

컴퓨터 프로그래밍 (Computer Programming)

이 선 순



8. 반복문의 기본 : for문



목차

1. 단순 for문
2. 중첩 for문
3. 기타 for문

02

중첩 for문

02 중첩 for문

■ 중첩 for 문의 개념

- for 문 내부에 또 다른 for 문이 들어 있는 형태
- 바깥쪽 for문이 한번 실행될 때마다 중첩된 for문은 지정된 횟수만큼 반복해서 돌다가 다시 바깥쪽 for문으로 돌아감

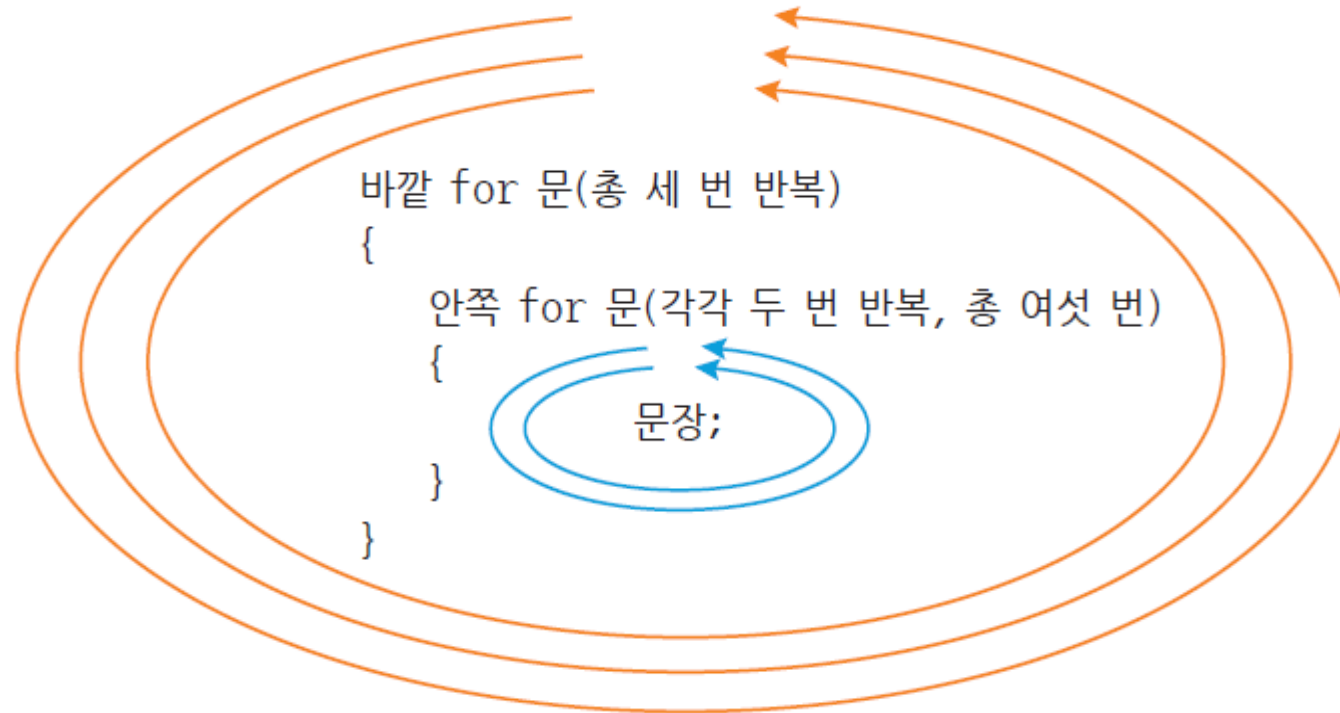


그림 6-17 중첩 for 문의 동작 개념

02 중첩 for문

■ 중첩 for 문의 개념

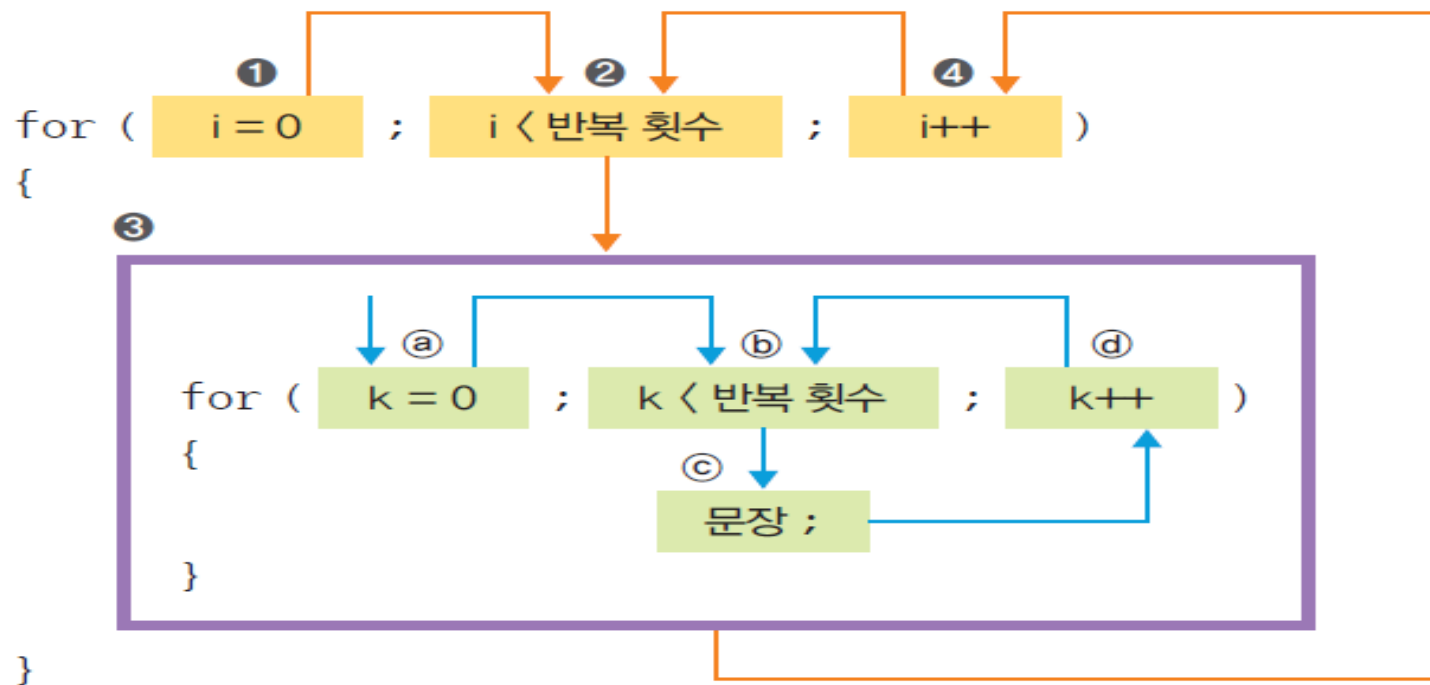


그림 6-18 중첩 for 문의 작동 방식

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

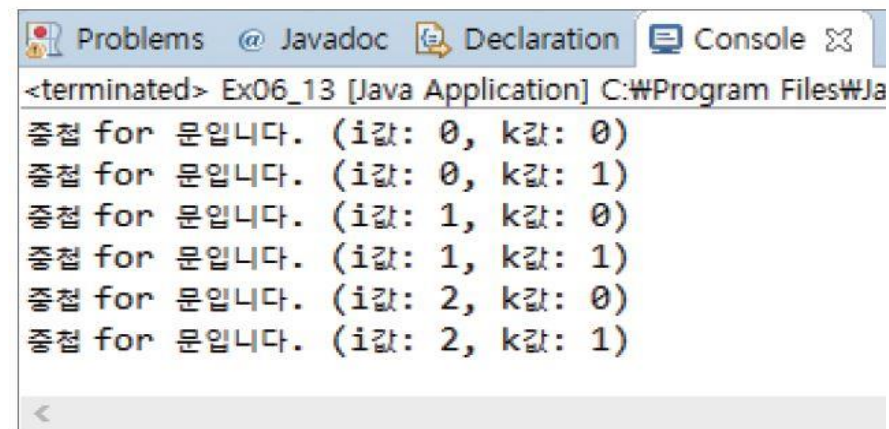
02 중첩 for문

■ 중첩 for 문의 개념

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);
10             }
11         }
12
13     }
14 }
```

i와 k 값을 총 여섯 번($=3 \times 2$) 출력한다.



```
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

02 중첩 for문

Tip. 프로그래머에 따라 다른 코딩

- [실습 6-13]의 5~11행을 코딩할 때 프로그래머의 취향에 따라 다음과 같이 코딩할 수도 있다. 그러면 줄의 수가 줄어들어 프로그램이 간단해 보인다.

```
for (i=0 ; i<3 ; i++) {  
    for (k=0 ; k<2 ; k++)    {  
        System.out.printf("중첩 for 문입니다. (i값: %d, k값: %d)\n", i, k);  
    }  
}
```

- 또한 실행문이 하나뿐이므로 다음과 같이 중괄호를 없애도 된다. 하지만 실행할 문장이 하나이더라도 중괄호를 사용하면 실수를 방지하고 코드의 가독성을 높일 수 있다.

```
for (i=0 ; i<3 ; i++)  
    for (k=0 ; k<2 ; k++)  
        System.out.printf("중첩 for 문입니다. (i값: %d, k값: %d)\n", i, k);
```


02 중첩 for문

■ [실습 6-13]의 처리 순서

①

외부 for 문 1회 : 5행의 i를 0으로 초기화한 후 'i<3'가 참이므로 바깥 for 문 수행
내부 for 문 1회 : 7행의 k를 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 k를 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 k를 2로 증가시킨 후 'k<2'가 거짓이므로 안쪽 for 문 종료

```
1 public class Ex06_13 {  
2     public static void main(String[] args) {  
3         int i, k;    // 반복할 변수 i, k 선언  
4  
5         for (i = 0; i < 3; i++) // 바깥 for 문 3회 반복  
6         {  
7             for (k = 0; k < 2; k++) //안쪽 for문 2회 반복  
8             {  
9                 System.out.printf("중첩 for 문입니다. (i값: %d, k값: %d)\n", i, k);  
10            }  
11        }  
12    }  
13 }  
14 }  
15 }
```

②

외부 for 문 2회 : 5행의 i++로 i를 1로 증가시킨 후 'i<3'가 참이므로 바깥 for 문 수행
내부 for 문 1회 : 7행의 k를 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 k를 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 k를 2로 증가시킨 후 'k<2'가 거짓이므로 안쪽 for 문 종료

02 중첩 for문

■ 실습[6-13]의 중첩 for 문에서 i와 k 값의 변화

3

외부 for 문 3회 : 5행의 i++로 i를 2로 증가시킨 후 'i<3'가 참이므로 바깥 for 문 수행
내부 for 문 1회 : 7행의 k를 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 k를 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 k를 2로 증가시킨 후 'k<2'가 거짓이므로 안쪽 for 문 종료

4

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

```
1 public class Ex06_13 {  
2     public static void main(String[] args) {  
3         int i, k;    // 반복할 변수 i, k 선언  
4  
5         for (i = 0; i < 3; i++) // 바깥 for 문 3회 반복  
6         {  
7             for (k = 0; k < 2; k++) //안쪽 for문 2회 반복  
8             {  
9                 System.out.printf("중첩 for 문입니다. (i값: %d, k값: %d)\n", i, k);  
10            }  
11        }  
12    }  
13 }  
14 }  
15 }
```

02 중첩 for문

- 실습[6-13]의 중첩 for 문에서 i와 k 값의 변화

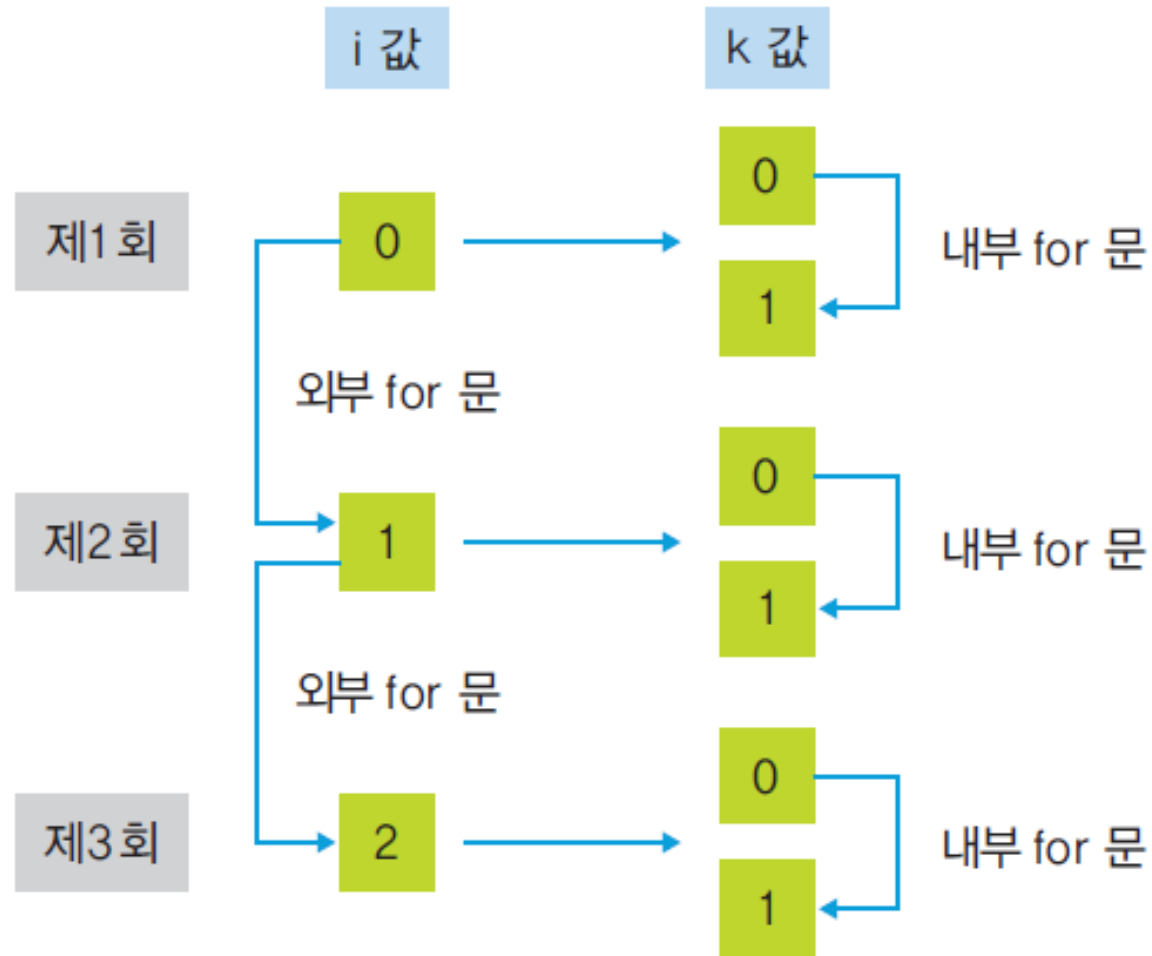


그림 6-20 중첩 for 문에서 i와 k 값의 변화

02 중첩 for문

■ 중첩 for 문의 활용

■ 구구단 2단~9단

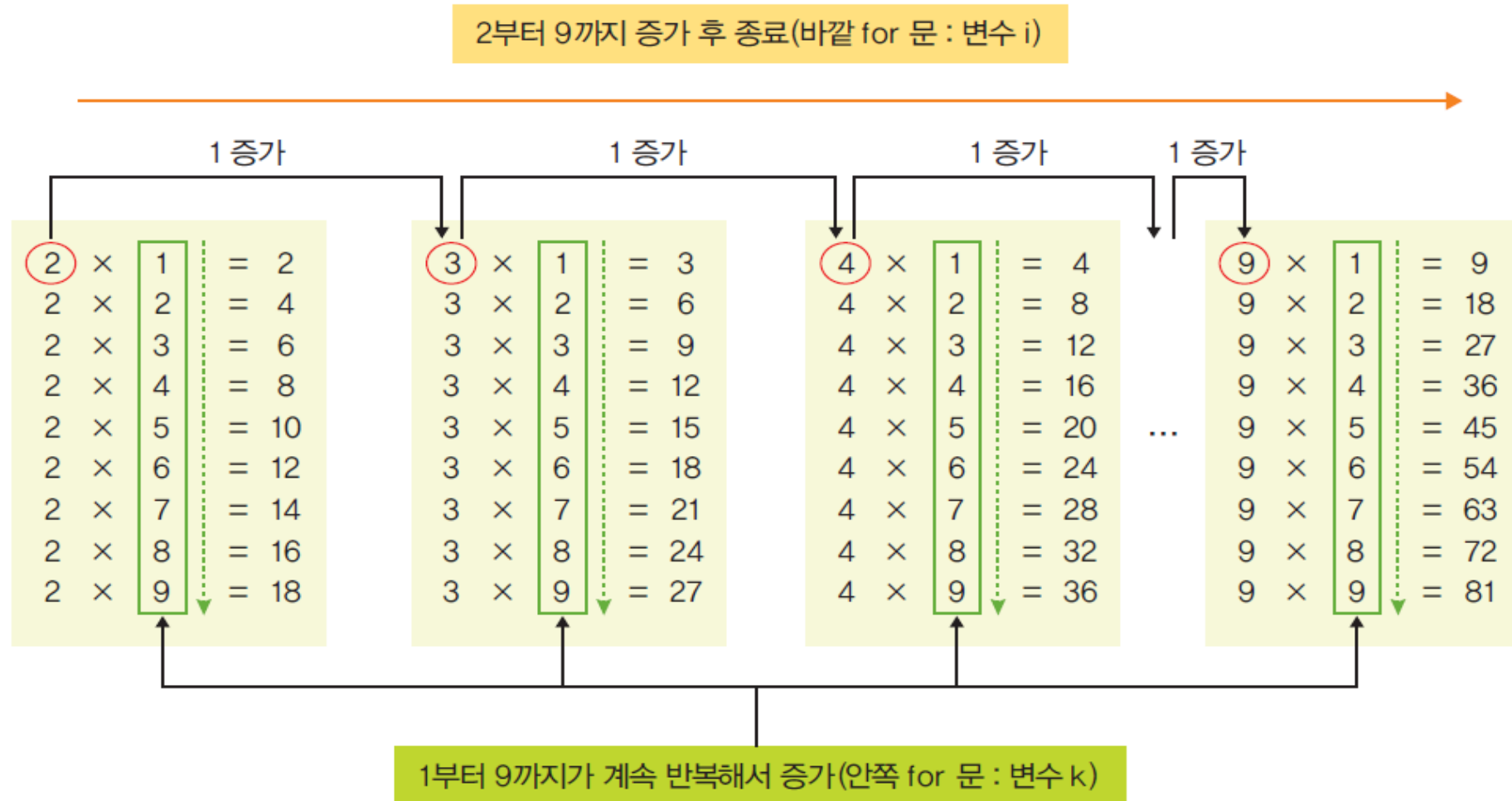


그림 6-21 구구단에서 변수 i와 k의 추출

02 중첩 for문

■ 중첩 for 문의 활용

- [실습6-14] 구구단 2단~9단 만들기

```
1 public class Ex06_14 {  
2     public static void main(String[] args) {  
3         int i, k;  
4  
5         for (i = 2; i <= 9; i++) { //2~9단을 반복  
6  
7             for (k = 1; k <= 9; k++) { //각 단의 뒷자리 숫자 1~9를 반복  
8                 System.out.printf(" %d X %d = %d \n", i, k, i * k); //구구단 출력  
9             }  
10            System.out.printf("\n"); //각 단이 끝나면 한 줄 띄움  
11        }  
12    }  
13 }  
14 }
```

Problems @ Javadoc Declaration
<terminated> Ex06_14 [Java Application] C:\WPro

```
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  
3 X 3 = 9  
3 X 4 = 12  
3 X 5 = 15  
3 X 6 = 18  
3 X 7 = 21  
3 X 8 = 24  
3 X 9 = 27  
  
4 X 1 = 4  
4 X 2 = 8  
4 X 3 = 12  
4 X 4 = 16  
4 X 5 = 20  
4 X 6 = 24
```

02 중첩 for문

■ 중첩 for 문의 활용

- 구구단에서 변수 i와 k의 추출(가로를 먼저 출력하는 경우)

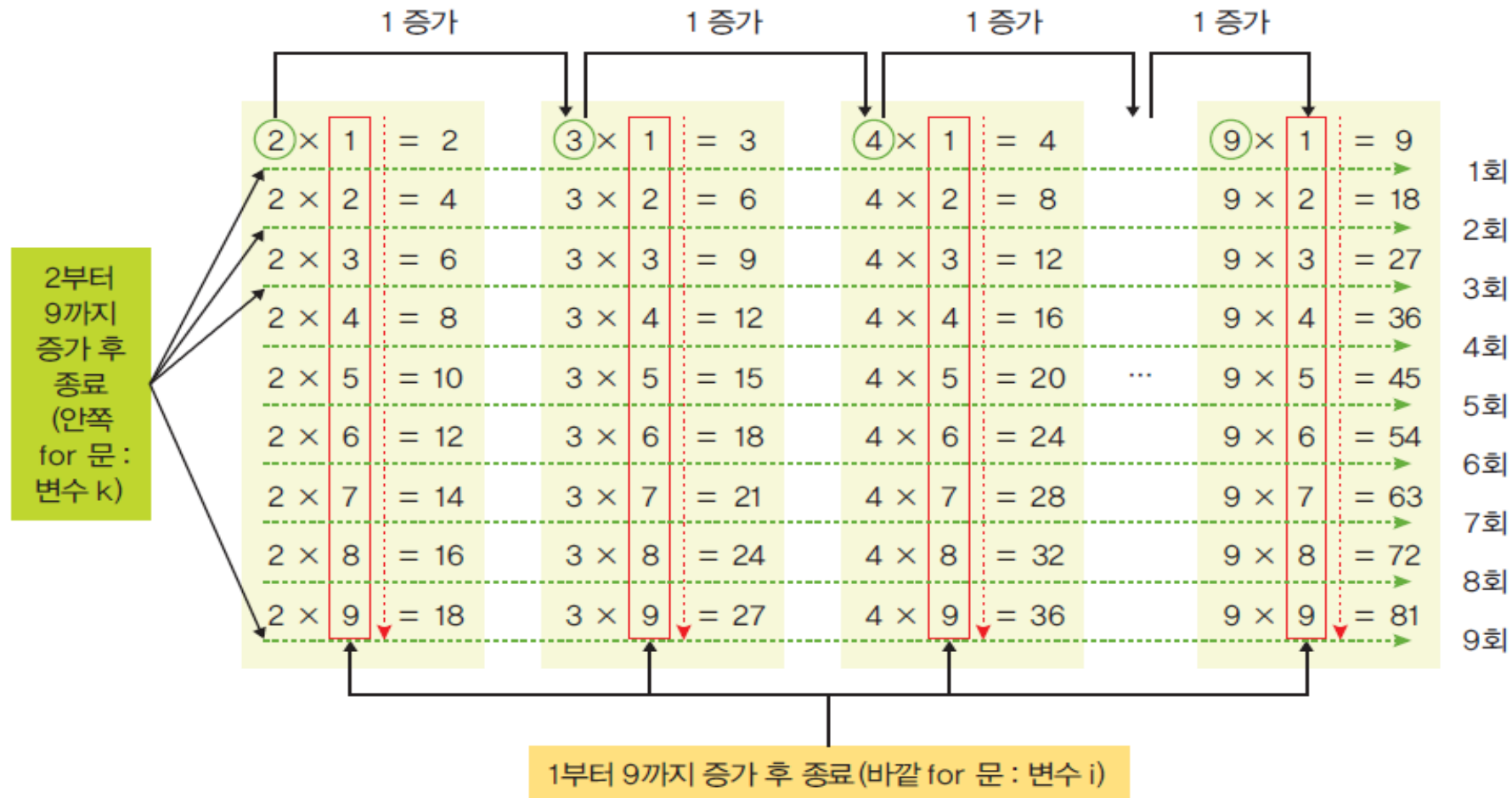


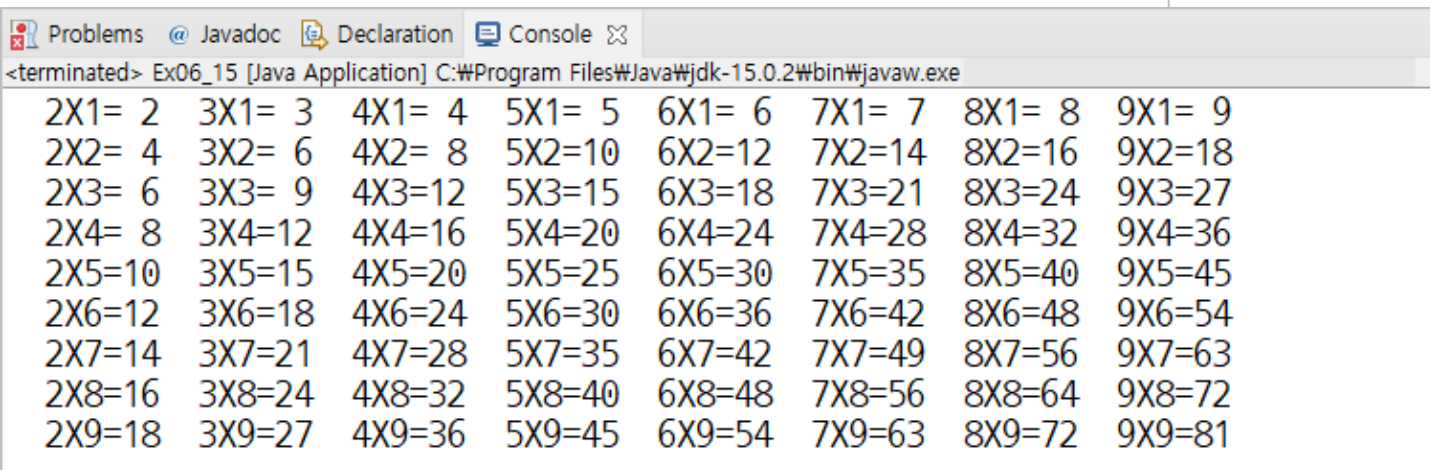
그림 6-23 구구단에서 변수 i와 k의 추출(가로를 먼저 출력하는 경우)

02 중첩 for문

■ 중첩 for 문의 활용

- [실습6-15] 구구단에서 변수 i와 k의 추출(가로를 먼저 출력하는 경우)

```
1 public class Ex06_15 {  
2     public static void main(String[] args) {  
3         int i, k;  
4  
5         for (i = 1; i <= 9; i++) { //각 단의 뒷자리 숫자 1~9를 반복  
6             for (k = 2; k <= 9; k++) { //2~9단을 반복  
7  
8                 System.out.printf("%3dX%d=%2d", k, i, k * i); //각 단별로 한 줄씩 출력  
9             }  
10            System.out.printf("\n"); //각 단의 한 줄을 출력한 후 다음 줄로 넘긴다.  
11        }  
12    }  
13 }  
14 }
```



```
<terminated> Ex06_15 [Java Application] C:\Program Files\Java\jdk-15.0.2\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
```

03

기타 for문

03 기타 for문

■ 여러 개의 초깃값과 증감식을 사용하는 for 문

- for문에 들어가는 초깃값이 꼭 하나일 필요는 없음
- 여러 개를 초기화할 때는 쉼표(,)로 구분해야 함

```
for (초깃값 1, 초깃값 2; 조건식; 증감식 1, 증감식 2)
```

```
for ( int i=0, j=100; i<50 && j>=50 ; i++ , j--) { ... }
```

03 기타 for문

■ 여러 개의 초깃값과 증감식을 사용하는 for 문

■ [실습6-16] 다양한 for문의 형태1

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 Declaration Console

<terminated> Ex06_16 [Java Application] C:\Program Files\Ja

```
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
```

03 기타 for문

■ 초깃값이 없는 for 문

- 초깃값은 조건식과 실행문, 증감식에서 사용할 변수를 초기화하는 역할을 함
- 초깃값이 필요없을 경우에는 다음과 같이 초깃값을 생략할 수 있음

```
int i=1;  
for ( ; i<=100; i++) { ... }
```

- 초깃값에 선언된 변수는 for문 블록 내부에서 사용되는 로컬 변수로 for문을 벗어나서는 사용할 수 없음

03 기타 for문

■ 초깃값과 증감식이 없는 for 문

- 예. 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) ;  
}
```

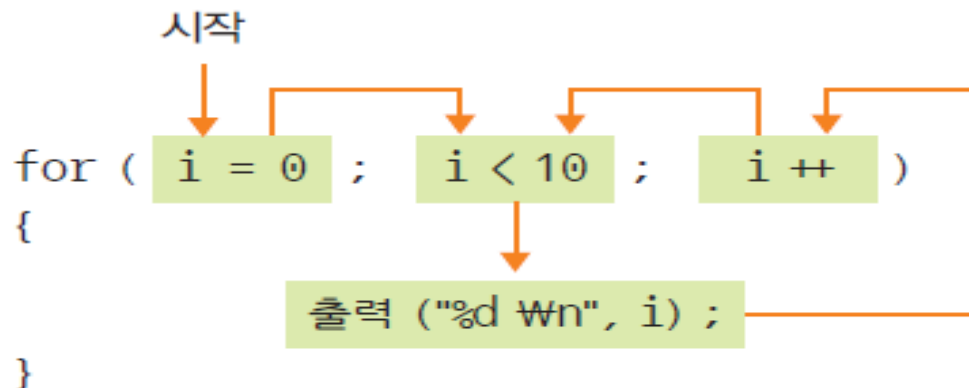
==

③ 초깃값과 증감식 빼기

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

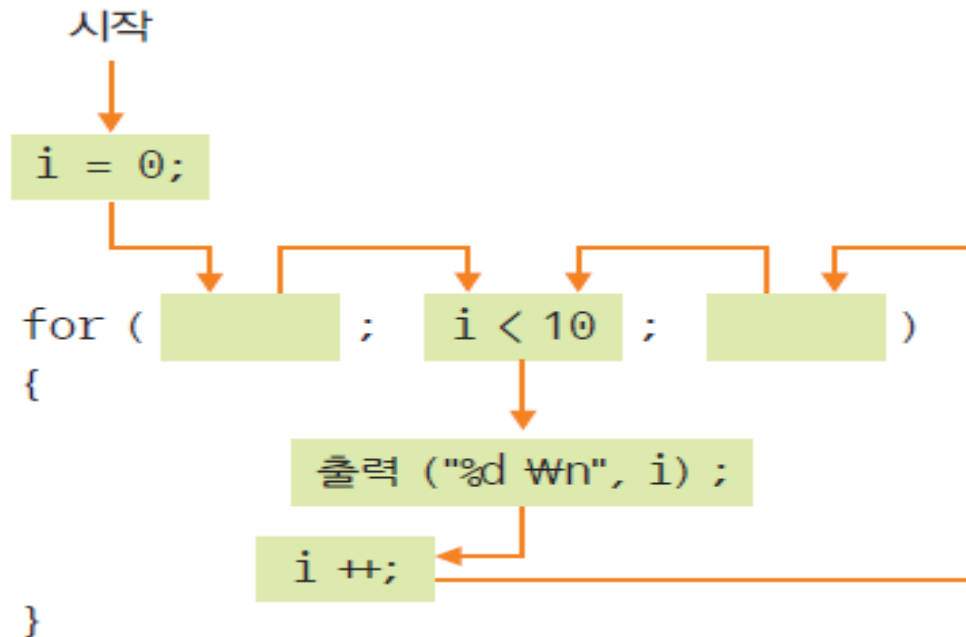
03 기타 for문

■ 초깃값과 증감식이 없는 for 문



❶의 동작 과정

그림 6-26 두 소스코드의 비교



❸의 동작 과정

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

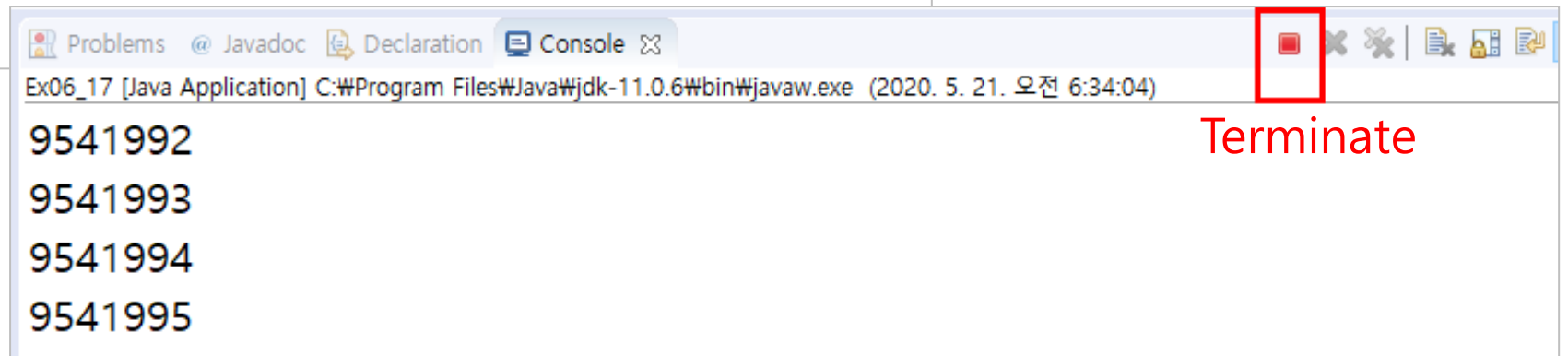
❸ 시작 → i=0 → 빈칸 → i<10 → System.out.printf() 메소드 → i++ → 빈칸 → i<10 → System.out.printf() 메소드 → i++ → ...

03 기타 for문

■ 초깃값과 증감식이 없는 for 문

- [실습6-17] 다양한 for문의 형태2 : for(;;)
- 조건식이 없어서 실행결과가 멈추지 않음. [Terminate] 버튼을 눌러 멈춤

```
1 public class Ex06_17 {  
2     public static void main(String[] args) {  
3         int i;  
4         i = 0;  
5         for (;;) { //초깃값, 조건식, 증감식이 없다.  
6             System.out.printf("%d \n", i);  
7             i++;  
8         }  
9     }  
10 }  
11 }
```

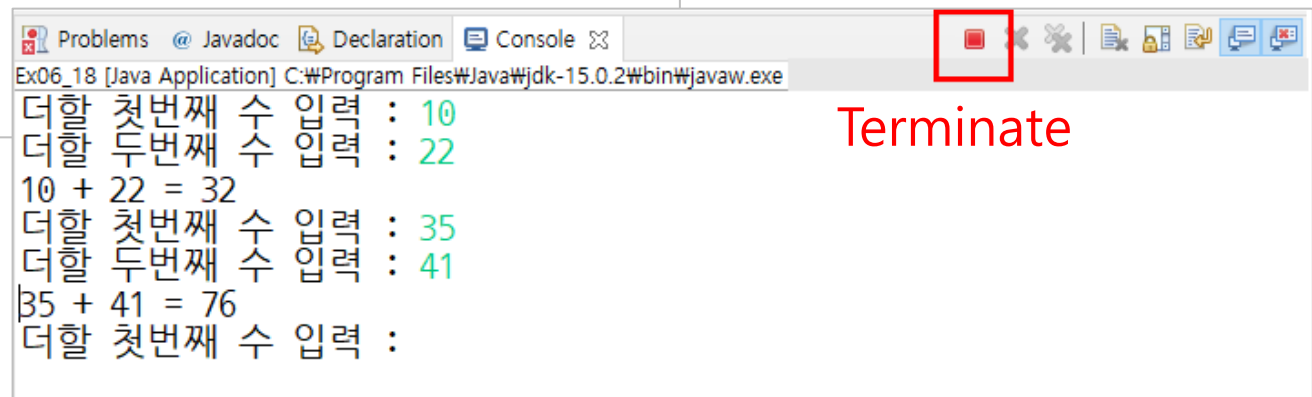


03 기타 for문

■ 초깃값과 증감식이 없는 for 문

■ [실습6-18] 다양한 for문의 형태3

```
1 import java.util.Scanner;
2
3 public class Ex06_18 {
4     public static void main(String[] args) {
5         Scanner s = new Scanner(System.in);
6         int a, b;
7
8         for (;;) { //무한루프
9             System.out.printf("더할 첫번째 수 입력 : ");
10
11             a = s.nextInt(); //숫자 2개를 입력
12
13             System.out.printf("더할 두번째 수 입력 : ");
14
15             b = s.nextInt(); //숫자 2개를 입력
16
17             System.out.printf("%d + %d = %d \n", a, b, a + b);
18         }
19     }
20 }
21 }
```



Problems @ Javadoc Declaration Console Ex06_18 [Java Application] C:\Program Files\Java\jdk-15.0.2\bin\javaw.exe

```
더할 첫번째 수 입력 : 10
더할 두번째 수 입력 : 22
10 + 22 = 32
더할 첫번째 수 입력 : 35
더할 두번째 수 입력 : 41
35 + 41 = 76
더할 첫번째 수 입력 :
```

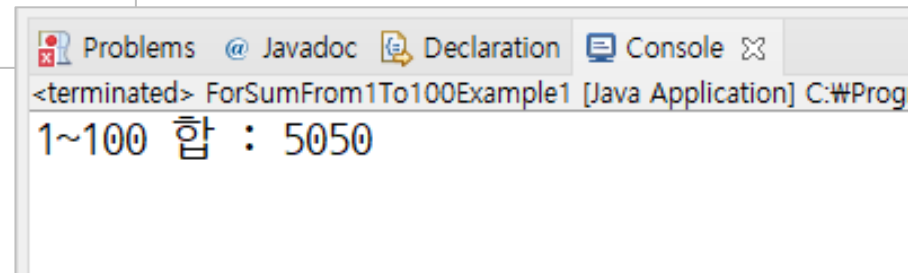
Terminate

03 기타 for문

■ 기타 for 문

- 예1. 1~100까지 합을 구하는 프로그램
- For문이 시작하기 전에 변수 sum을 선언한 이유는 for문을 끝내고 sum을 사용하기 때문임

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

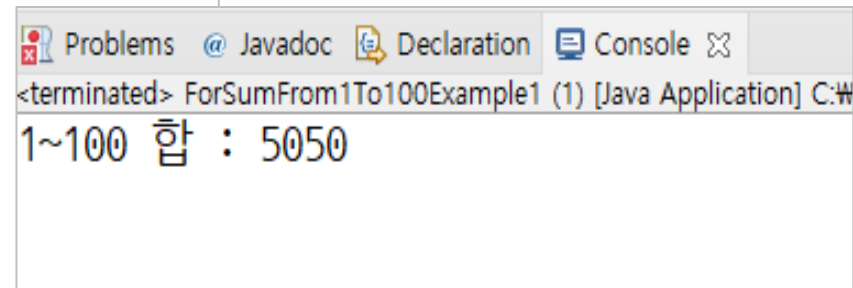


03 기타 for문

■ 기타 for 문

- 예2. 1~100까지 합을 구하는 프로그램
- 변수 i의 초기값을 for문에서 선언하지 않고 for문 전에 선언 : for문 내부 그리고 외부에서도 사용 가능

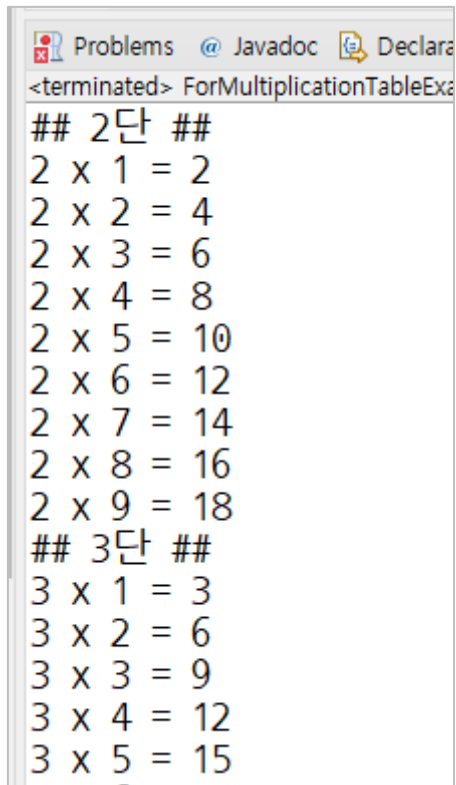
```
1 package sec02.exam03;
2
3 public class ForSumFrom1To100Example1 {
4     public static void main(String[] args) {
5         int sum = 0; //합계변수
6
7         int i = 0; //루프카운터 변수
8         for(i=1; i<=100; i++) {
9             sum += i;
10        }
11
12        System.out.println("1~" + (i-1) + " 합 : " + sum);
13    }
14 }
```



- For문 작성시 주의할 점 : 초기화식에서 루프카운터 변수를 선언할 때 부동소수점을 쓰는 float 타입을 사용하면 안됨.

Self study 8-2

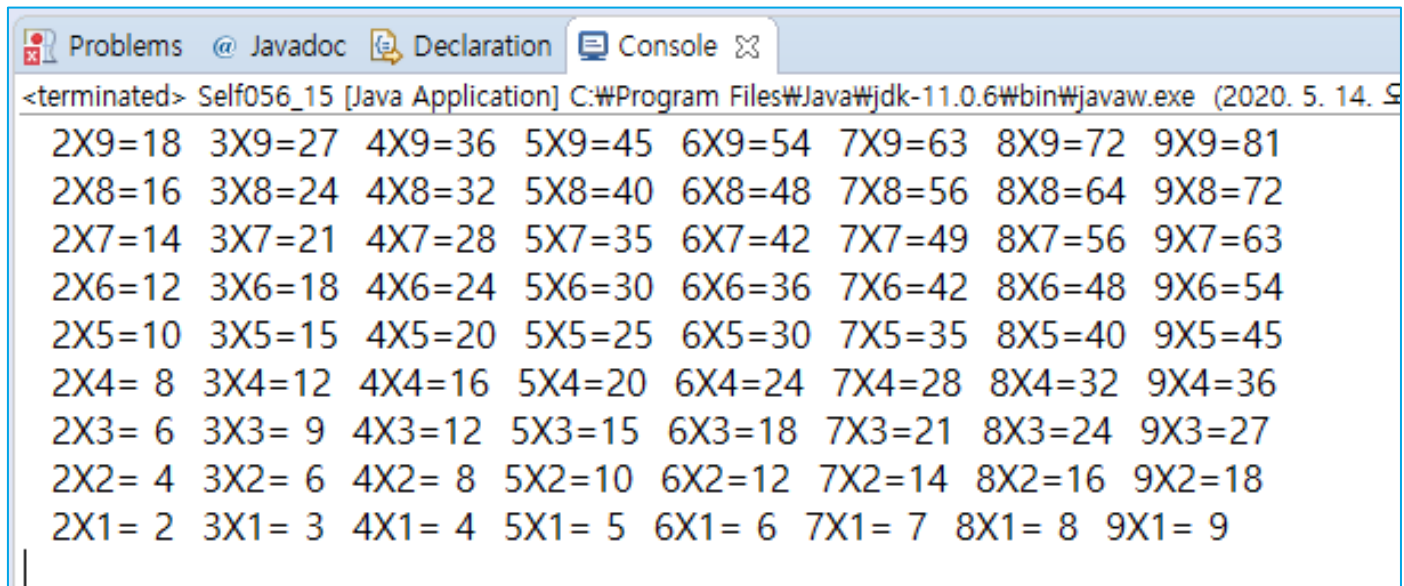
1. [실습6-13]을 수정하여 i값은 2부터 9까지, k값은 1부터 9까지 출력되도록 해보자. 즉 72행(8X9)이 출력되어야 한다.
2. [실습6-14]를 수정하여 다음 그림과 같이 각 단의 시작을 알리는 문구를 넣어보자.



```
Problems @ Javadoc Declara
<terminated> ForMultiplicationTableExa
## 2단 ##
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
## 3단 ##
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
```

Self study 8-2

3. [실습6-15]를 수정하여 다음과 같이 구구단이 거꾸로 출력되게 해보자.



The screenshot shows a Java IDE window with the 'Console' tab selected. The title bar indicates the application is 'Self056_15 [Java Application]' running at 'C:\Program Files\Java\jdk-11.0.6\bin\javaw.exe' on '2020. 5. 14. 9'. The console output displays a 9x9 multiplication table in reverse order, starting from 2x9 at the top and ending with 2x1 at the bottom. Each row contains nine multiplication problems and their results, separated by spaces.

```
<terminated> Self056_15 [Java Application] C:\Program Files\Java\jdk-11.0.6\bin\javaw.exe (2020. 5. 14. 9)
2X9=18 3X9=27 4X9=36 5X9=45 6X9=54 7X9=63 8X9=72 9X9=81
2X8=16 3X8=24 4X8=32 5X8=40 6X8=48 7X8=56 8X8=64 9X8=72
2X7=14 3X7=21 4X7=28 5X7=35 6X7=42 7X7=49 8X7=56 9X7=63
2X6=12 3X6=18 4X6=24 5X6=30 6X6=36 7X6=42 8X6=48 9X6=54
2X5=10 3X5=15 4X5=20 5X5=25 6X5=30 7X5=35 8X5=40 9X5=45
2X4= 8 3X4=12 4X4=16 5X4=20 6X4=24 7X4=28 8X4=32 9X4=36
2X3= 6 3X3= 9 4X3=12 5X3=15 6X3=18 7X3=21 8X3=24 9X3=27
2X2= 4 3X2= 6 4X2= 8 5X2=10 6X2=12 7X2=14 8X2=16 9X2=18
2X1= 2 3X1= 3 4X1= 4 5X1= 5 6X1= 6 7X1= 7 8X1= 8 9X1= 9
```

질문은 이메일을 이용해주세요.
ds.june2@gmail.com

감사합니다