

# 자바 프로그래밍

## 제3장 선택과 반복

# 이번 장에서 학습할 내용

## 학습목차

### 01 제어문

### 02 if-else 문

LAB 성적에 따라 학점 결정하기

LAB 인사말 출력하기

### 03 switch 문

LAB 월의 일수 출력하기

### 04 while 문

LAB 최대 공약수 찾기

### 05 for 문

LAB 약수 계산하기

LAB 난수의 합 계산하기

### 06 break와 continue

LAB 숫자 추측 게임

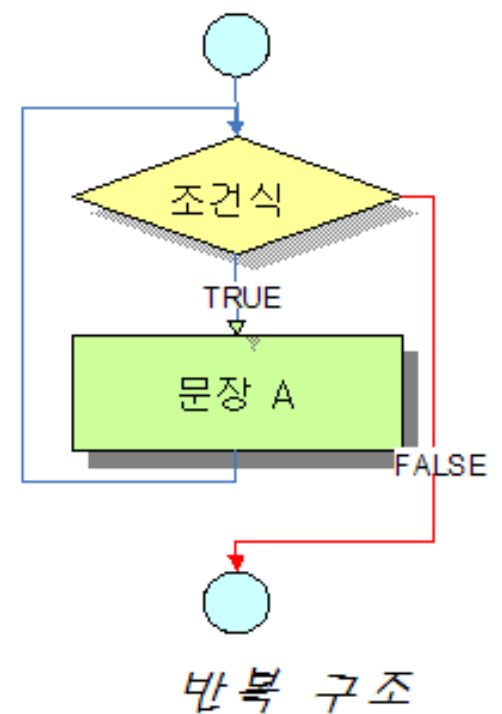
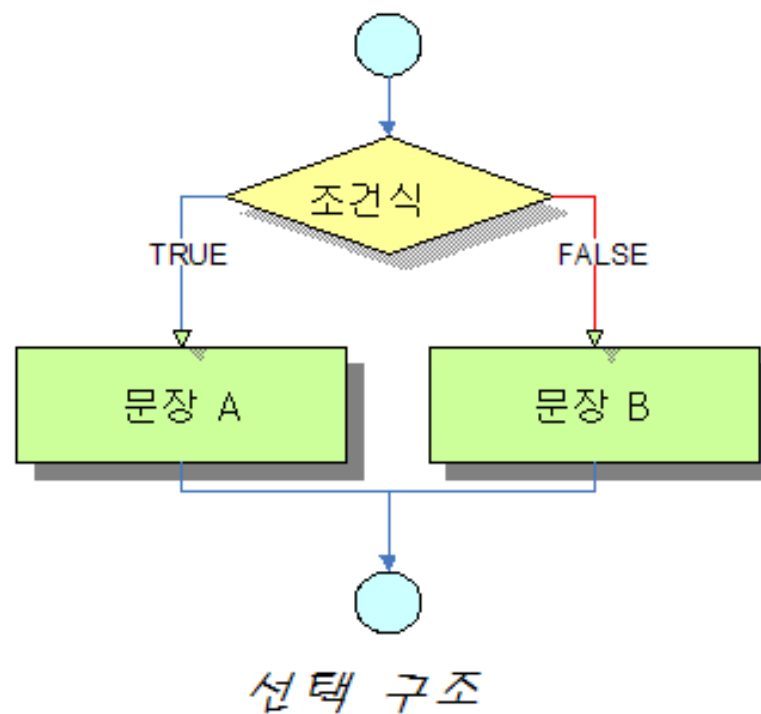
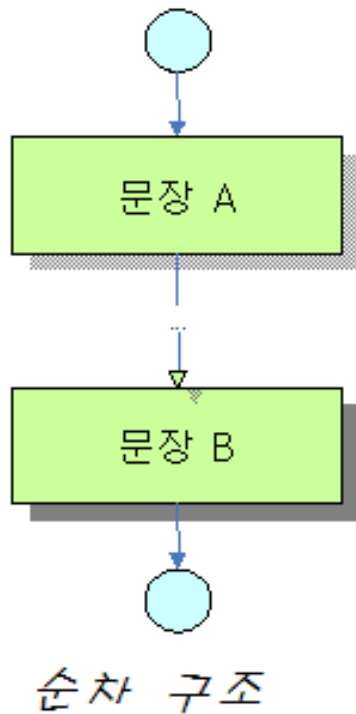
LAB 로또 확률 계산

자바에서는 조건을 따져서  
문장을 수행시키나봐요.

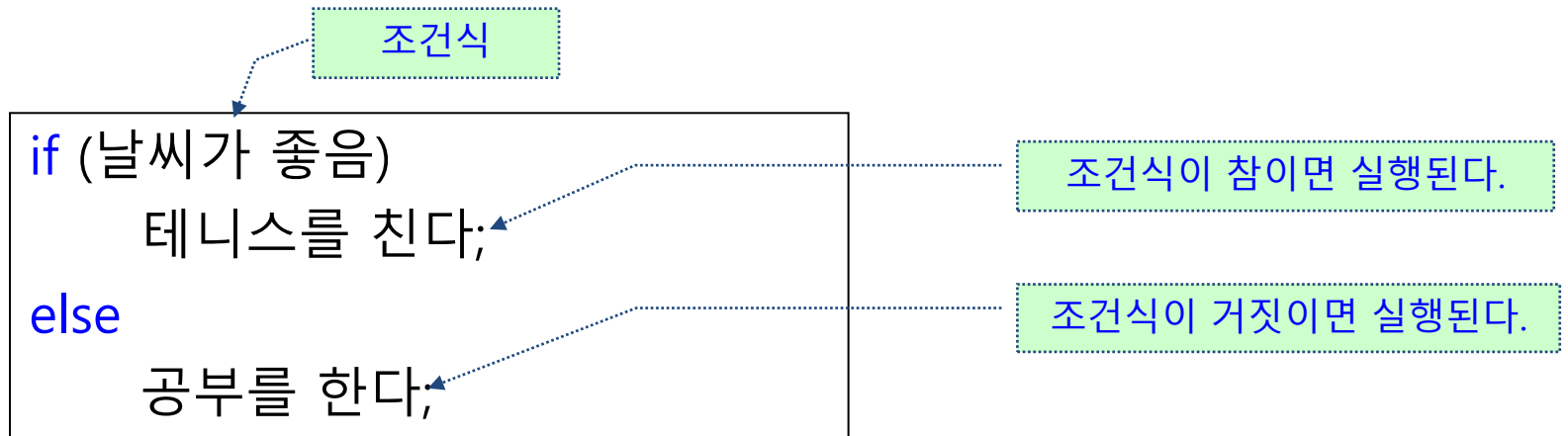
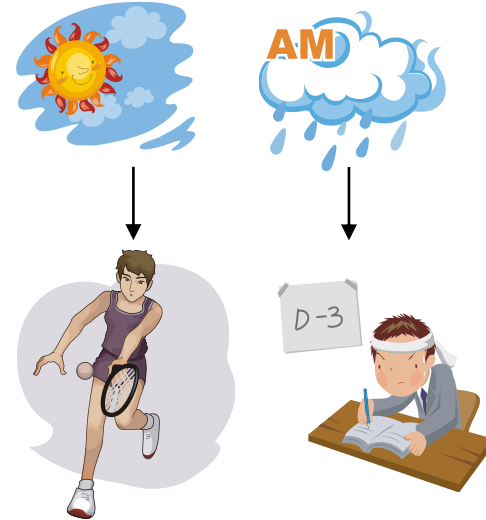
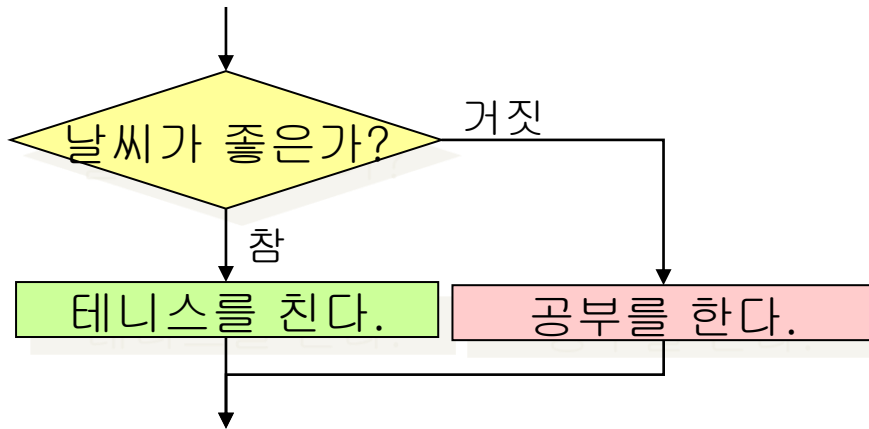
네, 조건은 프로그래밍에서는  
무척 중요합니다. 조건을 따져서  
문장을 수행시키기 때문에  
컴퓨터가 스마트한 것이죠.



# 3가지 제어 구조



# if-else 문



# 조건문의 예

```
if (grade >= 60)  
    System.out.println("합격입니다.");
```

```
if (grade >= 60)  
    System.out.println("합격입니다.");  
else  
    System.out.println("불합격입니다.");
```

# 중첩 if문

- else절은 가장 가까운 if와 매치된다.

```
score = 70;  
if( score >= 80 )  
    if( score >= 90 )  
        System.out.println("A");  
else  
    System.out.println("B");
```

=

```
score = 70;  
if( score >= 80 ) {  
    if( score >= 90 )  
        System.out.println("A");  
    else  
        System.out.println("B");  
}
```

- ✓ 읽기 쉽고 오류가 적은 프로그램을 작성하려면  
들여쓰기(indentation)를 잘 할 것  
(이클립스에서 전체 선택 후 Ctrl + i 기능 활용해 볼 것)

# 예: 짝수 홀수 구별하기

- 키보드에서 입력 받은 정수가 홀수인지 짝수인지를 말해주는 프로그램을 작성하여 보자.



정수를 입력하시오: 2  
입력된 정수는 짝수입니다.  
프로그램이 종료되었습니다.

```
01 import java.util.Scanner;
02
03 public class EvenOdd {
04     public static void main(String[] args) {
05         // if 문을 사용하여 홀수와 짝수를 구별하는 프로그램
06         int number;
07         Scanner input = new Scanner(System.in);
08         System.out.print("정수를 입력하시오: ");
09         number = input.nextInt();
```

number를 2로 나누어서  
0이면 짝수이다.

```
10
11     if (number % 2 == 0) {
12         System.out.println("입력된 정수는 짝수입니다.");
13     } else {
14         System.out.println("입력된 정수는 홀수입니다.");
15     }
16
17     System.out.println("프로그램이 종료되었습니다. ");
18 }
19 }
```



# 예: 성적에 따라 학점 결정하기 1

- if 문 사용

```
if (score >= 90)
    System.out.println( "A" );
if (score < 90 && score >= 80)
    System.out.println( "B" );
if (score < 80 && score >= 70)
    System.out.println( "C" );
if (score < 70 && score >= 60)
    System.out.println( "D" );
if (score < 60)
    System.out.println( "F" );
```

A로 결정된 이후에도 이어지는 조건 검사를 계속한다.  
→ else if 절을 이용하도록 수정해보자.

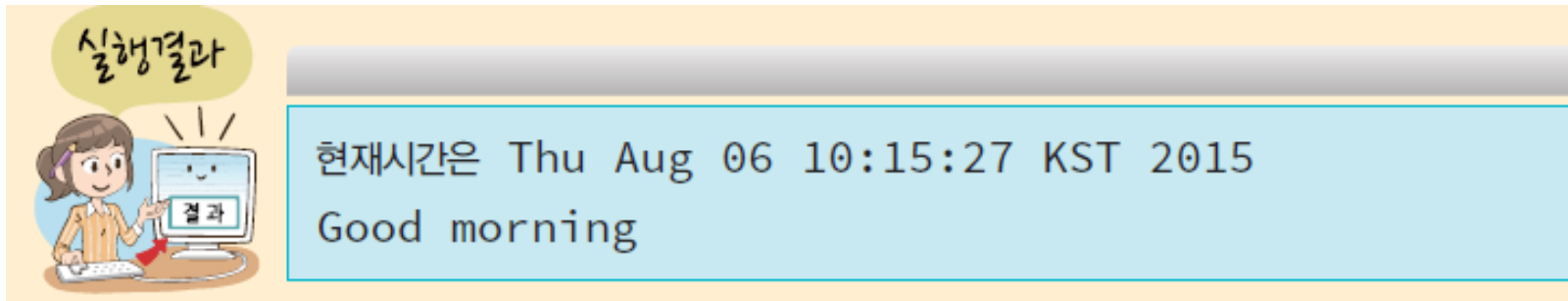
# 예: 성적에 따라 학점 결정하기 2

- 연속적인 else if 문 사용

```
if (score >= 90)
    System.out.println( "A" );
else if (score >= 80)
    System.out.println( "B" );
else if (score >= 70)
    System.out.println( "C" );
else if (score >= 60)
    System.out.println( "D" );
else
    System.out.println( "F" );
```

# 예: 인사말 출력하기

- 시스템으로부터 현재 시각을 알아내고, 시간에 따라 적절한 인사말을 출력하는 프로그램을 작성해보자.

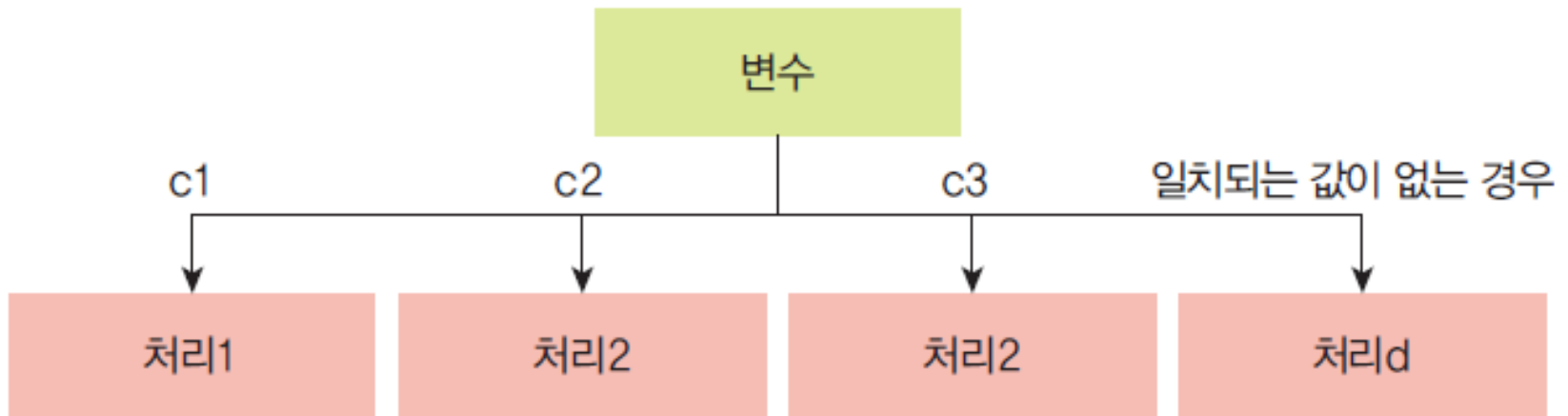


```
01 import java.util.Date;
02
03 public class Welcome {
04     public static void main(String args[]) {
05
06         Date date = new Date();
07         int currentHour = date.getHours();
08
09         System.out.println("현재시간은 "+ date);
10         if (currentHour < 11) {
11             System.out.println("Good morning");
12         } else if (currentHour < 15) {
13             System.out.println("Good afternoon");
14         } else if (currentHour < 20) {
15             System.out.println("Good evening");
16         } else {
17             System.out.println("Good night");
18         }
19     }
20 }
```

시스템으로부터 현재  
시각을 얻어온다.

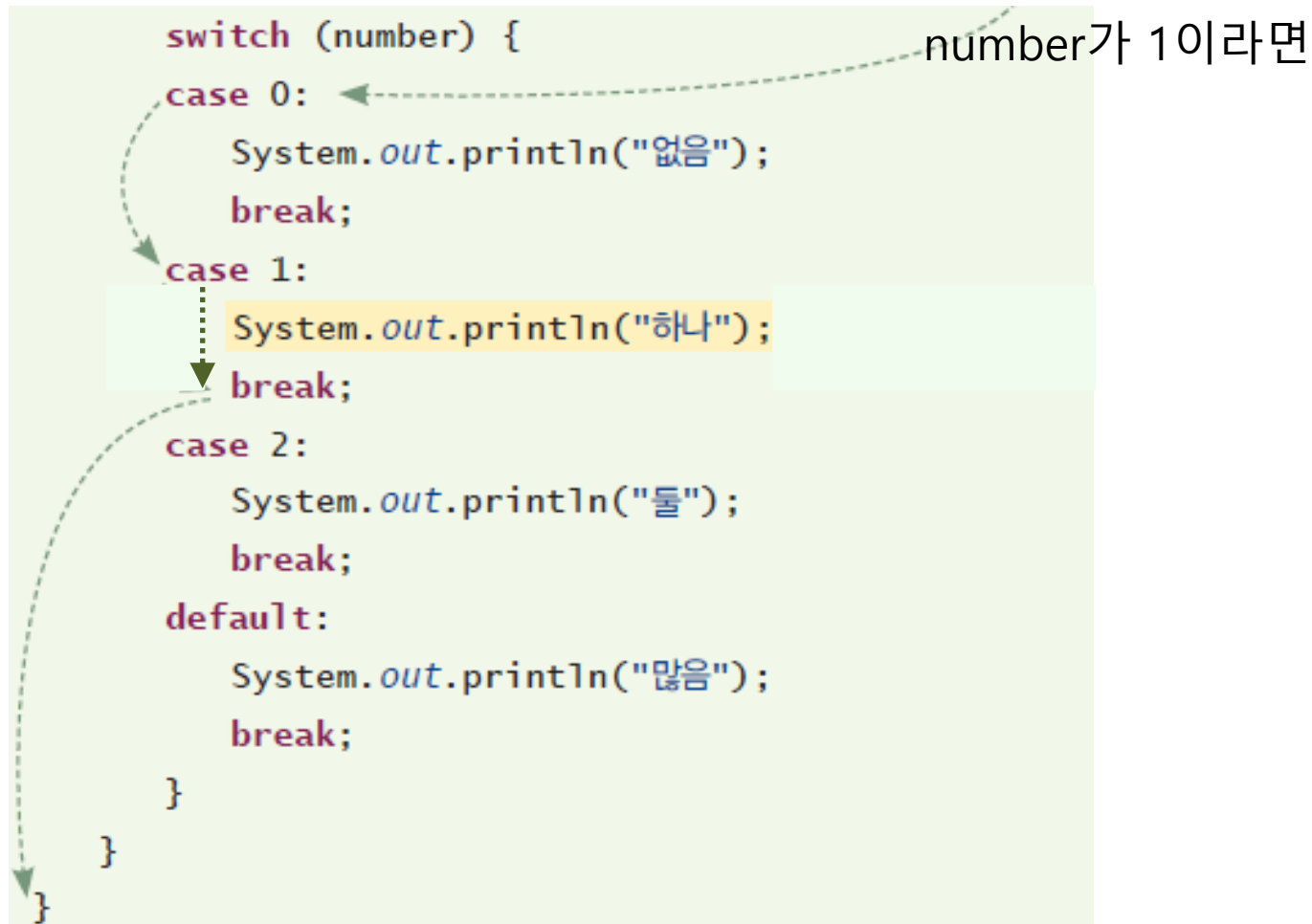
# switch 문

- if 문은 실행 경로가 두가지
- 가능한 실행 경로가 여러 개인 경우 switch 문을 사용



# switch 문의 예

- number 값에 따라 실행 경로가 달라짐



# switch 문에 문자열 사용

- JDK 7부터는 switch 문의 제어식으로 String 객체를 사용할 수 있다.
- 예: 월의 이름을 영어로 받아서, 월을 반환하는 프로그램을 작성해보자.



월의 이름을 입력하시오: january

1

```
01 import java.util.Scanner;
02
03 public class StringSwitch {
04     public static void main(String[] args) {
05         String month;
06
07         Scanner scan = new Scanner(System.in);
08         System.out.print("월의 이름을 입력하시오: ");
09         month = scan.next();
10
11         int monthNumber;
12         switch (month)
13         case "january":
14             monthNumber = 1;
15             break;
16         case "february":
17             monthNumber = 2;
18             break;
19         case "march":
20             monthNumber = 3;
21             break;
22         default:
23             monthNumber = 0;
24             break;
25     }
26     System.out.println(monthNumber);
27 }
28 }
```

String

String



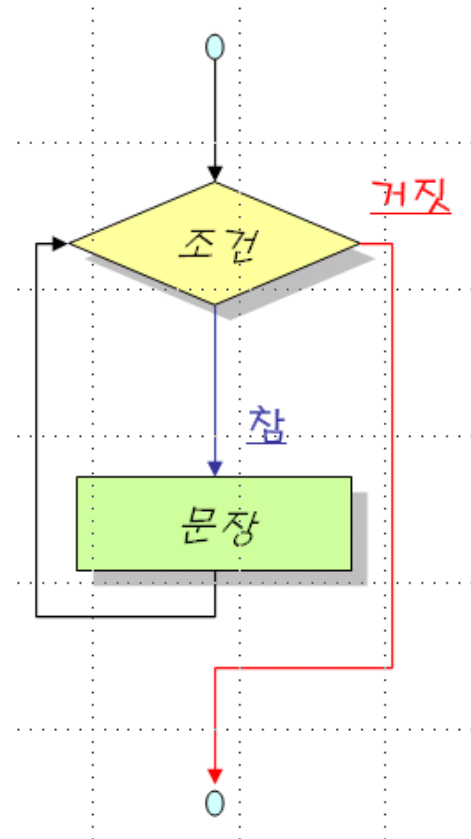
# 반복문

- 같은 처리 과정을 되풀이하는 것이 필요할 때 사용
  - while 문
  - do-while 문
  - for 문

# while 문

- 주어진 조건이 만족되는 동안 문장들을 반복 실행한다.

```
while( 조건식 ) {  
    문장;  
}
```



# 예: 환영 메시지 반복 출력

*WelcomeLoop.java*

```
01 public class WelcomeLoop {  
02     public static void main(String[] args) {  
03         int i = 0;  
04         while (i < 5) {  
05             System.out.println("환영합니다. ");  
06             i++;  
07         }  
08         System.out.println("반복이 종료되었습니다.");  
09     }  
10 }
```

환영합니다.

환영합니다.

환영합니다.

환영합니다.

환영합니다.

반복이 종료되었습니다.

# do-while 문

- 주어진 조건이 만족되는 동안 문장들을 반복 실행하되, while문과 달리 반복 조건을 루프의 끝에서 검사한다.

```
do {  
    문장;  
} while( 조건식 );
```

# 예: 올바른 입력 받기

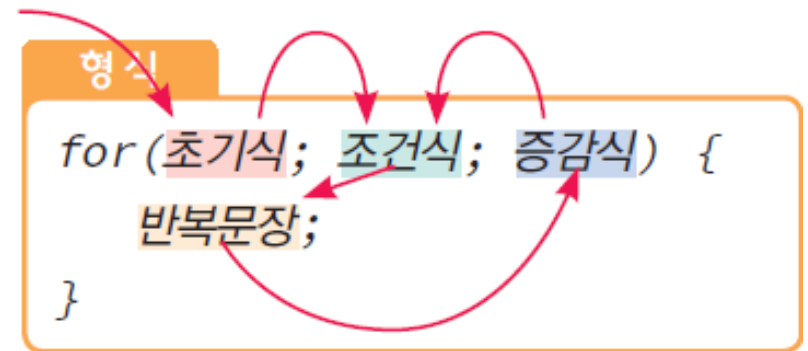
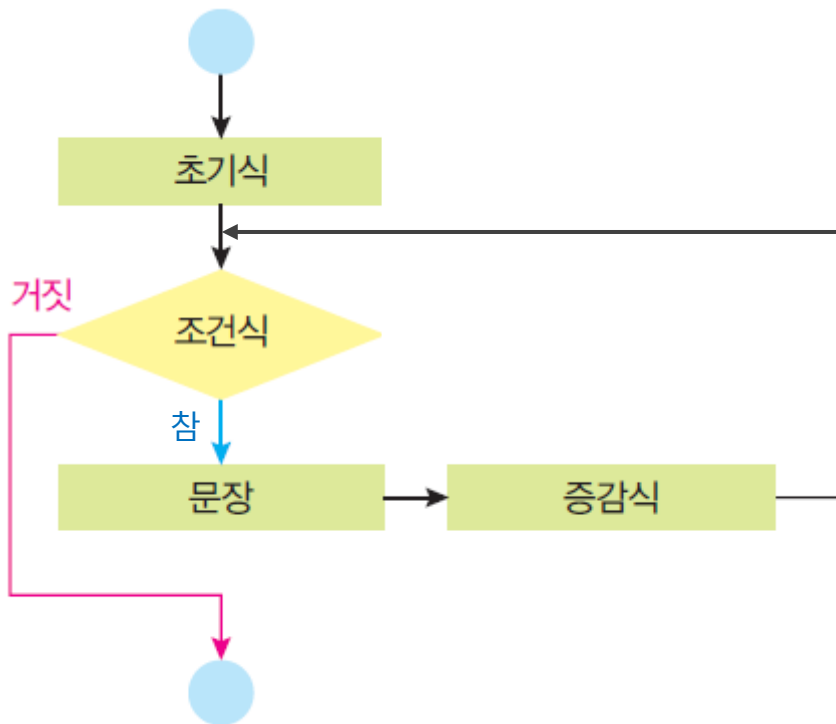
## CheckInput.java

```
01 import java.util.Scanner;
02
03 public class CheckInput {
04     public static void main(String args[]) {
05         Scanner input = new Scanner(System.in);
06         int month;
07         do {
08             System.out.print("올바른 월을 입력하시오 [1-12]: ");
09             month = input.nextInt();
10
11         } while (month < 1 || month > 12);
12
13     System.out.println("사용자가 입력한 월은 " + month);
14 }
```

```
올바른 월을 입력하시오 [1-12]: 13
올바른 월을 입력하시오 [1-12]: 14
올바른 월을 입력하시오 [1-12]: 0
올바른 월을 입력하시오 [1-12]: 1
사용자가 입력한 월은 1
```

# for 문

- 정해진 횟수만큼 반복하는 구조



# 예: 0부터 4까지 출력

## ForExample1

```
01 public class ForExample1 {  
02     public static void main(String[] args) {  
03         for (int i = 0; i < 5; i++) {  
04             System.out.println("i의 값은: " + i);  
05         }  
06     }  
07 }
```

for 문 안에서 제어 변수를 선언할 수 있다.

```
i의 값은: 0  
i의 값은: 1  
i의 값은: 2  
i의 값은: 3  
i의 값은: 4
```

# 중첩 for 문

```
for (int y = 0; y < 5; y++) {  
    for (int x = 0; x < 10; x++)  
        System.out.print("*");  
    System.out.println( );  
}
```

실행결과

```
*****  
*****  
*****  
*****  
*****
```



# break와 continue

- 반복문에서 제어의 흐름을 변경할 수 있는 방법
  - break : 반복 루프를 종료시킴
  - continue : 현재의 반복을 종료하고 다음 반복을 다시 시작시킴



break 문, continue 문은 남용하지 말 것.  
꼭 필요한 경우에만 사용하도록 하자.

# break 문

```
int total = 0;  
int count = 0;  
Scanner scan = new Scanner(System.in);
```

```
while (true) {  
    System.out.print("점수 입력: ");  
    int grade = scan.nextInt();  
    if (grade < 0)  
        break;  
    total += grade;  
    count++;  
}
```

```
System.out.println("평균은 " + total / count);
```

실행결과

```
점수 입력: 10  
점수 입력: 20  
점수 입력: -1  
평균은 15
```

# break 문

outer\_loop:

```
while (true) {
```

```
    while (true) {
```

```
        System.out.print("점수 입력: ");
```

```
        int grade = scan.nextInt();
```

```
        if (grade < 0)
```

```
            break outer_loop;
```

```
        ...
```

```
    }
```

```
}
```

레이블이 있는 break 문

# continue 문

// 문장에서 n 이 나오는 회수를 세는 코드

```
String s = "no news is good news";
```

```
int count = 0;
```

```
for (int i = 0; i < s.length(); i++) {
```

```
    if(s.charAt(i) != 'n')
```

```
        continue;
```

```
    count++; // n이 나온 회수를 하나 증가한다.
```

```
}
```

```
System.out.println("n의 개수 " + count);
```

실행결과

n의 개수 3