 for 문을 빠져나감

23. 중첩 for 문(3)

실습 6-13 중첩 for 문 사용 예 1

```
01 public class Ex06_13 {
02     public static void main(String[] args) {
03         int i, k;          ----- 반복할 변수 i, k를 선언한다.
04
05         for (i = 0; i < 3; i++) ----- 바깥 for 문을 세 번 반복한다.
06         {
07             for (k = 0; k < 2; k++) ----- 안쪽 for 문을 두 번 반복한다.
08             {
09                 System.out.printf("중첩 for 문입니다. (i값: %d, k값: %d)\n", i, k);
                                                                i와 k 값을 총 여섯 번(=3×2) 출력한다.
10             }
11         }
12
13     }
14 }
```



```
Problems @ Javadoc Declaration Console
<terminated> Ex06_13 [Java Application] C:\Program Files\Ja
중첩 for 문입니다. (i값: 0, k값: 0)
중첩 for 문입니다. (i값: 0, k값: 1)
중첩 for 문입니다. (i값: 1, k값: 0)
중첩 for 문입니다. (i값: 1, k값: 1)
중첩 for 문입니다. (i값: 2, k값: 0)
중첩 for 문입니다. (i값: 2, k값: 1)
```

그림 6-19 실행 결과

24. 중첩 for 문(4)

■ [실습 6-13]의 처리 순서

①

외부 for 문 1회 : 5행의 i 를 0으로 초기화한 후 ' $i < 3$ '가 참이므로 바깥 for 문 수행

내부 for 문 1회 : 7행의 k 를 0으로 초기화한 후 ' $k < 2$ '가 참이므로 안쪽 for 문 수행

9행의 `System.out.printf()`를 실행하여 '중첩 for 문입니다.' 출력

내부 for 문 2회 : 7행의 $k++$ 로 k 를 1로 증가시킨 후 ' $k < 2$ '가 참이므로 안쪽 for 문 수행

9행의 `System.out.printf()`를 실행하여 '중첩 for 문입니다.' 출력

내부 for 문 3회 : 7행의 $k++$ 로 k 를 2로 증가시킨 후 ' $k < 2$ '가 거짓이므로 안쪽 for 문 종료

②

외부 for 문 2회 : 5행의 $i++$ 로 i 를 1로 증가시킨 후 ' $i < 3$ '가 참이므로 바깥 for 문 수행

내부 for 문 1회 : 7행의 k 를 0으로 초기화한 후 ' $k < 2$ '가 참이므로 안쪽 for 문 수행

9행의 `System.out.printf()`를 실행하여 '중첩 for 문입니다.' 출력

내부 for 문 2회 : 7행의 $k++$ 로 k 를 1로 증가시킨 후 ' $k < 2$ '가 참이므로 안쪽 for 문 수행

9행의 `System.out.printf()`를 실행하여 '중첩 for 문입니다.' 출력

내부 for 문 3회 : 7행의 $k++$ 로 k 를 2로 증가시킨 후 ' $k < 2$ '가 거짓이므로 안쪽 for 문 종료

25. 중첩 for 문(5)

③

외부 for 문 3회 : 5행의 $i++$ 로 i 를 2로 증가시킨 후 ' $i < 3$ '가 참이므로 바깥 for 문 수행

내부 for 문 1회 : 7행의 k 를 0으로 초기화한 후 ' $k < 2$ '가 참이므로 안쪽 for 문 수행

9행의 `System.out.printf()`를 실행하여 '중첩 for 문입니다.' 출력

내부 for 문 2회 : 7행의 $k++$ 로 k 를 1로 증가시킨 후 ' $k < 2$ '가 참이므로 안쪽 for 문 수행

9행의 `System.out.printf()`를 실행하여 '중첩 for 문입니다.' 출력

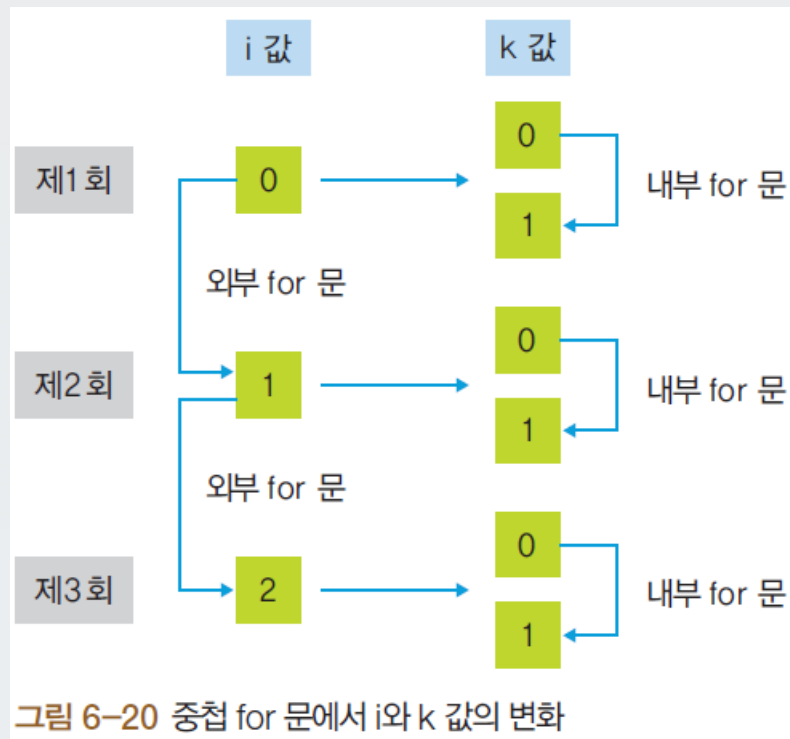
내부 for 문 3회 : 7행의 $k++$ 로 k 를 2로 증가시킨 후 ' $k < 2$ '가 거짓이므로 안쪽 for 문 종료

④

외부 for 문 4회 : 5행의 $i++$ 로 i 를 3으로 증가시킨 후 ' $i < 3$ '가 거짓이므로 바깥 for 문 종료

26. 중첩 for 문(6)

- 실습[6-13]의 중첩 for 문에서 i와 k 값의 변화



27. 중첩 for 문(7)

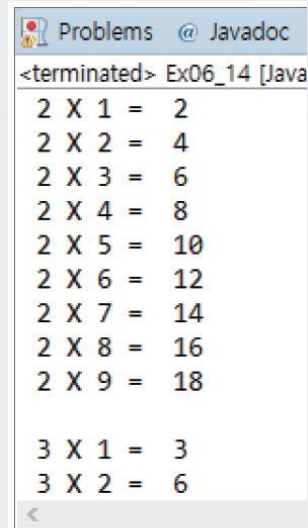
- 중첩 for 문의 활용
 - 구구단 2단~9단



28. 중첩 for 문(8)

실습 6-14 중첩 for 문 사용 예 2

```
01 public class Ex06_14 {
02     public static void main(String[] args) {
03         int i, k;
04
05         for (i = 2; i <= 9; i++) { ----- 2~9단을 반복한다.
06             for (  ) { ----- 각 단의 뒷자리 숫자 1~9를 반복한다.
07                 System.out.printf(" %d X %d = %d \n", i, k, i * k); ----- 구구단을 출력한다.
08             }
09             System.out.printf("\n"); ----- 각 단이 끝나면 한 줄 띄운다.
10         }
11
12     }
13 }
```



```
<terminated> Ex06_14 [Java]
2 X 1 = 2
2 X 2 = 4
2 X 3 = 6
2 X 4 = 8
2 X 5 = 10
2 X 6 = 12
2 X 7 = 14
2 X 8 = 16
2 X 9 = 18

3 X 1 = 3
3 X 2 = 6
```

그림 6-22 실행 결과

29. 중첩 for 문(9)

- 구구단에서 변수 i와 k의 추출(가로를 먼저 출력하는 경우)

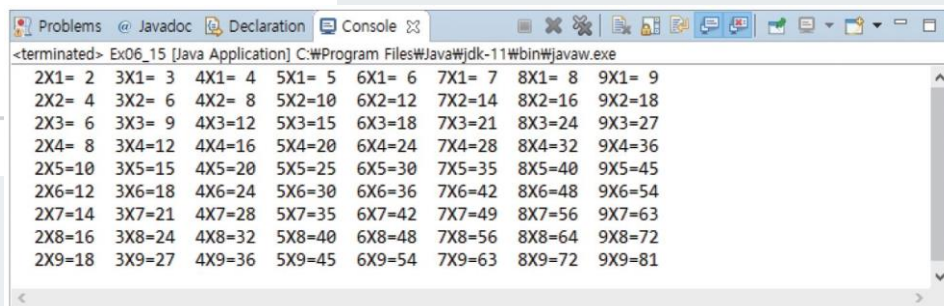


그림 6-23 구구단에서 변수 i와 k의 추출(가로를 먼저 출력하는 경우)

30. 중첩 for 문(10)

실습 6-15 중첩 for 문 사용 예 3

```
01 public class Ex06_15 {  
02     public static void main(String[] args) {  
03         int i, k;  
04  
05         for (i = 1; i <= 9; i++) { ----- 각 단의 뒷자리 숫자 1~9를 반복한다.  
06             for (k = 2; k <= 9; k++) { ----- 2~9단을 반복한다.  
07                 System.out.printf("%3dX%3d=",  ); ----- 각 단별로 한 줄씩 출력한다.  
08             }  
09             System.out.printf("\n"); ----- 각 단의 한 줄을 출력한 후 다음 줄로 넘긴다.  
10         }  
11  
12     }  
13 }
```



```
<terminated> Ex06_15 [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe  
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
```

그림 6-24 실행 결과

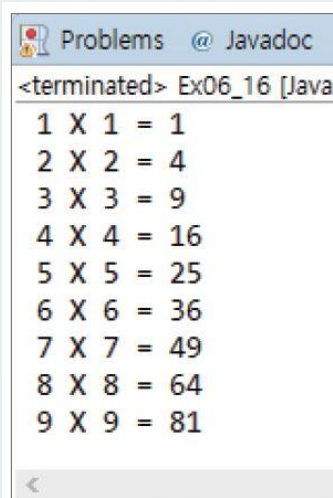
31. 기타 for 문(1)

- 여러 개의 초깃값과 증감식을 사용하는 for 문

for (초깃값 1, 초깃값 2; 조건식; 증감식 1, 증감식 2)

실습 6-16 다양한 for 문의 형태 1

```
01 public class Ex06_16 {  
02     public static void main(String[] args) {  
03         int i, k; ----- 반복할 변수 i와 k를 선언한다.  
04  
05         for (i = 1, k = 1; i <= 9; i++, k++) ----- 초깃값과 증감식이 2개이다.  
06             System.out.printf(" %d X %d = %d \n", i, k, i * k);  
07  
08     }  
09 }
```



Problems @ Javadoc
<terminated> Ex06_16 [Java]
1 X 1 = 1
2 X 2 = 4
3 X 3 = 9
4 X 4 = 16
5 X 5 = 25
6 X 6 = 36
7 X 7 = 49
8 X 8 = 64
9 X 9 = 81

그림 6-25 실행 결과

32. 기타 for 문(2)

- 초깃값과 증감식이 없는 for 문
 - 0~9를 출력하는 3가지 for 문

① 기본 형식

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

==

② 초깃값 빼기

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

==

③ 초깃값과 증감식 빼기

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

33. 기타 for 문(3)

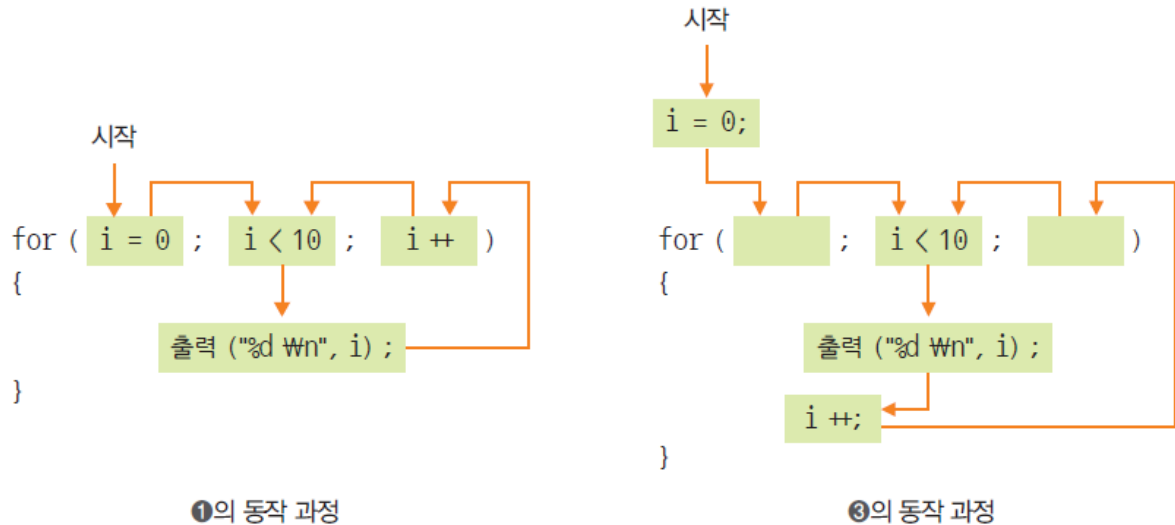


그림 6-26 두 소스코드의 비교

① 시작 → `i=0` → `i<10` → `System.out.printf()` 메소드 → `i++` → `i<10` → `System.out.printf()` 메소드 → `i++` → ...

③ 시작 → `i=0` → 빈칸 → `i<10` → `System.out.printf()` 메소드 → `i++` → 빈칸 → `i<10` → `System.out.printf()` 메소드 → `i++` → ...

34. 기타 for 문(4)

실습 6-17 다양한 for 문의 형태 2

```
01 public class Ex06_17 {  
02     public static void main(String[] args) {  
03         int i;  
04         i = 0;  
05         for ( ; ; ) { ----- 초깃값, 조건식, 증감식이 없다.  
06             System.out.printf("%d \n", i);  
07             i++;  
08         }  
09  
10     }  
11 }
```

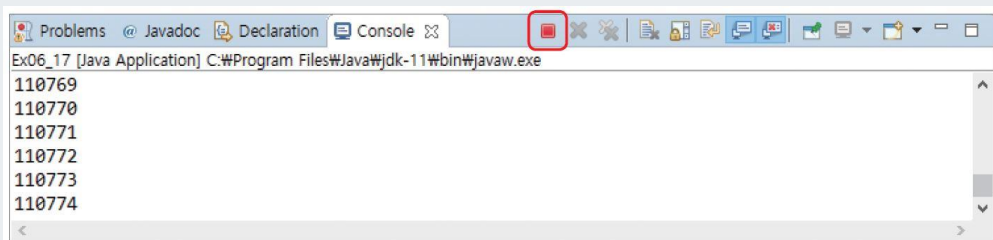


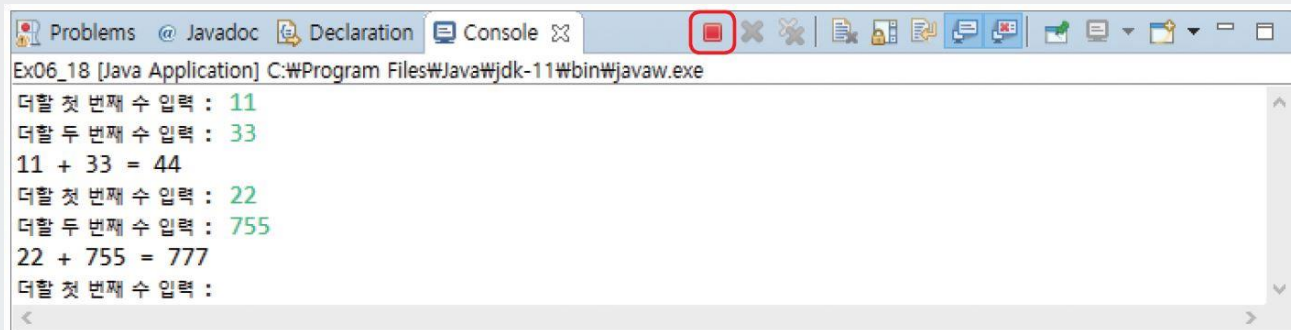
그림 6-27 실행 결과

35. 기타 for 문(5)

실습 6-18 다양한 for 문의 형태 3

```
01 import java.util.Scanner;
02
03 public class Ex06_18 {
04     public static void main(String[] args) {
05         Scanner s = new Scanner(System.in);
06         int a, b;
07
08         { ----- 무한 루프이다.
09             System.out.printf("더할 첫 번째 수 입력 : ");
10             a = s.nextInt();
11             System.out.printf("더할 두 번째 수 입력 : ");
12             b = s.nextInt(); ----- 숫자 2개를 입력받는다.
13
14             System.out.printf("%d + %d = %d \n", a, b, a + b); ----- 덧셈 결과를 출력한다.
15         }
16
17     }
18 }
```

36. 기타 for 문(6)



```
Ex06_18 [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe
더할 첫 번째 수 입력 : 11
더할 두 번째 수 입력 : 33
11 + 33 = 44
더할 첫 번째 수 입력 : 22
더할 두 번째 수 입력 : 755
22 + 755 = 777
더할 첫 번째 수 입력 :
```

그림 6-28 실행 결과

37. while 문(1)

while 문의 비교



그림 7-1 while 문의 실행 순서와 for 문의 실행 순서 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++ ;  
}
```

38. while 문(2)

실습 7-1 for 문을 while 문으로 바꾸기 1

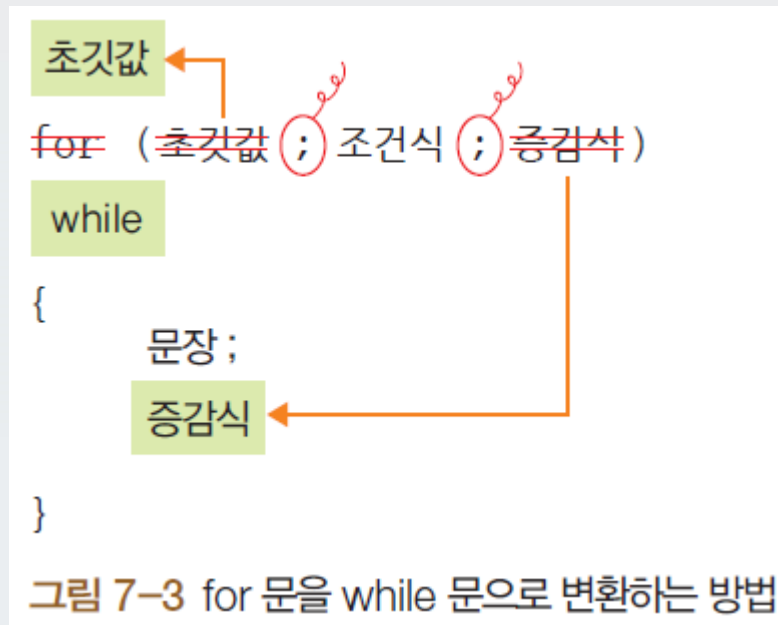
```
01 public class Ex07_01 {  
02     public static void main(String[] args) {  
03         int i;  
04         i = 0; ----- 초기값은 while 문 위로 이동한다.  
05  
06         while (i < 5) { ----- 조건식은 while 문과 함께 놓는다.  
07             System.out.printf("while 문을 공부합니다.\n");  
08             i++; ----- 증감식은 while 문 블록({}) 안으로 이동한다.  
09         }  
10     }  
11 }
```

The screenshot shows a Java IDE window with the 'Console' tab selected. The title bar indicates the application is 'Ex07_01 [Java Application]' running at 'C:\Program Files\Java\jdk-11\bin\javaw.exe'. The console output shows five lines of the text 'while 문을 공부합니다.' (while studying), confirming that the while loop executed five times as intended.

그림 7-2 실행 결과

39. while 문(3)

- [실습 7-1]은 [실습 6-2]의 'for (i=0 ; i<5 ; i++)' 문을 while문으로 바꾼 것임.



40. while 문(4)

실습 7-2 for 문을 while 문으로 바꾸기 2

```
01 public class Ex07_02 {  
02     public static void main(String[] args) {  
03         int hap = 0;  
04         int i;  
05  
06         1 ----- 초깃값을 지정한다.  
07         while (i <= 10) { ----- 조건식이다.  
08             hap = hap + i;  
09             2 ----- 증감식이다.  
10         }  
11  
12         System.out.printf(" 1에서 10까지의 합: %d \n", hap);  
13     }  
14 }
```

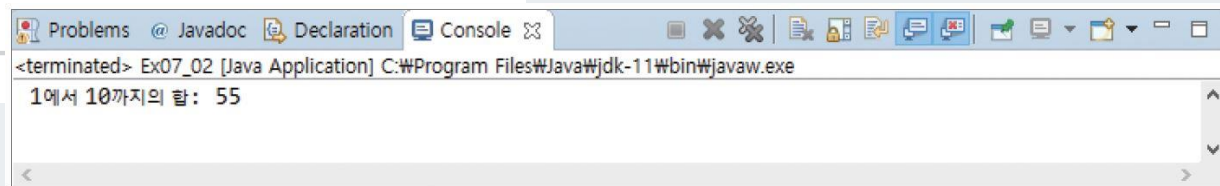
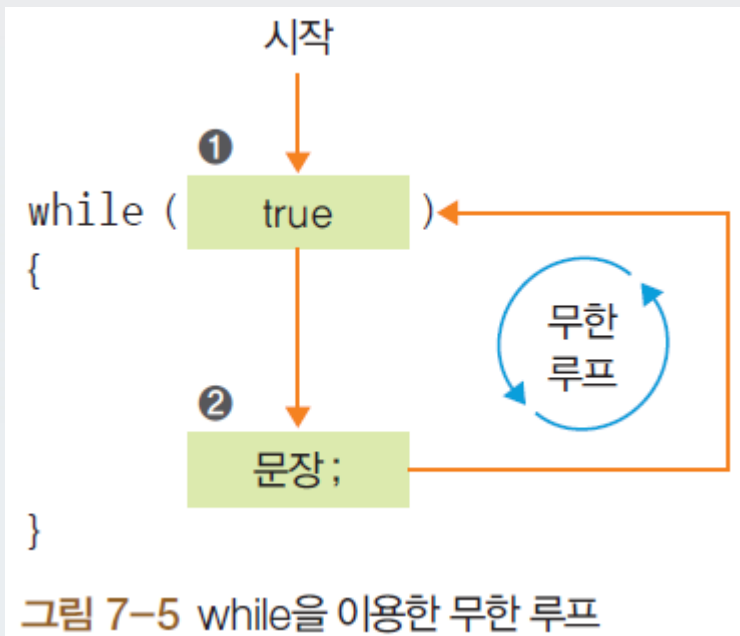


그림 7-4 실행 결과

41. while 문(5)

- 무한 루프를 위한 while 문

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



42. while 문(6)

실습 7-3 while 문의 무한 루프 만들기

```
01 import java.util.Scanner;
02
03 public class Ex07_03 {
04     public static void main(String[] args) {
05         Scanner s = new Scanner(System.in);
06         int a, b;
07
08         while (true) {
09             System.out.printf("더할 첫 번째 수 입력 : ");
10             a = s.nextInt();
11             System.out.printf("더할 두 번째 수 입력 : ");
12             b = s.nextInt();
13
14             System.out.printf("%d + %d = %d \n", a, b, a + b);
15         }
16
17     }
18 }
```

----- 값을 입력한다.

----- 값을 입력한다.

----- 결과를 출력한다.

무한 루프 이다.



```
Problems @ Javadoc Declaration
Ex07_03 [Java Application] C:\Program File
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 :
```

그림 7-6 실행 결과

43. while 문(7)

실습 7-4 무한 루프를 활용한 계산기

```
01 import java.io.IOException;
02 import java.util.Scanner;
03
04 public class Ex07_04 {
05     public static void main(String[] args) throws IOException {
06         Scanner s = new Scanner(System.in);
07         int a, b;
08         char ch;
09
10         {
11             System.out.printf("계산할 첫번째 수를 입력 : ");
12             a = s.nextInt();
13             System.out.printf("계산할 두번째 수를 입력 : ");
14             b = s.nextInt();
15             System.out.printf("계산할 연산자를 입력하세요 : ");
16             ch = (char) System.in.read();
```

throws IOException이 추가된 것은 16행의 System.in.read()를 사용하기 위한 필수 사항이다.

연산할 숫자를 입력받는다.

연산할 숫자를 입력받는다.

연산자를 문자형으로 입력받는다.

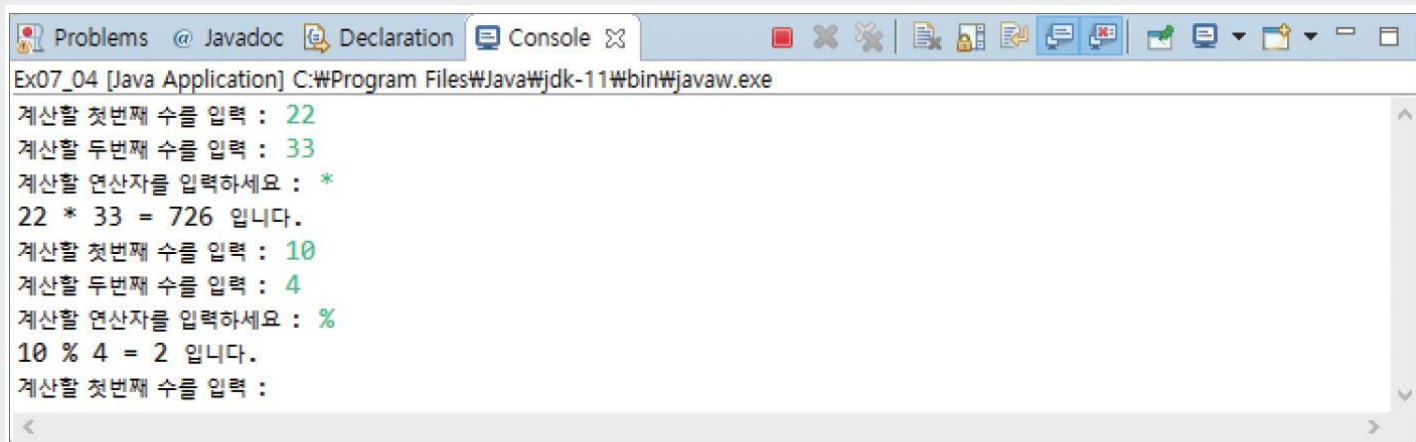
44. while 문(8)

```
17
18     (ch) {
19         case '+':
20             System.out.printf("%d + %d = %d 입니다. \n", a, b, a + b); break;
21         case '-':
22             System.out.printf("%d - %d = %d 입니다. \n", a, b, a - b); break;
23         case '*':
24             System.out.printf("%d * %d = %d 입니다. \n", a, b, a * b); break;
25         case '/':
26             System.out.printf("%d / %d = %f 입니다. \n", a, b, a / (float) b); break;
27         case '%':
28             System.out.printf("%d %% %d = %d 입니다. \n", a, b, a % b); break;
29         default:
30             System.out.printf("연산자를 잘못 입력했습니다. \n");
31     }
32 }
33
34 }
35 }
```

무한
루프
이다.

입력받은 ch 연산자에 의해 +, -, *, / , %로 분기한다. 그 외의 경우 오류 메시지를 출력한다.

45. while 문(9)



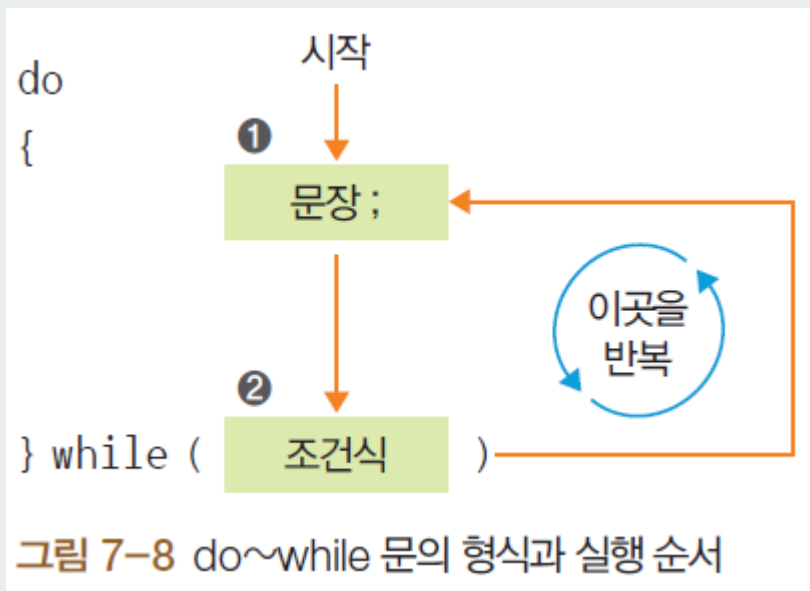
```
Ex07_04 [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe
계산할 첫번째 수를 입력 : 22
계산할 두번째 수를 입력 : 33
계산할 연산자를 입력하세요 : *
22 * 33 = 726 입니다.
계산할 첫번째 수를 입력 : 10
계산할 두번째 수를 입력 : 4
계산할 연산자를 입력하세요 : %
10 % 4 = 2 입니다.
계산할 첫번째 수를 입력 :
```

그림 7-7 실행 결과

46. do~while 문(1)

▪ do~while 문

- while 문이나 for 문은 조건식이 처음부터 거짓이면 한 번도 수행하지 않고 종료. 하지만 do~while 문은 어떠한 경우라도 한 번은 수행함



47. do~while 문(2)

실습 7-5 do~while 문 사용 예 1

```
01 public class Ex07_05 {
02     public static void main(String[] args) {
03         int a = 100;
04
05         while (a == 200) {
06             System.out.printf("while 문 내부에 들어 왔습니다.\n");
07         }
08
09         do {
10             System.out.printf("do ~ while 문 내부에 들어 왔습니다.\n");
11         } while (a == 200);
12     }
13 }
```

while 문 실행 : 먼저
조건식을 판단한다.

do~while 문 실행
먼저 내용을 실행한
다음 조건식을
판단한다.

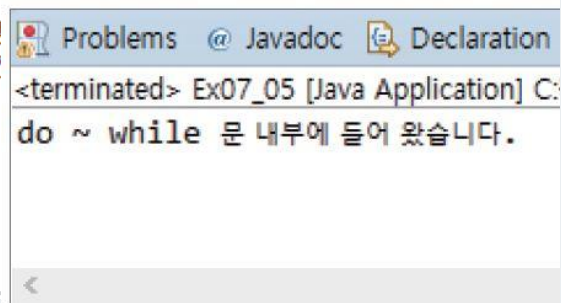


그림 7-9 실행 결과

48. do~while 문(3)

실습 7-6 do~while 문 사용 예 2

```
01 import java.util.Scanner;
02
03 public class Ex07_06 {
04     public static void main(String[] args) {
05         Scanner s = new Scanner(System.in);
06         int menu;
07
08         { ----- do~while 문이므로 반드시 한 번은 수행한다.
09             System.out.printf("₩n손님 주문하시겠습니까 ? ₩n");
10             System.out.printf("<1>카페라떼 <2>카푸치노 <3>아메리카노 <4>그만시킴래요 => ");
11             menu = s.nextInt(); ----- 커피 종류를 선택한다.
```

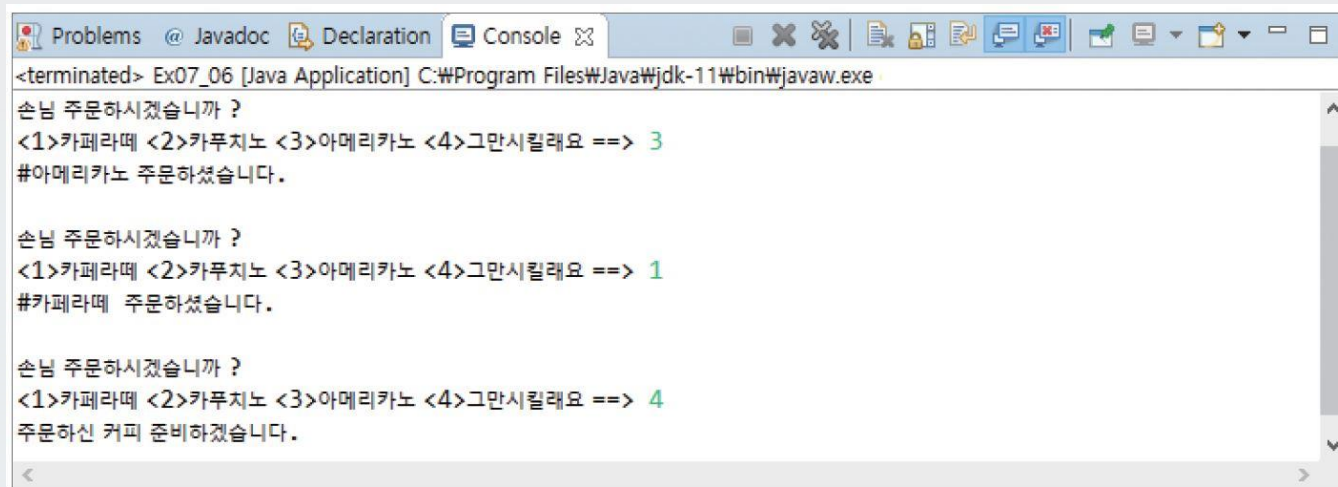
49. do~while 문(4)

```
12      switch (menu) {
13          case 1:
14              System.out.printf("#카페라떼 주문하셨습니다.\n"); break;
15          case 2:
16              System.out.printf("#카푸치노 주문하셨습니다.\n"); break;
17          case 3:
18              System.out.printf("#아메리카노 주문하셨습니다.\n"); break;
19          case 4:
20              System.out.printf("주문하신 커피 준비하겠습니다.\n"); break;
21          default:
22              System.out.printf("잘못 주문하셨습니다.\n");
23      }
24  }  (menu != 4); ----- 선택한 메뉴가 4번이 아니라면 계속 반복해서 주문을 받는다.
25  }
26 }
```

선택한 커피에
따라 주문을
접수한다.

while ☒ do ☒ ☒

50. do~while 문(5)



```
<terminated> Ex07_06 [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe
손님 주문하시겠습니까 ?
<1>카페라떼 <2>카푸치노 <3>아메리카노 <4>그만시킬래요 ==> 3
#아메리카노 주문하셨습니다.

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

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

그림 7-10 실행 결과

51. 기타 제어문(1)

반복문을 탈출하는 break 문

- 지금까지 배운 반복문을 빠져나가는 방법은 조건식의 결과가 거짓이거나 사용자가 직접 [Terminate] 버튼을 누를 때임. 이외에도 반복문을 논리적으로 빠져나가는 방법인 break 문이 있음

반복문(for, while, do~while)

{

...

break;

...

}



무조건 반복문
블록 밖으로 탈출

그림 7-11 break 문의 작동

52. 기타 제어문(2)

실습 7-7 break 문 사용 예 1

```
01 public class Ex07_07 {
02     public static void main(String[] args) {
03         int i;
04
05         for (i = 1; i <= 100; i++) { ----- 100번 반복한다.
06             System.out.printf("for 문을 %d 회 실행했습니다.\n", i); ----- 변수 i번째를 출력한다.
07             break; ----- 무조건 for 문을 빠져나간다.
08         }
09
10         System.out.printf("for 문을 종료했습니다.\n");
11     }
12 }
```

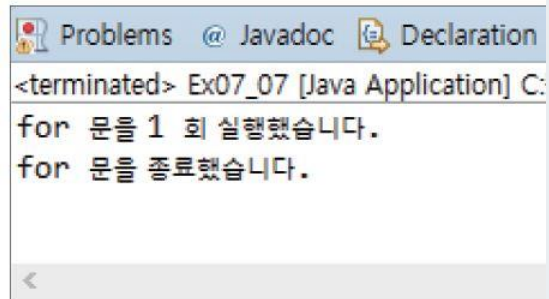


그림 7-12 실행 결과

53. 기타 제어문(3)

실습 7-8 break 문 사용 예 2

```
01 import java.util.Scanner;
02
03 public class Ex07_08 {
04     public static void main(String[] args) {
05         Scanner s = new Scanner(System.in);
06         int a, b;
07
08         while (true) {
09             System.out.printf("더할 첫 번째 수 입력 : ");
10             a = s.nextInt();
11             System.out.printf("더할 두 번째 수 입력 : ");
12             b = s.nextInt();
13
14             if (a == 0)
15                 break;
16
17             System.out.printf("%d + %d = %d Wn", a, b, a + b);
18         }
19
20         System.out.printf("@을 입력해서 반복문을 탈출했습니다.Wn");
21     }
22 }
```

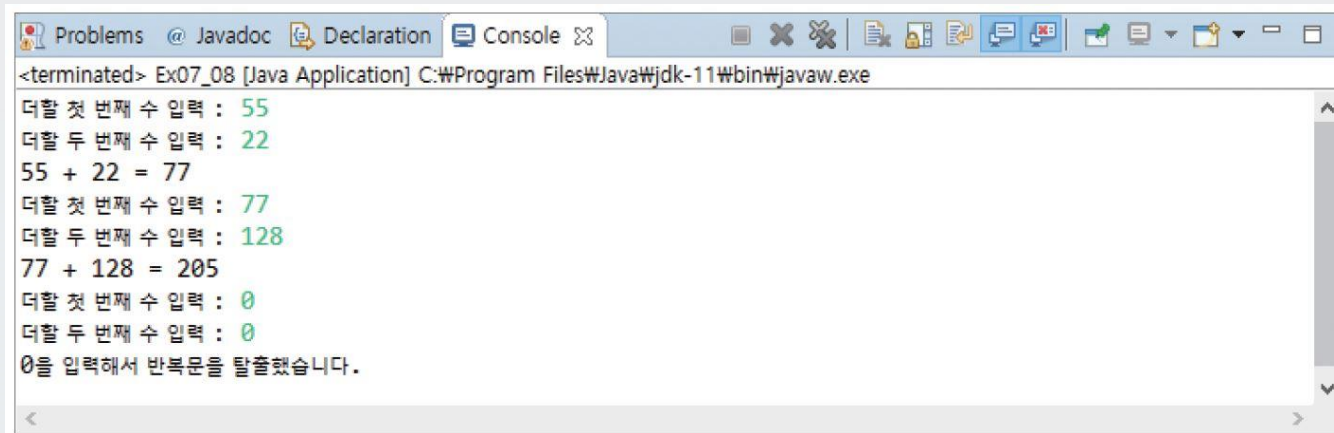
숫자를 입력받는다.

숫자를 입력받는다.

첫 번째 입력 값이 0이면 무조건 while 문을 빠져나간다.

무한 루프이다.

54. 기타 제어문(4)



```
<terminated> Ex07_08 [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe
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 : 0
더할 두 번째 수 입력 : 0
0을 입력해서 반복문을 탈출했습니다.
```

그림 7-13 실행 결과

55. 기타 제어문(5)

실습 7-9 break 문 사용 예 3

```
01 public class Ex07_09 {
02     public static void main(String[] args) {
03         int hap = 0;
04         int i;
05
06         for (i = 1; i <= 100; i++) {
07             hap = hap + i;
08
09             if (  )
10                 break;
11         }
12
13         System.out.printf(" 1~100의 합에서 최초로 1000이 넘는 위치는? : %d\n", i);
14     }
15 }
```

i 값을 hap에 누적한다.

hap이 1000보다 크거나 같으면
for 반복문을 빠져나간다.

i 값을 1부터 100까지
100회 실행한다.

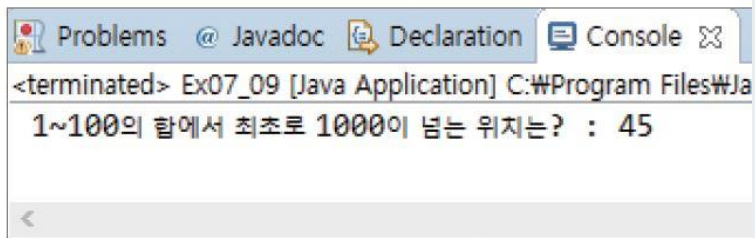
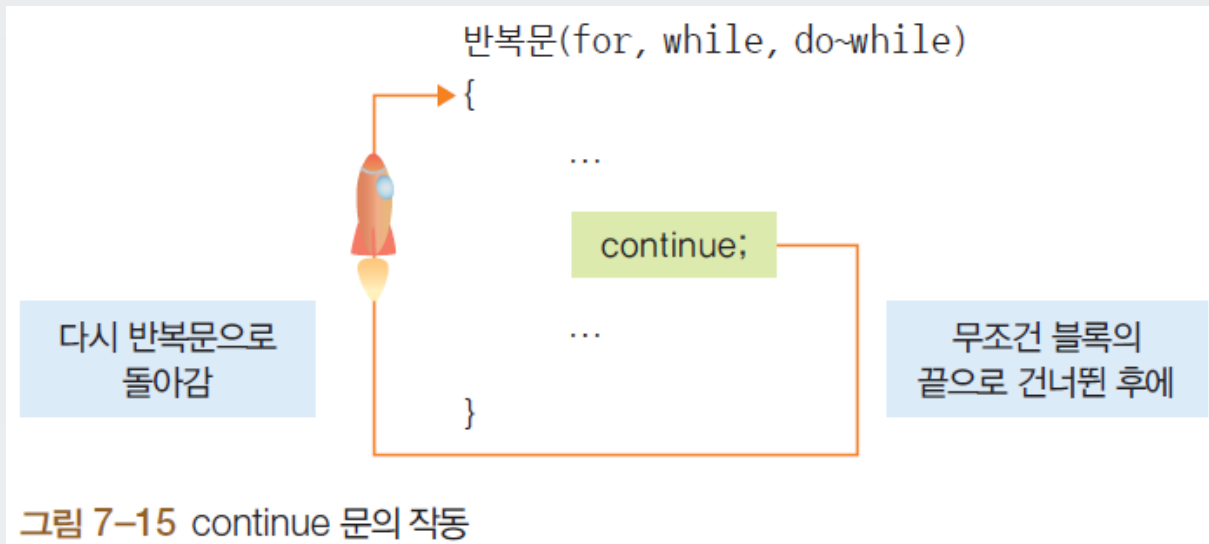


그림 7-14 실행 결과

56. 기타 제어문(6)

- 반복문으로 다시 돌아가는 continue 문
 - continue 문을 만나면 블록의 남은 부분을 건너뛰고 반복문의 처음으로 돌아감



57. 기타 제어문(7)

실습 7-10 continue 문 사용 예

```
01 public class Ex07_10 {
02     public static void main(String[] args) {
03         int hap = 0;
04         int i;
05
06         for (i = 1; i <= 100; i++) {
07             if (i % 3 == 0)
08                 continue;
09
10             hap += i;
11         }
12
13         System.out.printf(" 1~100까지의 합(3의 배수 제외): %d\n", hap);
14     }
15 }
```

1부터 100까지
100회 반복한다.

i 값을 3으로 나눈 나머지 값이 0이면
(=3의 배수이면) 블록의 끝으로 건너뛰고
다시 6행으로 돌아간다.

3의 배수가 아닌 i 값을 누적한다.

누적된 값을 출력한다.



그림 7-16 실행 결과

58. 기타 제어문(8)

- 7행 : $i \% 3 == 0$ 은 i 를 3으로 나눈 나머지 값이 0일 때 참이라는 의미(즉 3의 배수)
-
- 제1회 : i 값 1을 3으로 나누면 나머지는 1(거짓) → $hap += 1$ 수행
- 제2회 : i 값 2를 3으로 나누면 나머지는 2(거짓) → $hap += 2$ 수행
- 제3회 : i 값 3을 3으로 나누면 나머지는 0(참) → continue 문 수행
- 맨 끝(11행)으로 건너뛰고 다시 6행으로 돌아가서 증감식 수행
- 제4회 : i 값 4를 3으로 나누면 나머지는 1(거짓) → $hap += 4$ 수행
- 제5회 : i 값 5를 3으로 나누면 나머지는 2(거짓) → $hap += 5$ 수행
- 제6회 : i 값 6을 3으로 나누면 나머지는 0(참) → continue 문 수행
- 맨 끝(11행)으로 건너뛰고 다시 6행으로 돌아가서 증감식 수행
- ⋮
- 이렇게 계속 진행하면 $hap=1+2+4+5+7+\dots$ 과 같은 계산식이 됨

59. 기타 제어문(9)

다중 반복문의 지정된 위치로 이동하는 break 레이블문

실습 7-11 다중 반복문의 무한 루프

```
01 public class Ex07_11 {  
02     public static void main(String[] args) {  
03         int hap = 0;  
04         int i;  
05  
06         for (;;) {  
07             for (i = 1; i <= 100; i++) {  
08                 hap += i;  
09                 if (hap > 2000) {  
10                     System.out.printf("%d\n", hap);  
11                     hap = 0;  
12                     break;  
13                 }  
14             }  
15             System.out.printf("아직도 반복중...\n");  
16         }  
17     }  
18 }
```

합계를 누적한다.

누적된 값이 2000을 넘으면 hap을 출력하고 초기화한 다음 12행의 break로 반복문을 빠져나간다.

100회 반복한다.

무한 반복한다.

무한 반복을 확인하기 위한 출력이다.

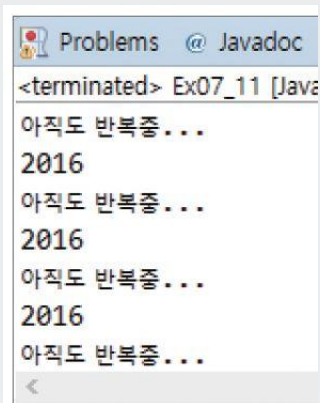


그림 7-17 실행 결과

60. 기타 제어문(10)

- 이 코드는 무한 반복. 2000이 넘는 값을 만나면 12행에서 break를 만나 반복문을 빠져나가려고 시도. 하지만 가장 가까운 for 문(7~14행)의 끝인 14행을 빠져나가서 15행을 출력하고, 다시 6~16행의 무한 루프 for 문을 만나 7행부터 다시 시작. 결국 무한 반복.
- break 문을 별도로 지정한 레이블(label)과 함께 사용. 'break 레이블이름'과 같이 지정해줌

레이블 : 반복문(for, while, do~while)

```
{  
    반복문(for, while, do~while)  
    {  
        break 레이블;  
    }  
}
```

레이블이 지정된
반복문 블록 밖으로
빠져나감

그림 7-18 break 레이블문의 작동

61. 기타 제어문(11)

실습 7-12 break 레이블문 사용 예

```
01 public class Ex07_12 {
02     public static void main(String[] args) {
03         int hap = 0;
04         int i;
05
06         myLabel: for (;;) { ----- 바깥 for 문에 레이블을 지정한다.
07             for (i = 1; i <= 100; i++) {
08                 hap += i;
09                 if (hap > 2000) {
10                     System.out.printf("%d\n", hap);
11                     hap = 0;
12                      ----- 지정된 myLabel의 반복문을 빠져나간다.
13                 }
14             }
15             System.out.printf("아직도 반복중...\n");
16         }
17     }
18 }
```

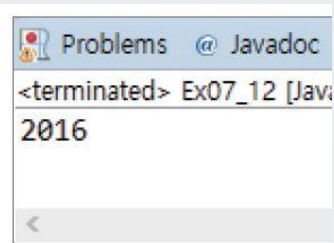
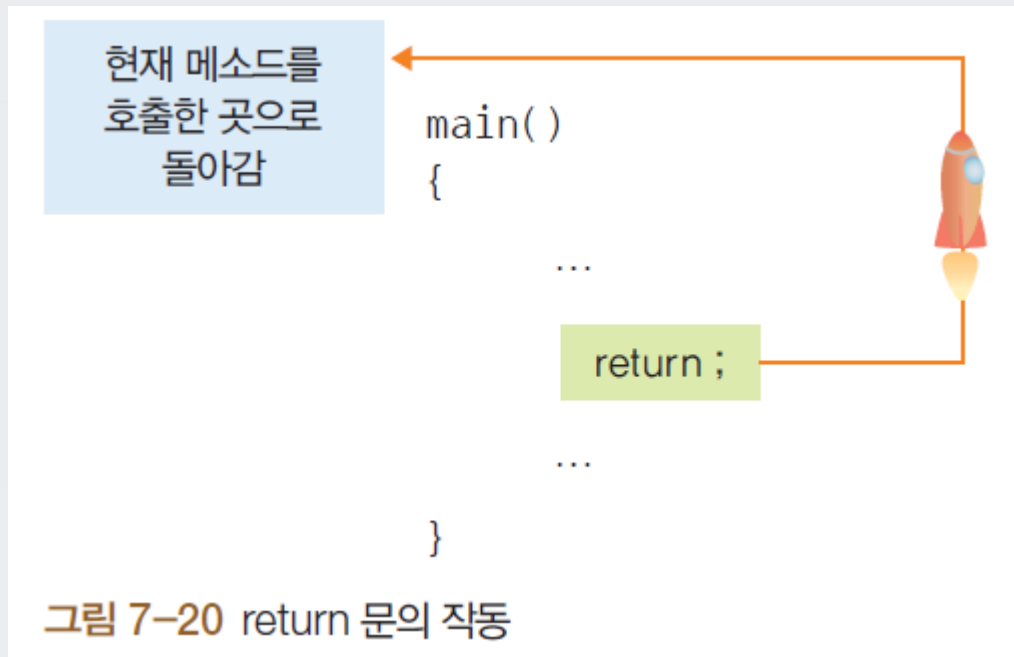


그림 7-19 실행 결과

break myLabel;

62. 기타 제어문(12)

- 현재 메소드를 불렀던 곳으로 돌아가는 return 문



63. 기타 제어문(13)

실습 7-13 return 문 사용 예

```
01 public class Ex07_13 {
02     public static void main(String[] args) {
03         int hap = 0;
04         int i;
05
06         for (i = 1; i <= 100; i++) {
07             hap += i;
08
09             System.out.printf("1부터 100까지의 합은 %d 입니다.\n", hap);
10
11             if (hap > 5000)
12                 return;
13
14             System.out.printf("프로그램의 끝입니다.");
15         }
16     }
```

1부터 100까지 합계를 누적한다.

합계를 출력한다.

현재 메소드를 호출한 곳으로 복귀한다.

한 번도 실행되지 않는다.



그림 7-21 실행 결과

다음 시간

소프트웨어공학 정리 2

