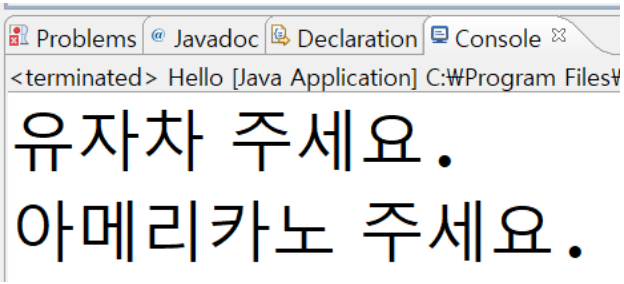
Java

SMHRD

지난 시간 문제

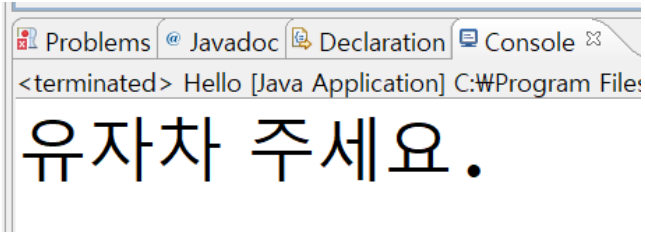
단순 if문 여러 개

```
int 돈 = 3000;

if (돈 >= 3000) {
    System.out.println("유자차 주세요.");
}
if (돈 >= 2000) {
    System.out.println("아메리카노 주세요.");
}
```



다중 if문

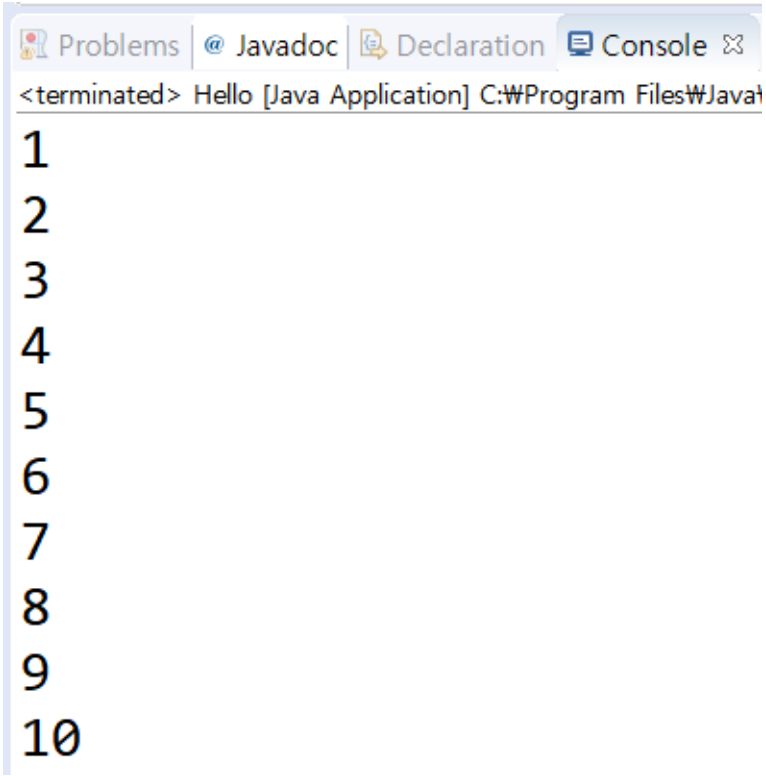
```
int 돈 = 3000;

if (돈 >= 3000) {
    System.out.println("유자차 주세요.");
} else if (돈 >= 2000) {
    System.out.println("아메리카노 주세요.");
}
```



학습목표

1. 반복문의 필요성을 이해한다.
2. 반복문의 종류와 특성을 안다.

반복문이란?




```
System.out.println(1);
System.out.println(2);
System.out.println(3);
System.out.println(4);
System.out.println(5);
System.out.println(6);
System.out.println(7);
System.out.println(8);
System.out.println(9);
System.out.println(10);
```

반복문이란?

```
System.out.println(1);
System.out.println(2);
System.out.println(3);
System.out.println(4);
System.out.println(5);
System.out.println(6);
System.out.println(7);
System.out.println(8);
System.out.println(9);
System.out.println(10);
```

```
for(int i = 1; i<=10; i++) {
    System.out.println(i);
}
```

반복문이란?

- **반복문**이란? :

어떤 조건에 만족할 때까지 실행문장을 반복하여 실행하는 구조

```
    System.out.println(1);
    System.out.println(1);
    System.out.println(1);
for(int i = 1; i<=10; i++) {
    System.out.println(1);
}
    System.out.println(1);
    System.out.println(1);
    System.out.println(1);
    System.out.println(1);
```

- 반복문이 필요한 이유 :

- 특정한 명령을 반복적으로 사용하기 위해 사용
- 코드의 간소화

반복문이란?

- 반복문의 종류

for문

while문

for문

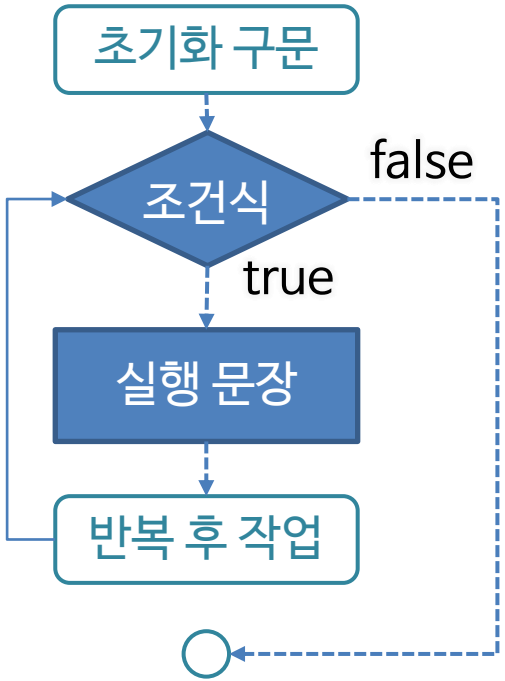
1. for문

- 주로 반복 횟수가 정해진 경우에 사용함.

〈for문의 구조〉

```
for (초기화 구문 ; 조건식 ; 반복 후 작업) {
    실행문장;
    // 조건식 결과가 true일 동안 실행됨
}
```

〈for문의 흐름도〉



for문

1-1. for문 예제

for문을 사용하여 1에서 10까지 출력하시오.

```
for(    초기화 구문     ;   조건식    ;    반복 후 작업      ) {
        실행문장   ;
}
```

```
for (int i = 1; i <= 10; i++ ) {
    System.out.println(i);
}
```

1-2. for문 예제

for문을 사용하여 21에서 57까지 출력하시오.

for문을 사용하여 96에서 53까지 출력하시오.

for문을 사용하여 21에서 57까지의 수 중 홀수만 출력하시오.

1-3. for문 예제

1. 1~100까지 중 3의 배수를 출력하세요.

2. 1~100까지 중 3의 배수이면서 5의 배수를 출력하세요.

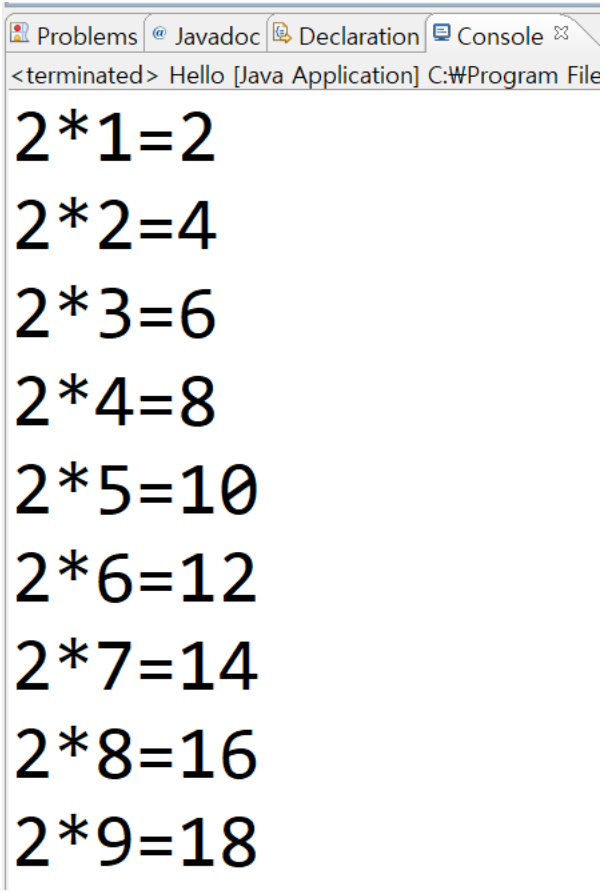
3. 한 개의 자연수를 입력 받아 그 수의 배수를 차례로 10개 출력하는 프로그램을 작성하시오.

```
자연수를 입력하세요 : 5
5    10    15    20    25    30    35    40    45    50
```

for문

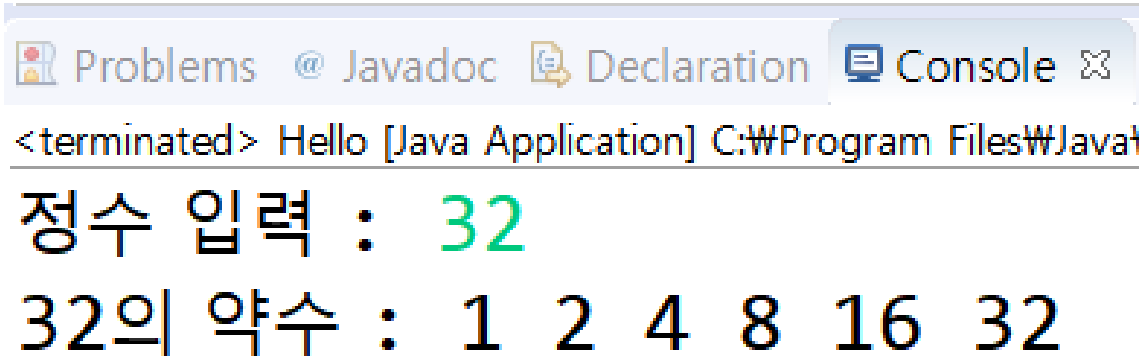
1-4. for문 예제

for문을 사용하여 구구단 2단을 출력하세요.



1-5. for문 예제

입력 받은 정수의 약수를 구하세요.



1-6. for문 예제

1부터 1000까지의 숫자 중 완전수인 숫자를 모두 출력하세요.

완전수란?
자신의 약수 중에서 자신을 제외한 모든 약수의 합이 자신
과 같은 수를 의미함.

Ex) 6의 약수 : 1,2,3,4
자신을 제외한 약수의 합은? 1+2+3
자신과 자신을 제외한 약수의 합이 같기 때문에 6은 완전수!

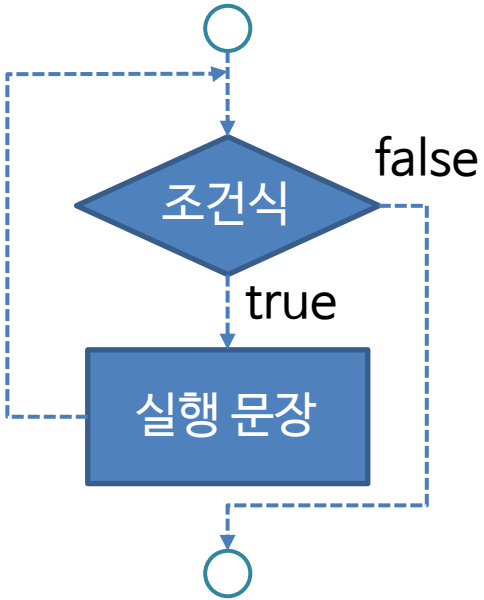
2. while문

- 정확하게 몇 번 반복해야 할 지 정해지지 않은 경우에 사용.

〈while문의 구조〉

```
while (조건식) {
    실행문장;
    //조건식 결과가 true일
      동안 실행됨
}
```

〈while문의 흐름도〉



while문

2-1. while문 예제

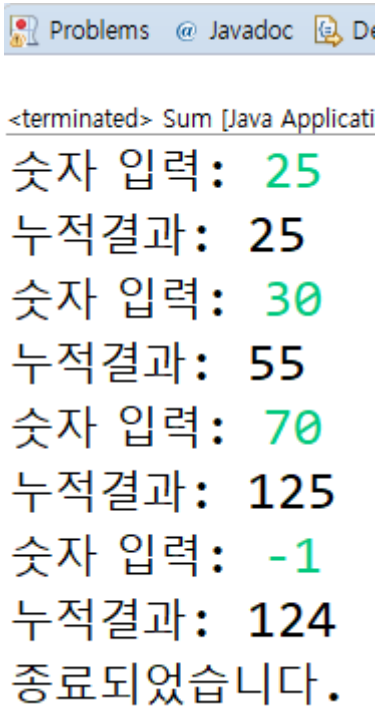
while문을 사용하여 키보드로부터 입력 받은 수가
10보다 작을 때만 계속 정수를 입력 받으세요.

* 10보다 큰 수를 입력하면 “종료되었습니다”를 출력

2-2. while문 예제

숫자를 입력 받아 누적하는 프로그램을 작성하세요.

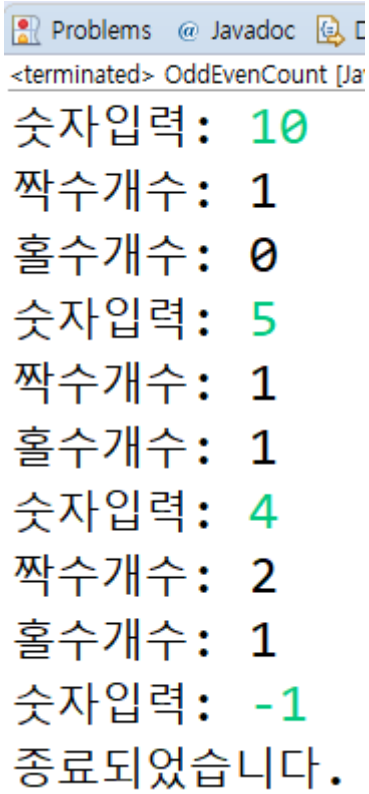
*-1을 입력한 경우 프로그램 종료



2-3. while문 예제

숫자를 입력 받아 홀수와 짝수가 각각 몇 개 입력되었는지 출력하는 프로그램을 작성하세요.

*-1을 입력한 경우 프로그램 종료



do-while문

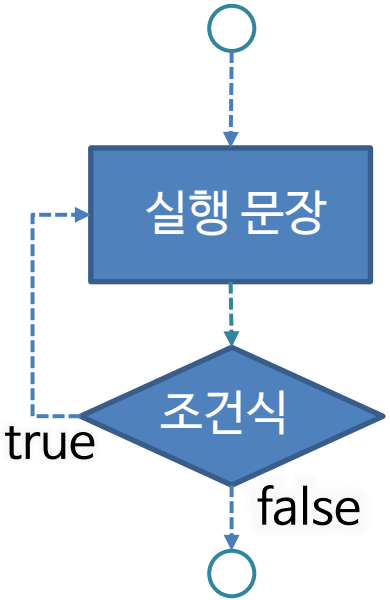
3. do-while문

- 정확하게 몇 번 반복해야 할 지 정해지지 않은 경우에 사용.

〈do-while문의 구조〉

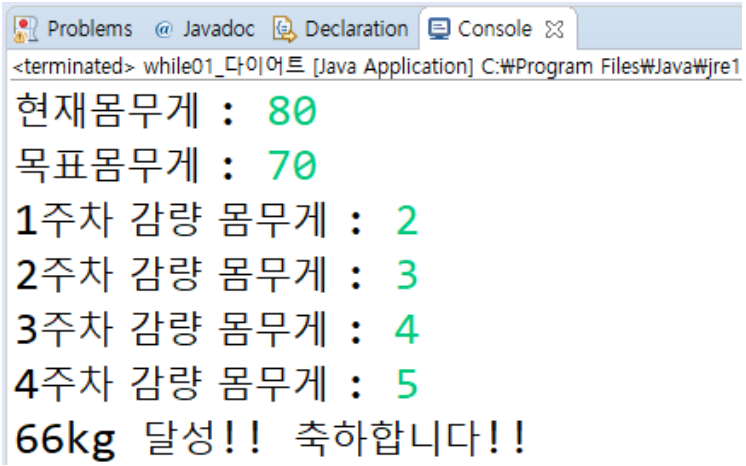
```
do {
    반드시 한 번은
    실행되어야 하는 문장
    //그 후 조건식 결과가
      true일 동안 실행됨
} while (조건식);
```

〈do-while문의 흐름도〉



3-1. 다이어트 관리 프로그램

1. 현재 몸무게와 목표몸무게를 입력 받고 주차 별 감량 몸무게를 입력 받으세요.
2. 목표 몸무게를 달성하면 축하한다는 문구를 출력하고 입력을 멈추세요!



3-2. Login프로그램

로그인 프로그램을 만들어 보자.
아이디와 비밀번호를 각각 입력 받고 일치할 경우 “로그인성공!”
일치하지 않은 경우에는 “로그인 실패!”

아이디 : Hello　　비밀번호 : 1234

```
아이디를 입력해 주세요 >> hi
비밀번호를 입력해 주세요 >> 1234
아이디와 비밀번호가 잘못되었습니다.
```

```
아이디를 입력해 주세요 >> bye
비밀번호를 입력해 주세요 >> 1234
아이디와 비밀번호가 잘못되었습니다.
```

```
아이디를 입력해 주세요 >> Hello
비밀번호를 입력해 주세요 >> 1234
로그인 성공!
```

3-2. Login프로그램

로그인이 실패했을 경우에 계속 입력
로그인이 성공하면 프로그램 종료

```
아이디를 입력해 주세요 >> hello
비밀번호를 입력해 주세요 >> 1234
아이디와 비밀번호가 잘못되었습니다.
아이디를 입력해 주세요 >> bye
비밀번호를 입력해 주세요 >> 1234
아이디와 비밀번호가 잘못되었습니다.
아이디를 입력해 주세요 >> Hello
비밀번호를 입력해 주세요 >> 1234
로그인 성공!
```

3-2. Login프로그램

아이디와 비밀번호가 틀렸을 경우 계속 하시겠습니까?
라는 문장을 출력하세요.
Y를 입력하면 아이디 비밀번호 입력 계속,
N을 입력하면 종료/로그인 성공 시 종료

```
아이디를 입력해 주세요 >> hi
비밀번호를 입력해 주세요 >> 1234
아이디와 비밀번호가 잘못되었습니다.
계속 하시겠습니까 ? (Y/N) >>Y
아이디를 입력해 주세요 >> hello
비밀번호를 입력해 주세요 >> 1234
아이디와 비밀번호가 잘못되었습니다.
계속 하시겠습니까 ? (Y/N) >>N
종료합니다
```

```
아이디를 입력해 주세요 >> hello
비밀번호를 입력해 주세요 >> 1234
아이디와 비밀번호가 잘못되었습니다.
계속 하시겠습니까 ? (Y/N) >>Y
아이디를 입력해 주세요 >> Hello
비밀번호를 입력해 주세요 >> 1234
로그인 성공!
```

3-2. Login프로그램

아이디와 비밀번호가 3번 이상 틀렸을 때,
아이디와 비밀번호가 3회 틀렸습니다. 본인인증을 해주세요.
라는 문장을 출력하고 프로그램을 종료하세요.

```
아이디를 입력해 주세요 >> hi
비밀번호를 입력해 주세요 >> 1234
아이디와 비밀번호가 잘못되었습니다.
계속 하시겠습니까 ? (Y/N) >>Y
아이디를 입력해 주세요 >> bye
비밀번호를 입력해 주세요 >> 456
아이디와 비밀번호가 잘못되었습니다.
계속 하시겠습니까 ? (Y/N) >>Y
아이디를 입력해 주세요 >> hello
비밀번호를 입력해 주세요 >> 1234
아이디와 비밀번호가 3회 틀렸습니다. 본인인증을 해주세요.
```

3-3. (+, -) 계산기프로그램

- 첫 번째 정수를 입력 받는다.
- 두 번째 정수를 입력 받는다.
- 연산자를 선택한다.
- 선택한 연산자에 따라 연산결과를 출력한다.
- 다시 실행할 것인가를 물어본다.
-“Y” 를 입력하면 다시 실행.
-“N”을 입력하면 반복문 종료.

```
첫 번째 정수를 입력하세요 >> 50
두 번째 정수를 입력하세요 >> 25
[1]더하기 [2]빼기 >> 1
더하기 연산 결과는 75입니다.
다시 실행하시겠습니까? (Y/N) >> Y
첫 번째 정수를 입력하세요 >> 30
두 번째 정수를 입력하세요 >> 46
[1]더하기 [2]빼기 >> 2
빼기 연산 결과는 -16입니다.
다시 실행하시겠습니까? (Y/N) >> N
종료합니다.
```

3-4. PlusGame

랜덤으로 정수 2개를 뽑아 아래와 같이 출력
사용자는 두 수의 합을 입력
두 수의 합과 입력한 수가 일치하면 “Success”
두 수의 합과 입력한 수가 일치하지 않으면 “Fail”을 출력

```
===Plus Game===
9+2=11
Success
```

```
===Plus Game===
7+5=10
Fail
```

3-4. PlusGame

두 수의 합이 일치하지 않았을 때만
다시 실행할 것인지 물어보고 “Y”를 입력하면
계속 실행 “N” 입력하면 프로그램을 종료한다.

```
===Plus Game===
3+4=7
Success
6+9=15
Success
6+3=9
Success
5+3=2
Fail
계속 하시겠습니까? >> Y
5+4=9
Success
8+5=2
Fail
계속 하시겠습니까? >> N
종료합니다.
```

do-while문

가위바위보 게임입니다.
user1, user2의 이름을 각각 입력 받고 결과를 출력하세요.

```
user1 이름입력 : 박민지
user2 이름입력 : 박근영
박민지님 가위,바위,보 중 하나를 입력하세요 >>가위
박근영님 가위,바위,보 중 하나를 입력하세요 >>바위
박근영님 승리!
```

do-while문

게임 종료 후 '다시 하시겠습니까?'라는 문장을 출력하세요.
'Y' 또는 'y'를 입력하면 게임진행,
'N' 또는 'n'을 입력하면 종료

```
user1 이름입력 : 박민지
user2 이름입력 : 박근영
박민지님 가위,바위,보 중 하나를 입력하세요 >>가위
박근영님 가위,바위,보 중 하나를 입력하세요 >>바위
박근영님 승리!
다시 하시겠습니까?Y
박민지님 가위,바위,보 중 하나를 입력하세요 >>바위
박근영님 가위,바위,보 중 하나를 입력하세요 >>바위
비겼습니다!
다시 하시겠습니까?Y
박민지님 가위,바위,보 중 하나를 입력하세요 >>보
박근영님 가위,바위,보 중 하나를 입력하세요 >>바위
박민지님 승리!
다시 하시겠습니까?N
프로그램을 종료합니다.
```

do-while문

1부터 100 사이의 랜덤한 숫자 num을 생성하고, 사용자로부터 숫자 input을 입력 받아 num과 input이 일치할 때까지 반복하는 프로그램을 만들어주세요.

```
======1부터 100 사이 숫자 맞추기 게임!======
1과 100사이의 정수를 입력하세요 >>50
1과 100사이의 정수를 입력하세요 >>60
1과 100사이의 정수를 입력하세요 >>22
1과 100사이의 정수를 입력하세요 >>4
1과 100사이의 정수를 입력하세요 >>50
1과 100사이의 정수를 입력하세요 >>8
1과 100사이의 정수를 입력하세요 >>30
1과 100사이의 정수를 입력하세요 >>
```

do-while문

사용자가 입력한 숫자(input)가 랜덤으로 생성된 숫자(num)보다 더 큰 경우
"더 작은 수로 다시 시도 해보세요." 라는 문구가 출력되고
사용자가 입력한 숫자(input)가 랜덤으로 생성된 숫자(num)보다 더 작은 경우
"더 큰 수로 다시 시도 해보세요." 라는 문구가 출력되게 만들어주세요.

```
======1부터 100 사이 숫자 맞추기 게임!======
1과 100사이의 정수를 입력하세요 >>50
더 작은 수로 다시 시도 해보세요
1과 100사이의 정수를 입력하세요 >>40
더 작은 수로 다시 시도 해보세요
1과 100사이의 정수를 입력하세요 >>30
더 작은 수로 다시 시도 해보세요
1과 100사이의 정수를 입력하세요 >>20
더 큰 수로 다시 시도 해보세요
1과 100사이의 정수를 입력하세요 >>25
더 큰 수로 다시 시도 해보세요
1과 100사이의 정수를 입력하세요 >>27
더 큰 수로 다시 시도 해보세요
1과 100사이의 정수를 입력하세요 >>28
28 정답입니다!!
```

다중 for문

다중 for문

<다중 for문의 구조>

```
for (int j = 1; j <= 3; j++) {
    System.out.println("j : " + j + " ");
    for (int i = 1; i <= 3; i++) {
        System.out.println("i : " + i + " ");
    } System.out.println("===========");
}
```

Problems Javadoc Declaration Console

<terminated> Hello [Java Application] C:₩Program Files₩Java

```
j : 1
i : 1
i : 2
i : 3
===========
j : 2
i : 1
i : 2
i : 3
===========
j : 3
i : 1
i : 2
i : 3
===========
```

4-1. 다중 for문 예제

구구단 2단~9단까지 출력하세요.

```
2단 : 2*1=2 2*2=4 2*3=6 2*4=8 2*5=10 2*6=12 2*7=14 2*8=16 2*9=18
3단 : 3*1=3 3*2=6 3*3=9 3*4=12 3*5=15 3*6=18 3*7=21 3*8=24 3*9=27
4단 : 4*1=4 4*2=8 4*3=12 4*4=16 4*5=20 4*6=24 4*7=28 4*8=32 4*9=36
5단 : 5*1=5 5*2=10 5*3=15 5*4=20 5*5=25 5*6=30 5*7=35 5*8=40 5*9=45
6단 : 6*1=6 6*2=12 6*3=18 6*4=24 6*5=30 6*6=36 6*7=42 6*8=48 6*9=54
7단 : 7*1=7 7*2=14 7*3=21 7*4=28 7*5=35 7*6=42 7*7=49 7*8=56 7*9=63
8단 : 8*1=8 8*2=16 8*3=24 8*4=32 8*5=40 8*6=48 8*7=56 8*8=64 8*9=72
9단 : 9*1=9 9*2=18 9*3=27 9*4=36 9*5=45 9*6=54 9*7=63 9*8=72 9*9=81
```

다중 for문

4-2. for문 예제

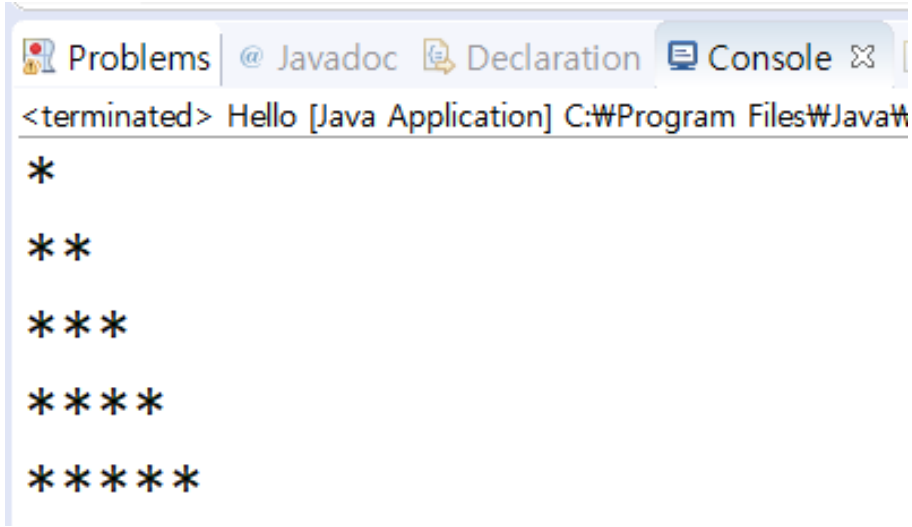
2~30까지의 약수를 구하세요.

```
2의 약수 : 1 2
3의 약수 : 1 3
4의 약수 : 1 2 4
5의 약수 : 1 5
6의 약수 : 1 2 3 6
7의 약수 : 1 7
8의 약수 : 1 2 4 8
9의 약수 : 1 3 9
10의 약수 : 1 2 5 10
11의 약수 : 1 11
12의 약수 : 1 2 3 4 6 12
13의 약수 : 1 13
14의 약수 : 1 2 7 14
15의 약수 : 1 3 5 15
16의 약수 : 1 2 4 8 16
17의 약수 : 1 17
18의 약수 : 1 2 3 6 9 18
19의 약수 : 1 19
20의 약수 : 1 2 4 5 10 20
21의 약수 : 1 3 7 21
22의 약수 : 1 2 11 22
23의 약수 : 1 23
24의 약수 : 1 2 3 4 6 8 12 24
25의 약수 : 1 5 25
26의 약수 : 1 2 13 26
27의 약수 : 1 3 9 27
28의 약수 : 1 2 4 7 14 28
29의 약수 : 1 29
30의 약수 : 1 2 3 5 6 10 15 30
```

다중 for문

4-4. for문 예제

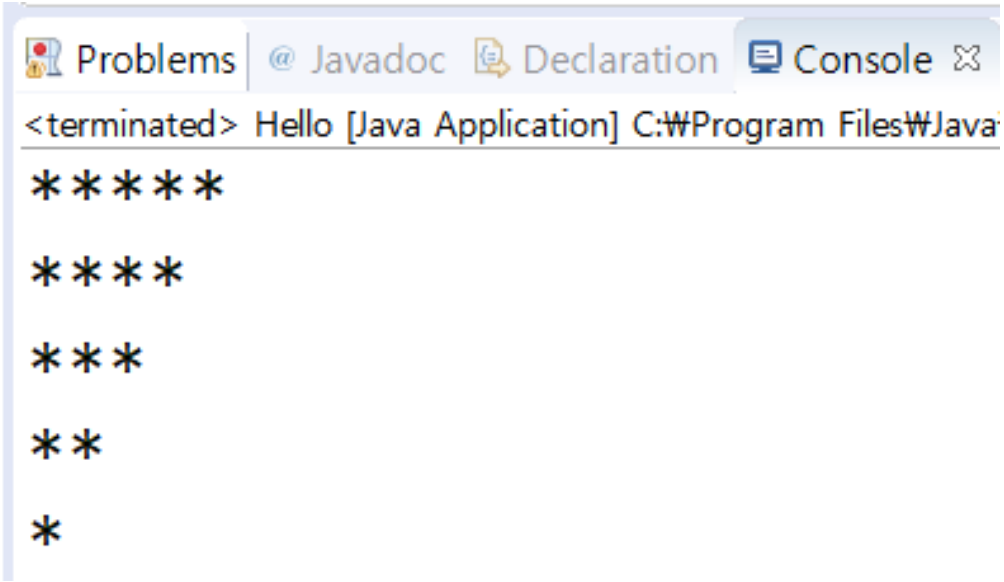
다음과 같은 별 모양으로 출력하세요.



다중 for문

4-5. for문 예제

다음과 같은 별 모양으로 출력하세요.



반복문 비교

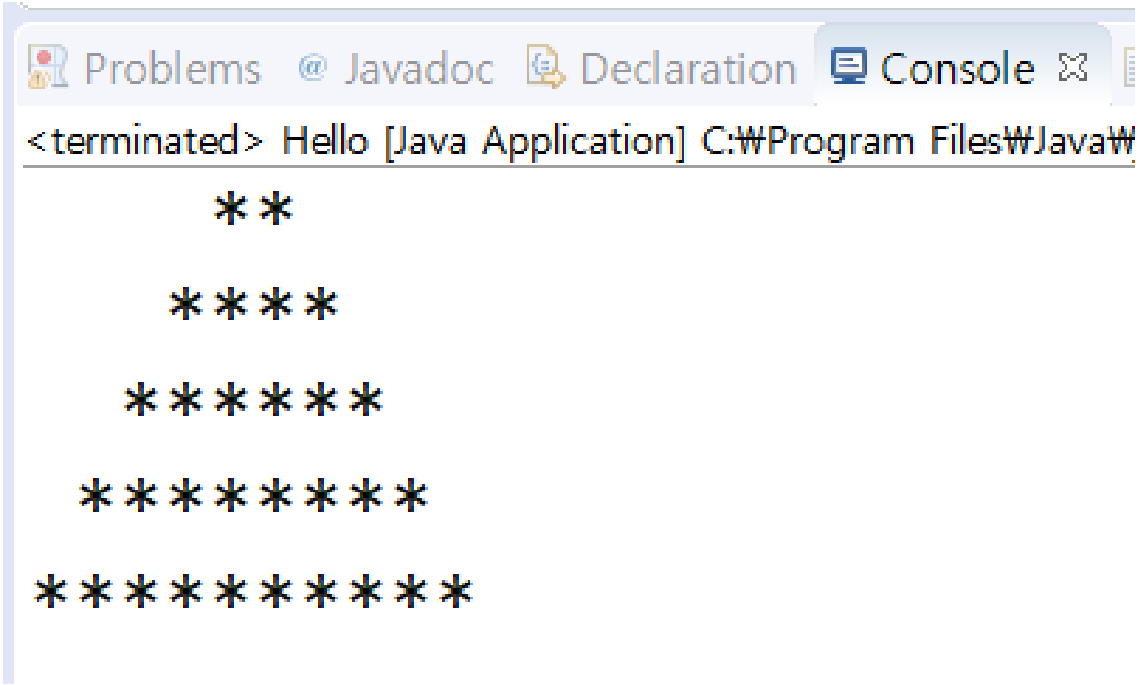
for문	일정한 반복 횟수가 정해진 경우	for (초기화 구문 ; 조건식 ; 반복 후 작업) {
		  실행문장;
		  // 조건식 결과가 true일 동안 실행됨
		}

while문	반복횟수가 정해지지 않고 조건에 따라 달라지는 경우	while (조건식) {
		  실행문장;
		  //조건식 결과가 true일 동안 실행됨
		}

do-while문	반복횟수가 정해지지 않고 조건에 따라 달라지는 경우	do {
		  반드시 한 번은 실행되어야 하는 문장
		  //그 후 조건식 결과가 true일 동안 실행됨
		} while (조건식);

마무리 문제

다음과 같은 별 모양으로 출력하세요.

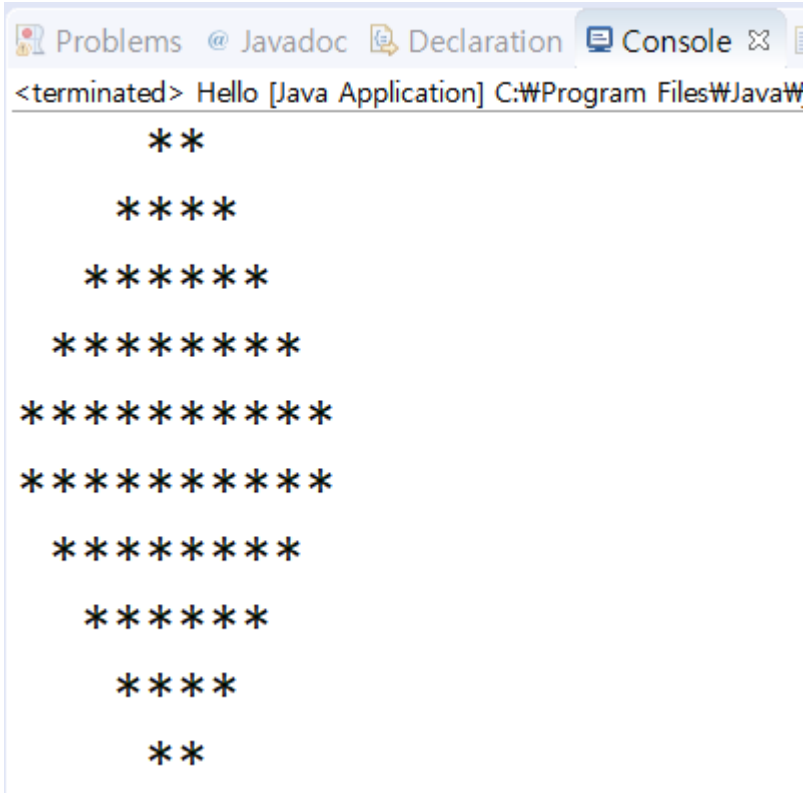


학습목표 재확인

1. 반복문의 필요성을 이해한다.
2. 반복문의 종류와 특성을 안다.

과제

다음과 같은 별 모양으로 출력하세요.



감사합니다

자바

* 다음 시간에 배울 내용
  -배열