

컴퓨터 프로그래밍 (Computer Programming)

이 선 순



8. 반복문의 기본 : for문



목차

- 1. 단순 for문
- 2. 중첩 for문
- 3. 기타 for문

01

단순 for문

01 단순 for문

■ 반복문

- 어떤 작업(코드)이 반복적으로 실행되도록 할 때 사용
- 반복문의 종류 : for문, while문, do-while문
- for문과 while문은 서로 변환이 가능
- for문은 반복횟수를 알고 있을 때 주로 사용, while문은 조건에 따라 반복할 때 주로 사용함
- while문과 do-while문은 조건을 먼저 검사하느냐, 나중에 검사하느냐일 뿐 동작 방식은 동일함

01 단순 for문

■ for문의 개념

- 문장을 반복하게 만들어주는 것

```
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. ^^
```

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. ^Wn");  
04         System.out.printf("안녕하세요? 빙글빙글 for 문을 공부중입니다. ^Wn");  
05         System.out.printf("안녕하세요? 빙글빙글 for 문을 공부중입니다. ^Wn");  
06         System.out.printf("안녕하세요? 빙글빙글 for 문을 공부중입니다. ^Wn");  
07         System.out.printf("안녕하세요? 빙글빙글 for 문을 공부중입니다. ^Wn");  
08     }  
09 }
```

내용을
출력한다.

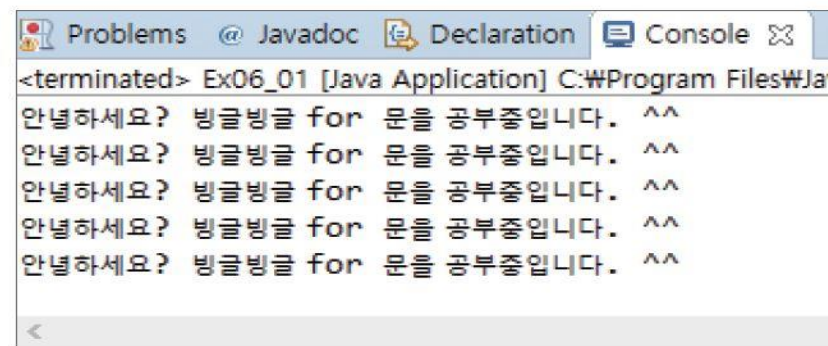


그림 6-1 실행 결과

Tip. 입출력 메소드와 증감 연산자의 복습

- 앞에서 공부했던 내용을 떠올려보자.

① 출력 메소드 : `System.out.printf()`

`System.out.printf("안녕?")` → '안녕?' 출력

`System.out.printf("%d", 10)` → '10' 출력

② 입력 메소드 : `nextInt()`

`Scanner s = new Scanner(System.in)` → 클래스 준비

`num = s.nextInt()` → 사용자가 키보드로 숫자를 입력

③ 증감 연산자(++ , --)

`a++`는 `a`에 1을 증가시키라는 의미로 `a = a+1`과 동일하다.

또한 `a--`는 `a`에서 1을 감소시키라는 의미로 `a = a-1`과 동일하다

01 단순 for문

■ for문의 개념

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.

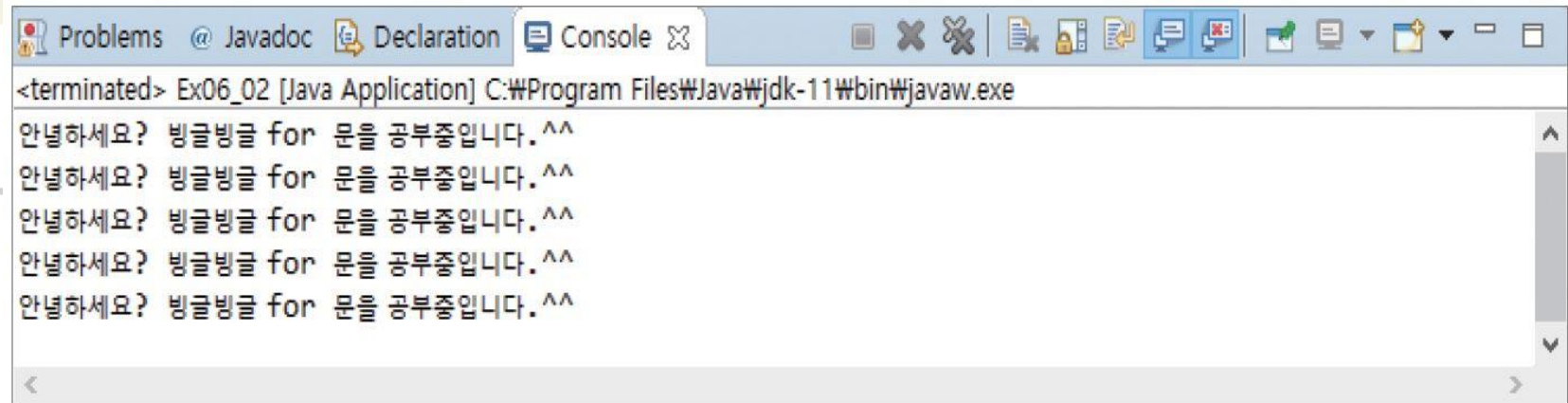


그림 6-2 실행 결과

01 단순 for문

■ for문의 활용

```
for (i = 0; i < 5; i++)  
{  
    System.out.printf("안녕하세요? 빙글빙글 for 문을 공부중입니다.^Wn");  
}
```

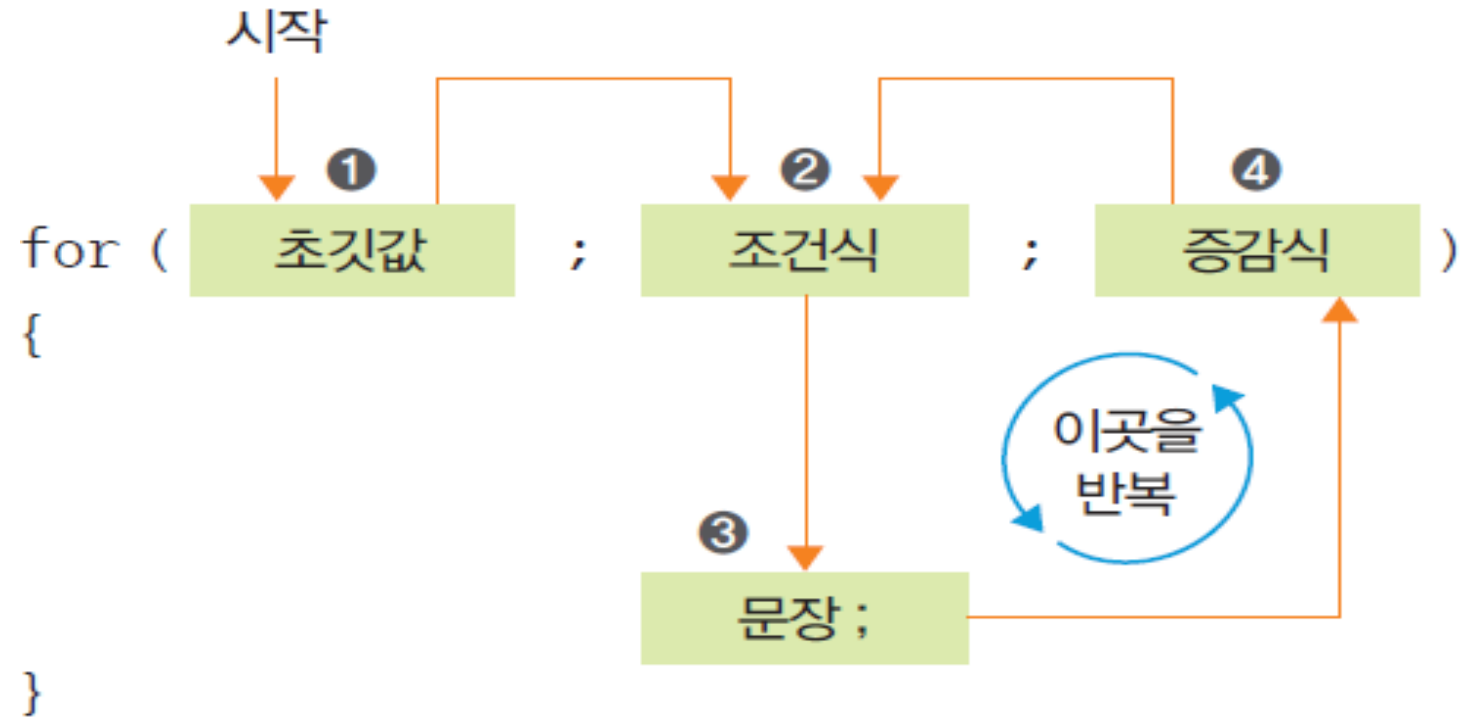


그림 6-3 for 문의 실행 순서

01 단순 for문

■ for문의 활용

- [실습 6-2]의 5~8행에서 사용한 for 문의 기본 구조

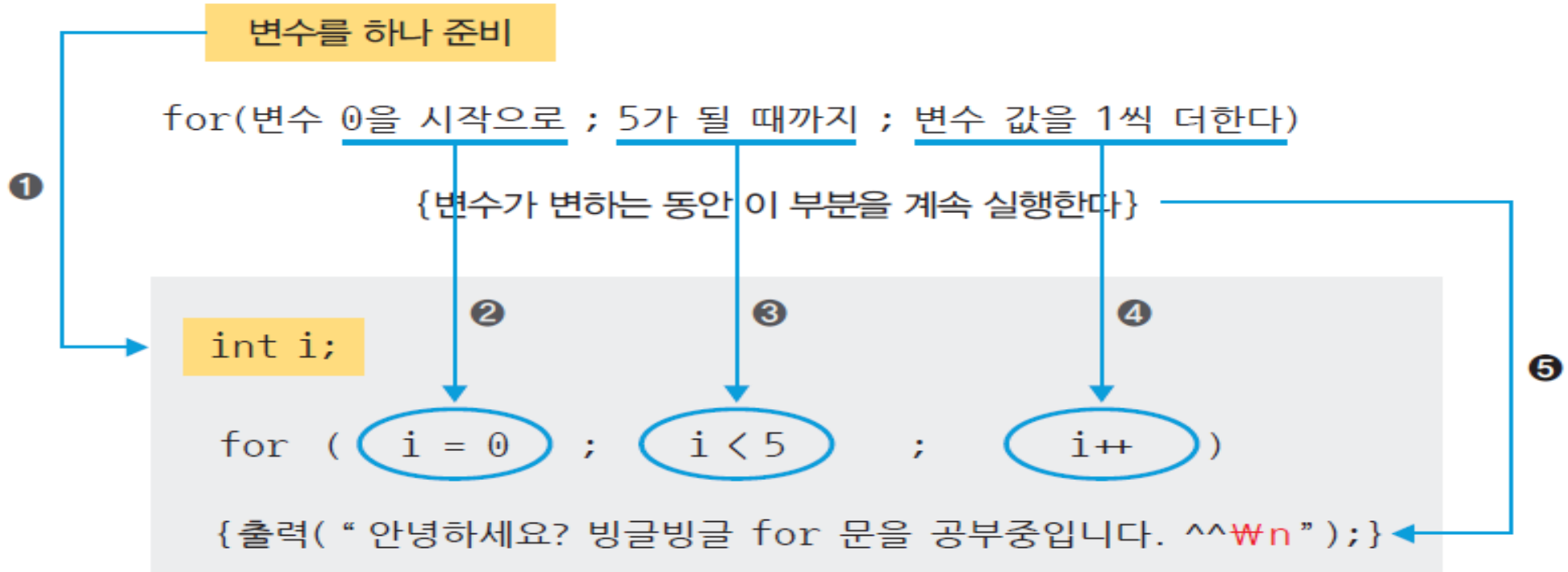


그림 6-4 for 문의 개념과 실제 사용

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

01 단순 for문

■ for문의 활용

- [실습 6-2]의 5~8행에서 사용한 for 문의 기본 구조

① 초기값 수행

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

② 조건식 확인

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

조건이
거짓이면

⑤ 반복문 탈출

조건이 참이면

01 단순 for문

■ for문의 활용

- [실습 6-2]의 5~8행에서 사용한 for 문의 기본 구조

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

01 단순 for문

■ for문의 활용

- [실습 6-2]의 5~8행에서 사용한 for 문의 기본 구조

제1회 : ❶ 초기값을 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.

01 단순 for문

■ for문의 활용

- 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. ^ Wn");  
}
```

==

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

01 단순 for문

■ for문의 활용

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? \n");  
07             System.out.printf("##또 안녕하세요?## \n");  
08         }  
09  
10         System.out.printf("\n\n");  
11  
12         for (i = 0; i < 3; i++)  
13             System.out.printf("안녕하세요? \n");  
14             System.out.printf("##또 안녕하세요?## \n");  
15  
16     }  
17 }
```

for 문에 중괄호를 사용했다.

for 문에 중괄호를 사용하지 않았다.

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

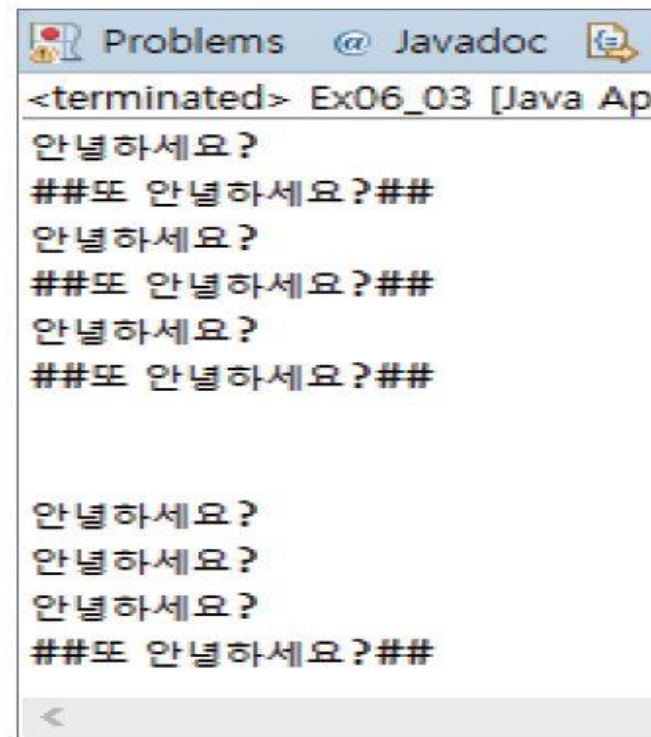
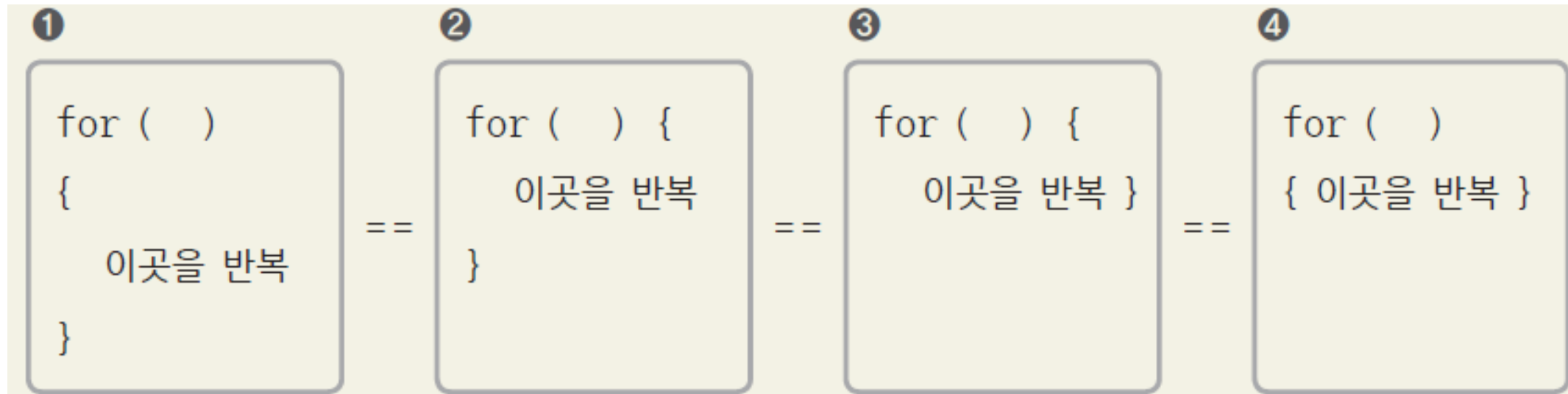


그림 6-6 실행 결과

01 단순 for문

■ for문의 개념

- 중괄호의 위치는 어디든 관계없다. 다음의 네 가지 경우는 모두 동일하다



이클립스에서 코딩한 다음 **Ctrl** + **Shift** + **F**를 누르면 소스코드가 ②와 같은 형태로 자동 정렬된다.

01 단순 for문

■ for문 연습

- 중괄호의 위치는 어디든 관계없다. 다음의 네 가지 경우는 모두 동일하다

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 }
```

--! : 0<! : 5=! ~!~

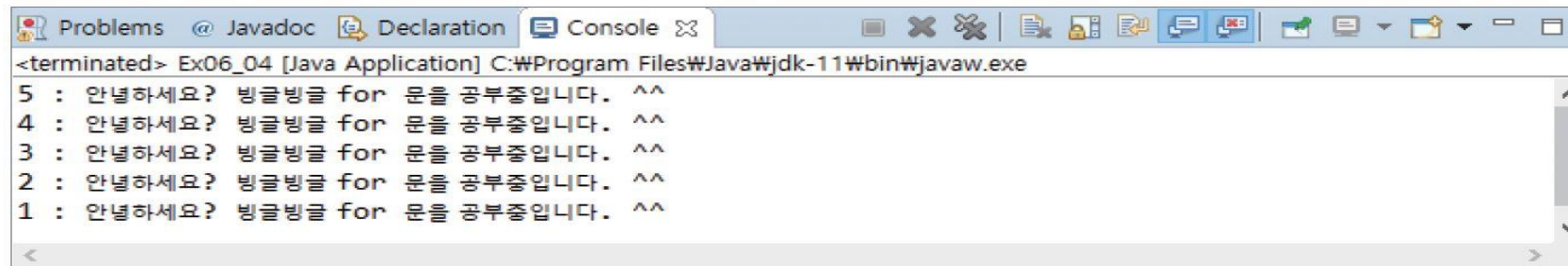


그림 6-7 실행 결과

01 단순 for문

■ for문 연습

- [실습6-2]를 수정하여 1부터 5까지 차례대로 출력: for문안에서 i값을 활용하여 i값을 출력

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 }
```

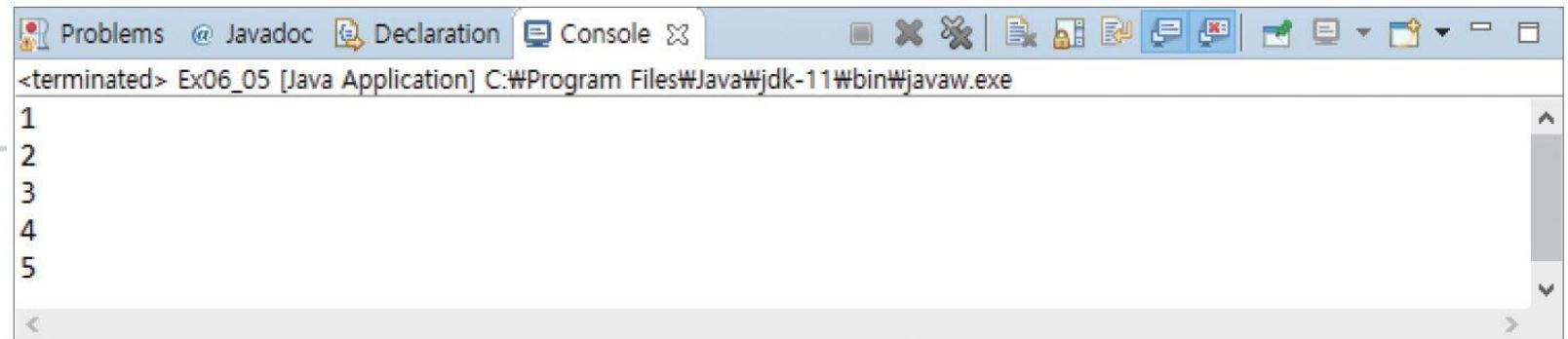
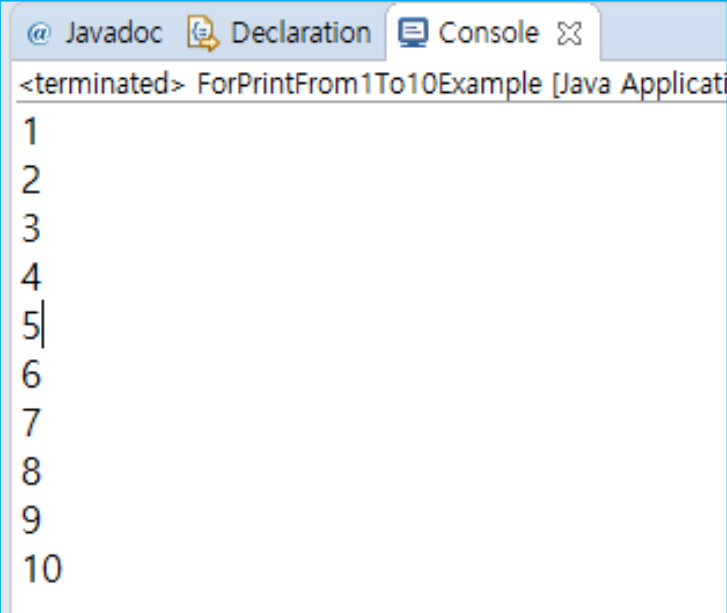


그림 6-8 실행 결과

01 단순 for문

■ for문 연습

```
1 // 예1. 1부터 10까지 출력
2 package sec02.exam01;
3
4 public class ForPrintFrom1To10Example {
5     public static void main(String[] args) {
6
7         for(int i=1; i<=10; i++) {
8
9             System.out.println(i);
10        }
11    }
12 }
13
```



The screenshot shows a Java IDE window with tabs for Javadoc, Declaration, and Console. The Console tab is active, displaying the output of the program: numbers 1 through 10, each on a new line. The text in the console is: <terminated> ForPrintFrom1To10Example [Java Applicati... followed by the numbers 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

01 단순 for문

■ 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 }
```

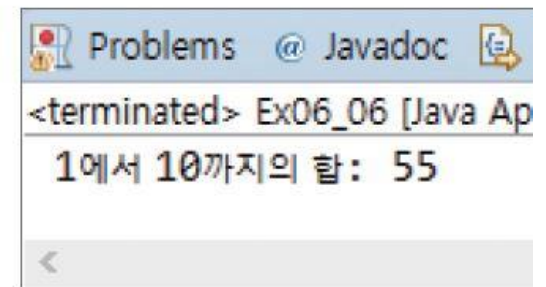


그림 6-9 실행 결과

- 1~10이 아니라 1~100까지 합계를 구해야 한다면? for문을 이용한 반복문 활용

01 단순 for문

■ for 문을 활용하여 합계 구하기

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
}
```

01 단순 for문

■ for 문을 활용하여 합계 구하기

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 }
```

3행 : **hap**의 값이 초기화 되지 않음.
7행의 계산결과가 잘못된 값이 계산됨
hap은 누적값을 계산하므로 반드시 **0**으로 초기화해야 함

6행의 **i**는 1로 초기화함

for 문에 의해 1부터
10까지 10회 반복된다.

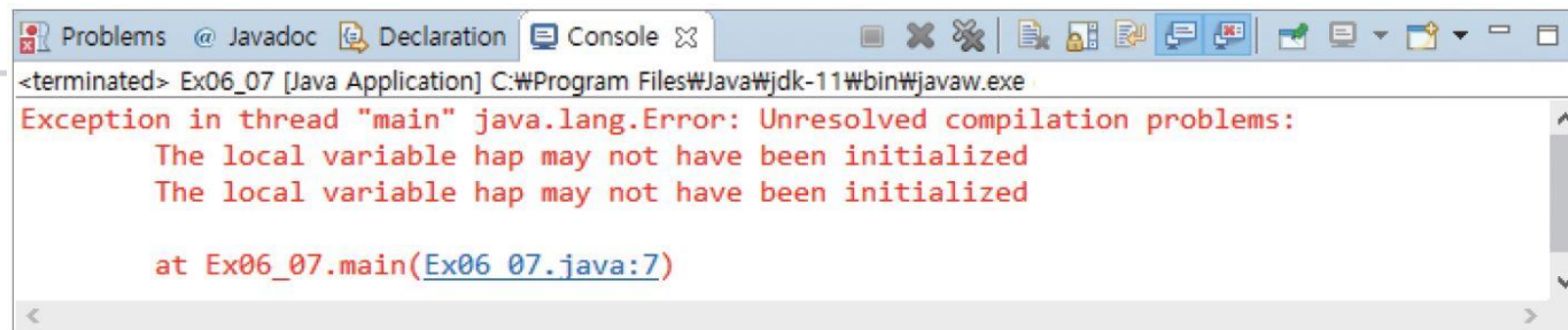


그림 6-10 실행 결과

01 단순 for문

■ for 문을 활용하여 합계 구하기

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 }
```

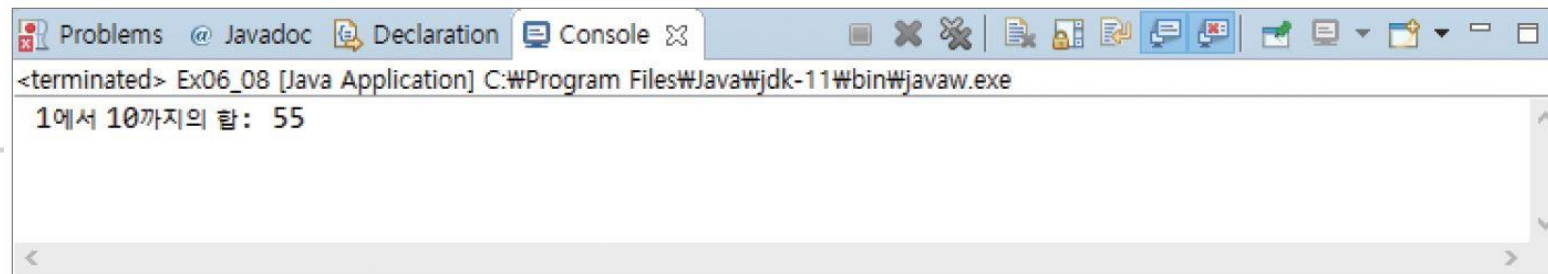


그림 6-11 실행 결과

01 단순 for문

■ for 문을 활용하여 합계 구하기

- [실습 6-8]의 변수 i와 hap의 변화

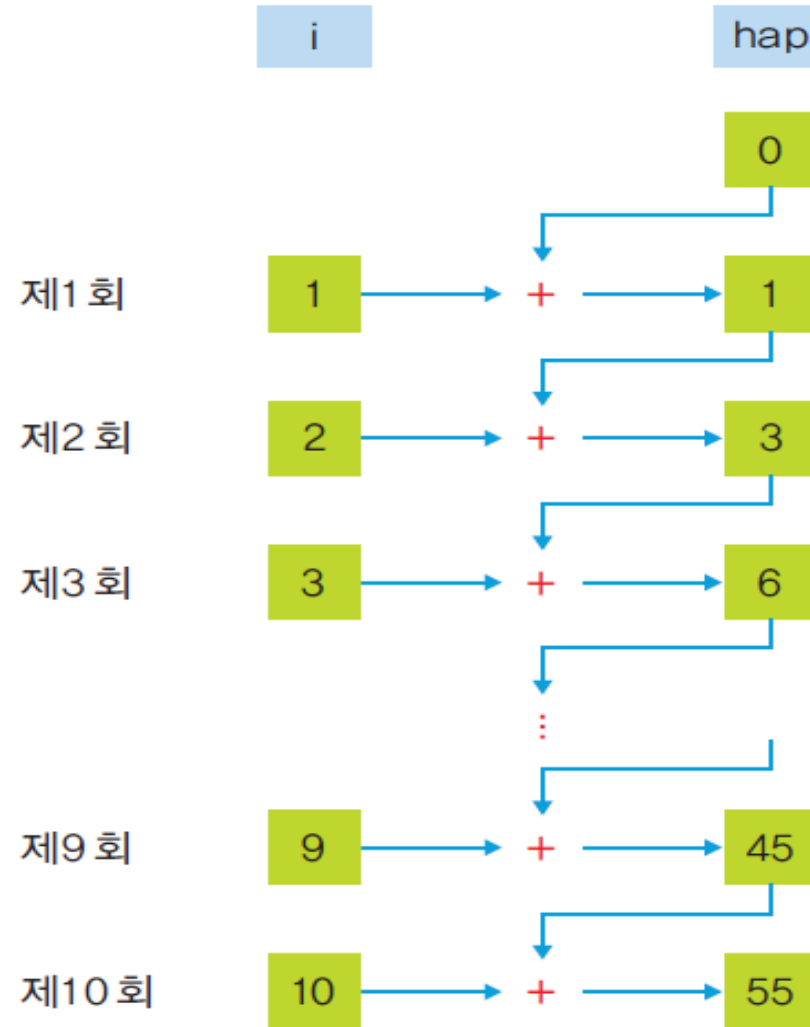


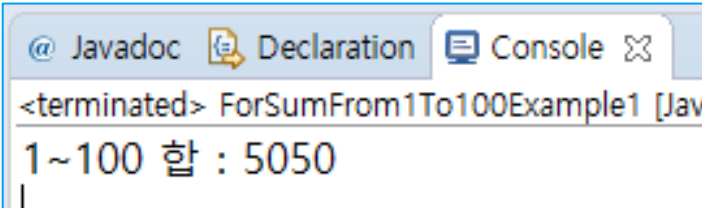
그림 6-12 변수 i와 hap의 변화

01 단순 for문

■ for 문을 활용하여 합계 구하기

- for 문이 시작하기 전에 합계 변수 sum을 선언한 이유는 for문을 끝내고 9라인에서 sum을 사용하기 때문임

```
1 //예2. 1부터 100까지 합을 출력
2
3 package sec02.exam02;
4
5 public class ForSumFrom1To100Example1 {
6     public static void main(String[] args) {
7         int sum = 0;
8
9         for(int i=1; i<=100; i++) {
10             sum += i;
11         }
12
13         System.out.println("1~100 합 : " + sum);
14     }
15 }
16
```



The screenshot shows an IDE console window with tabs for Javadoc, Declaration, and Console. The Console tab is active, displaying the output of the program: "<terminated> ForSumFrom1To100Example1 [Java] 1~100 합 : 5050".

01 단순 for문

■ for 문을 활용하여 합계 구하기

- [실습6-9]. 500부터 1000사이 홀수의 합을 구하는 프로그램 : for문의 초깃값, 조건식, 증감식을 적절히 변경하여 다양한 형태의 합계를 구할 수 있음

```
1 public class Ex06_09 {  
2     public static void main(String[] args) {  
3         int hap = 0;  
4         int i;  
5  
6         for (i = 501; i <= 1000; i += 2) {  
7             hap += i;  
8         }  
9  
10        System.out.printf(" 500에서 1000까지의 홀수의 합 : %d \n", hap);  
11    }  
12 }
```

Problems @ Javadoc Declaration Console

<terminated> Ex06_09 [Java Application] C:\Program Files\Java\jdk-15.0.2\bin\javaw.exe

500에서 1000까지의 홀수의 합 : 187500

01 단순 for문

■ for 문을 활용하여 합계 구하기

- [실습6-10]. 실행 시 숫자를 입력하게 한 뒤 입력한 수 까지의 합계를 구하는 프로그램 : Scanner 클래스 사용

```
1 import java.util.Scanner;
2
3 public class Ex06_10 {
4     public static void main(String[] args) {
5         Scanner s = new Scanner(System.in);
6         int hap = 0;        //합계를 누적할 변수를 선언하고 0으로 초기화
7         int i;              //1씩 증가할 변수 선언
8         int num;            //입력받을 끝값 선언
9
10        System.out.printf("값 입력 : ");
11        num = s.nextInt();    //끝값 입력
12
13        for (i = 1; i <= num; i++) { //1부터 끝값까지 1씩 증가하며 반복
14            hap = hap + i;
15        }
16
17        System.out.printf(" 1에서 %d까지의 합 : %d \n", num, hap);
18    }
19 }
```

- 사용자가 입력한 값을 저장할 변수 **num**을 선언,
- **nextInt()** 메소드를 사용하여 사용자가 입력한 숫자를 **num**에 대입
- 13행 : 1부터 사용자가 입력한 숫자(**num**)까지 1씩 증가하면서 **for**문 반복
- 사용자가 입력한 숫자까지의 합계를 구하여 19행에서 사용자가 입력한 숫자와 함께 출력

```
<terminated> Ex06_10 [Java Application] C:\Program Files\Java\jdk-15.0.2\bin\javaw.exe
값 입력 : 100
1에서 100까지의 합 : 5050
```

01 단순 for문

■ for 문을 활용하여 합계 구하기

- [실습6-11]. 시작값, 최종값, 증가값까지 입력받아 합계를 구하는 프로그램

```
1 import java.util.Scanner;
2
3 public class Ex06_11 {
4     public static void main(String[] args) {
5         Scanner s = new Scanner(System.in);
6         int hap = 0;
7         int i;
8         int num1, num2, num3; //입력받을 세변수 선언
9
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 (i = num1; i <= num2; i = i + num3) {
18            hap = hap + i;
19        }
20
21        System.out.printf(" %d에서 %d까지 %d씩 증가한 값의 합 : %d \n", num1, num2, num3, hap);
22    }
23 }
```

Problems @ Javadoc Declaration Console

<terminated> Ex06_11 [Java Application] C:\Program Files\Java\jdk-15.0.2\bin\javaw.exe

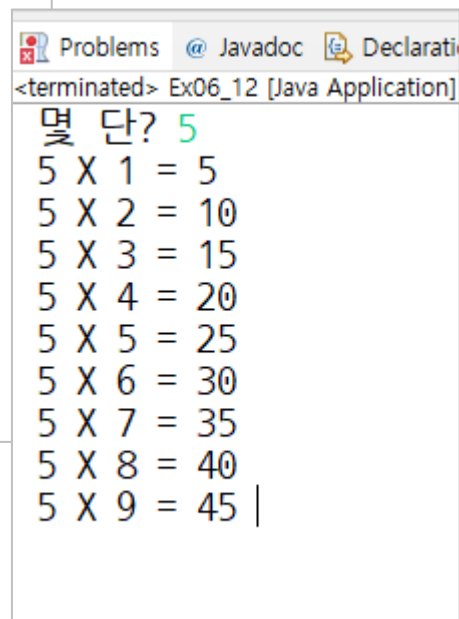
시작값 입력 : 1
끝값 입력 : 500
증가값 입력 : 3
1에서 500까지 3씩 증가한 값의 합 : 41750

01 단순 for문

■ for 문을 활용하여 합계 구하기

- [실습6-12]. For문을 활용하여 사용자가 입력한 숫자의 구구단을 출력하는 프로그램

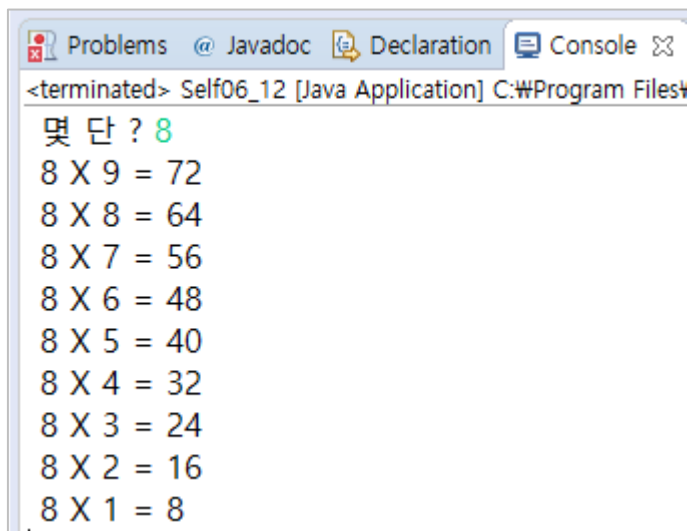
```
1 import java.util.Scanner;
2
3 public class Ex06_12 {
4     public static void main(String[] args) {
5         Scanner s = new Scanner(System.in);
6         int i;
7         int dan; //계산할 단을 입력받을 변수 선언
8
9         System.out.printf(" 몇 단? ");
10        dan = s.nextInt(); //계산할 단을 선언
11
12        for (i = 1; i <= 9; i++) { //입력한 단에 대한 구구단을 1~9까지 반복해서 출력
13            System.out.printf(" %d X %d = %d \n", dan, i, dan * i);
14        }
15    }
16 }
```



```
Problems @ Javadoc Declarati
<terminated> Ex06_12 [Java Application]
몇 단? 5
5 X 1 = 5
5 X 2 = 10
5 X 3 = 15
5 X 4 = 20
5 X 5 = 25
5 X 6 = 30
5 X 7 = 35
5 X 8 = 40
5 X 9 = 45 |
```

Self study 8-1

1. [실습6-2]를 수정해서 '안녕하세요.? 빙글빙글 for문을 공부중입니다.^.^'를 20번 출력하되, 줄바꿈이 되지 않고 문장이 이어서 출력되도록 수정해보자.
2. [실습6-5]를 10부터 20까지 출력되도록 수정해보자.
3. [실습6-8]을 수정하여 1부터 12345까지의 합계를 구해보자.
4. [실습6-9]를 수정하여 1~100중에서 3의배수의 합계를 구하는 프로그램을 작성해보자.
(Hint : 3의 배수이므로 3부터 시작하고 증가값은 3씩 더하면 된다.)
5. [실습6-12]를 수정하여 다음과 같이 구구단이 거꾸로 출력되게 해보자.



```
<terminated> Self06_12 [Java Application] C:\Program Files\
몇 단 ? 8
8 X 9 = 72
8 X 8 = 64
8 X 7 = 56
8 X 6 = 48
8 X 5 = 40
8 X 4 = 32
8 X 3 = 24
8 X 2 = 16
8 X 1 = 8
```

질문은 이메일을 이용해주세요.
ds.june2@gmail.com

감사합니다