

Java CobinCabin iRaCha 제어문

제어문

제어문 종류

- 제어문을 쓰면 프로그램의 흐름을 바꿀 수 있다.
- 제어문의 종류

조건(선택)문	if
	switch
반복문	for
	while
	do~while
분기문	break, continue
	goto

조건문

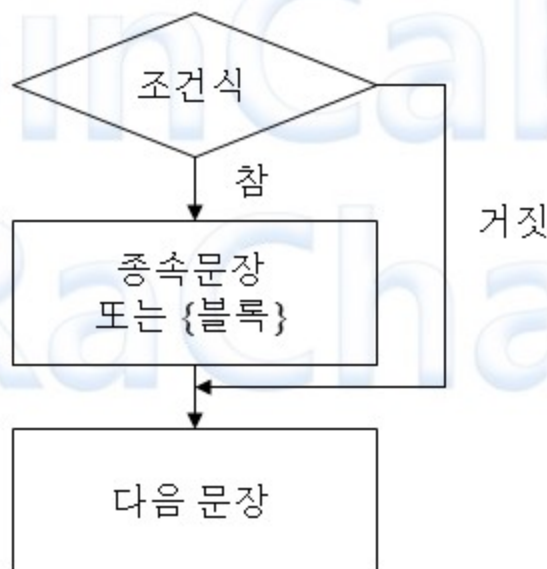
if 문

- 단순 if문
 - 조건식을 판별하여 조건이 참이면 중괄호({})안의 명령문을 수행
 - 거짓이면 중괄호({})안의 명령문을 점프한 후 다음 명령을 수행
 - 조건에 따라 수행해야 할 명령문이 한 개일 경우 중괄호는 생략 가능하다.

- 기본 형식

```
if(조건){
    명령문 1;
    명령문 2;
    ...
    명령문 n;
}
```

다음명령;



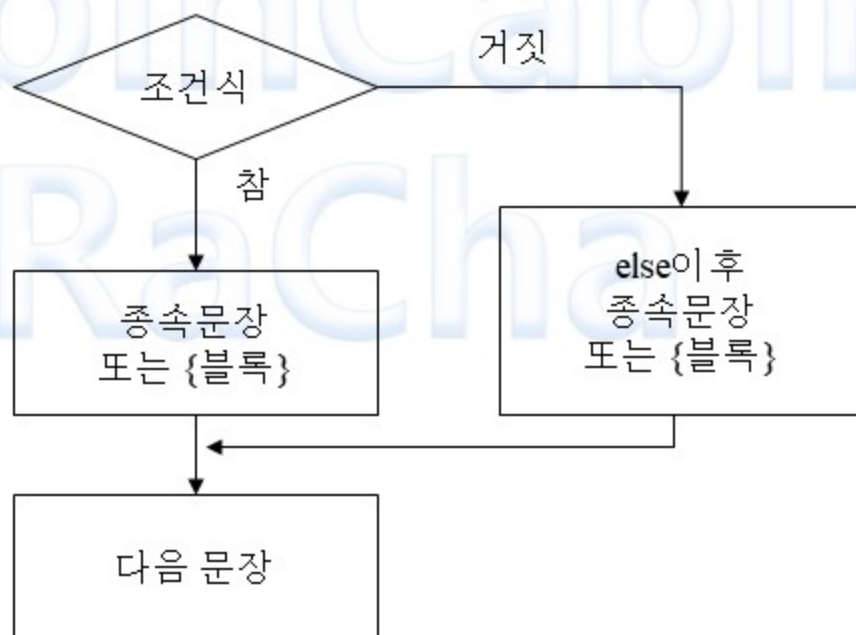
조건문

if ~ else 문

- 둘 중 한 개를 선택하는 조건문으로 조건이 참인 경우와 거짓인 경우에 수행하는 명령이 다르다.
 - if문의 조건을 만족하지 못하면 else문 수행
 - else 다음에는 조건을 쓸 수 없다.
- 조건에 따라 수행해야 할 명령문이 한 개일 경우에는 중괄호는 생략 가능

기본 형식

```
if(조건){
    명령문1;
}
else{
    명령문2;
}
```



조건문

if ~ else 문

- System.currentTimeMillis()
 - 1970년 1월 1일 0시 0분 0초 부터 현재까지의 시간을 1ms 단위로 반환
- 사용법
 - long a = System.currentTimeMillis();
 - int p_year = (int)(System.currentTimeMillis()/1000/60/60/24/365) + 1970;

CobinCabin
iRaCha

조건문

형식화 클래스

- 형식화 클래스
 - 숫자, 날짜, 텍스트 데이터를 일정한 형식에 맞게 표현 할 수 있다

CobinCabin
iRaCha

조건문

형식화 클래스

- DecimalFormat

0	10진수, 빈자리는 0으로 채운다.	0 0.0 000.000
#	10진수, 빈자리는 채우지 않는다.	# #.## #####.###
.	소수점 표시	###.#####.00 00.00
,	단위 구분 기호 표시	#,###.#
+, -	음수, 양수 표시	+###.## -###.##
E	지수 문자	
;	양수와 음수 패턴을 모두 사용할 경우 패턴 구분자	+###.##;-###.##
%	100을 곱하고 %를 붙인다.	#####%
\u00A4	통화 표시 \을 붙인다.	\u00A4#####.##

조건문

형식화 클래스

- SimpleDateFormat
 - DateFormat은 추상클래스로 SimpleDateFormat의 조상
 - 인스턴스를 생성하기 위해서 static getDateInstance()메서드를 이용해야 한다
 - Date today = new Date();
 - SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");
 - String str = df.format(today);

CobinCabin
iRaCha

조건문

형식화 클래스

- SimpleDateFormat : Parttern syntax

G	Era designator (before christ, after christ)
y	Year (e.g. 12 or 2012). Use either yy or yyyy.
M	Month in year. Number of M's determine length of format (e.g. MM, MMM or MMMMM)
d	Day in month. Number of d's determine length of format (e.g. d or dd)
h	Hour of day, 1-12 (AM / PM) (normally hh)
H	Hour of day, 0-23 (normally HH)
m	Minute in hour, 0-59 (normally mm)
s	Second in minute, 0-59 (normally ss)
S	Millisecond in second, 0-999 (normally SSS)
E	Day in week (e.g Monday, Tuesday etc.)
D	Day in year (1-366)
F	Day of week in month (e.g. 1st Thursday of December)
w	Week in year (1-53)
W	Week in month (0-5)
a	AM / PM marker
k	Hour in day (1-24, unlike HH's 0-23)
K	Hour in day, AM / PM (0-11)
z	Time Zone
'	Escape for text delimiter
'	Single quote

조건문

형식화 클래스

- SimpleDateFormat : Pattern Examples

Pattern	Example
dd-MM-yy	31-01-12
dd-MM-yyyy	31-01-2012
MM-dd-yyyy	01-31-2012
yyyy-MM-dd	2012-01-31
yyyy-MM-dd HH:mm:ss	2012-01-31 23:59:59
yyyy-MM-dd HH:mm:ss.SSS	2012-01-31 23:59:59.999
yyyy-MM-dd HH:mm:ss.SSSZ	2012-01-31 23:59:59.999+0100
EEEE MMMMM yyyy HH:mm:ss.SSSZ	Saturday November 2012 10:45:42.720+0100

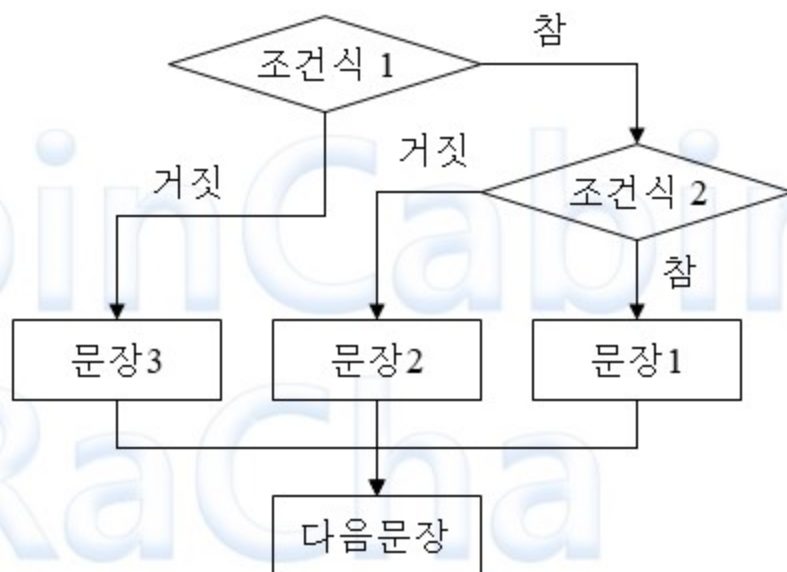
조건문

중첩 if 문

- 기본 형식

```

if(조건 1){
    명령문 1;
    if(조건 2){
        명령문 2;
    }
    else{
        명령문 3;
    }
}
else{
    명령문 4;
}
  
```



조건문

중첩 if 문

- Math.random()
 - 0.0000 ~ 0.9999 까지의 임의의 실수를 구하는 메소드
 - System.out.println(Math.random());

CobinCabin
iRaCha

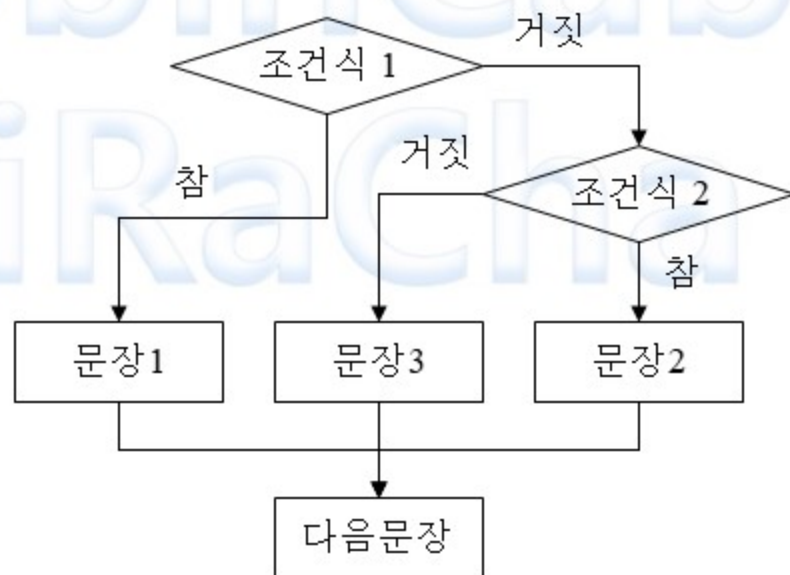
조건문

다중 if 문

- 여러 개 중 한 개를 선택하는 경우 사용하는 조건문
 - 조건에 따라 수행되는 명령문이 다르다.
- 조건이 참이 되어 명령을 수행하게 되면 아래 있는 나머지의 조건은 비교하지 않고 다중 if문을 전체를 빠져 나온다.
- 조건에 따라 수행해야 할 명령문이 한 개일 경우에는 중괄호는 생략 가능하다.

```

기본 형식
if(조건 1){
    명령문 1;
}
else if(조건 2){
    명령문 2;
}
else if(조건 3){
    명령문 3;
}
else{
    명령문 4;
}
  
```



조건문

switch 문

- 변수의 값을 **case**에서 찾아 명령을 수행하는 조건문
 - 값이 없을 경우에는 **default**의 명령문을 수행
 - **case**와 명령문 사이에 콜론(:)을 써야 한다
 - **case**의 값은 **byte, short, char, int, String** 중에 하나의 상수만 사용
 - 문과 식이 올 수 있지만 결과는 상수여야 한다

조건문

switch 문

- 기본 형식

```
switch(변수){  
    case 상수:  
        명령문;  
        break;  
    case 상수:  
        명령문;  
        break;  
    ...  
    case 상수:  
        명령문;  
        break;  
    default:  
        명령문;  
}
```

CobinCabin
iRaCha

Java CobinCabin

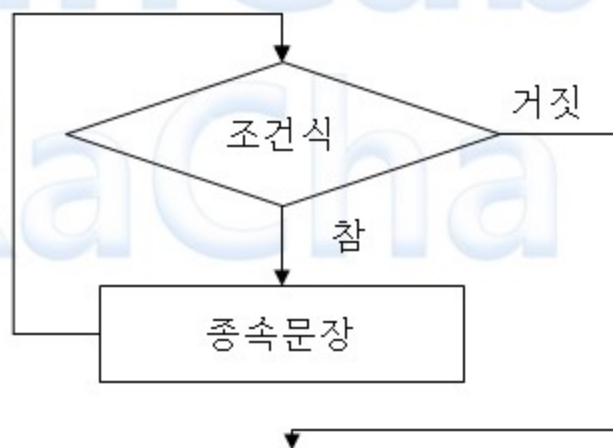
iRaCha 반복 제어문

while 문

while 문

- 조건이 참인 경우 중괄호({})안의 명령문을 반복적으로 수행하는 제어문
- 수행해야 할 명령문이 한 개일 경우에는 중괄호는 생략 가능하다.
- 조건이 참인 경우 명령문을 반복 수행한다.
- 기본형식

```
while(조건){
    명령문1;
    명령문2;
    ...
    명령문n;
}
```



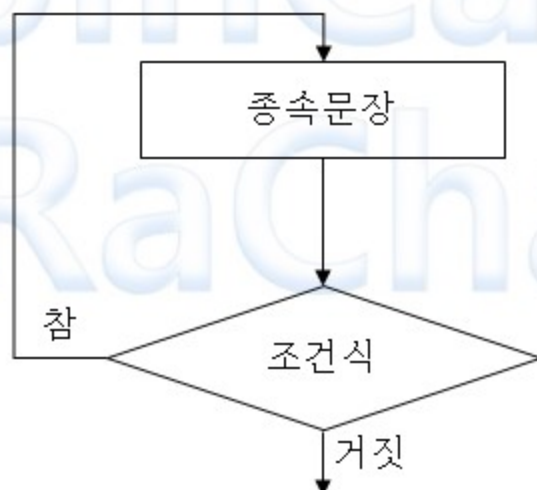
while 문

do~while 문

- while문과 비슷한 형태의 반복문이지만, 명령문을 먼저 실행하고 조건을 비교한다는 점이 다르다.
 - do~while문의 명령문은 최소 한번은 반드시 수행
 - do~while문의 while문 뒤에 세미콜론(;)을 써야 한다
 - 조건에 따라 수행해야 할 명령문이 한 개일 경우에는 중괄호는 생략 가능하다.

기본 형식

```
do{
    명령문1;
    명령문2;
    ...
    명령문n;
}while(조건);
```



while 문

기타 제어문

- break 문
 - 반복 명령의 실행 도중 강제적으로 반복문을 빠져나올 때 사용되는 명령이다.
 - 여러 개의 반복문이 중첩되어 있는 경우 반복문 한 개만 빠져나온다.
- continue 문
 - 반복문의 나머지 부분을 무시하고 그 반복문의 선두로 이동할 때 사용
 - 여러 개의 반복문이 중첩되어 있는 경우 **continue** 명령이 있는 반복문의 선두로 이동한다.

CobinCabin
iRaCha

for 문

for 문

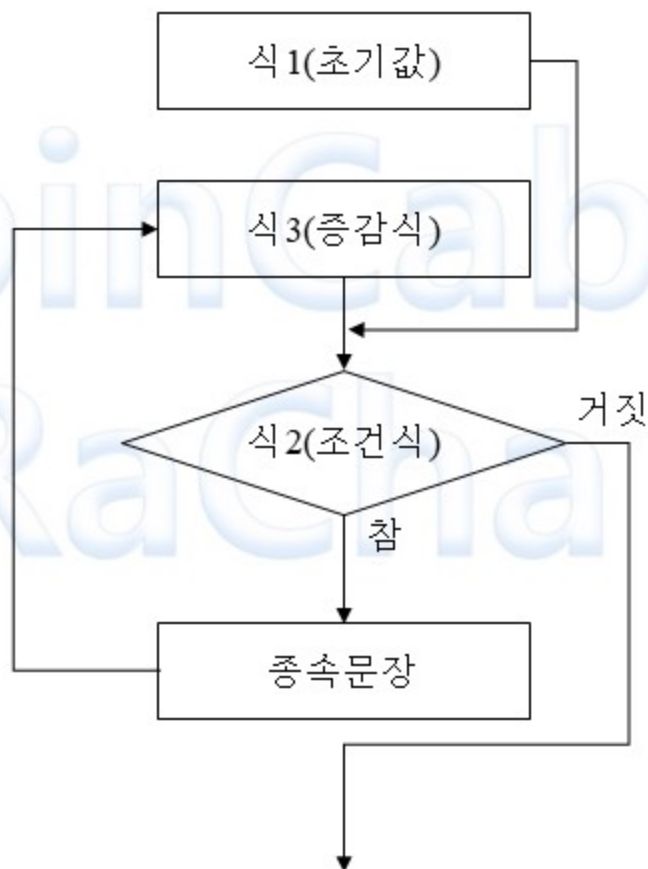
- 조건이 참인 경우 중괄호({})안의 명령문을 반복적으로 수행하는 제어문
- for문의 괄호 안에는 3개의 명령을 쓸 수 있는데 이 명령은 세미콜론(;)으로 분리
 - 3개의 명령 중 필요 없는 명령은 생략 가능
- for문내에서 초기 값과 증/감 값이 여러 개인 경우 콤마(,)로 나열
- 조건이 여러 개인 경우는 &&(AND) 또는 ||(OR) 연산자로 조건을 연결해야 한다.
 - 콤마로 나열 시 가장 우측에 있는 조건만 수행
- 조건이 참인 경우 명령문을 반복 수행한다.
 - 초기 값은 반복 수행되지 않는다.

for 문

사용 예

- 기본 형식

```
for(초기값; 조건; 증/감값){
    명령문1;
    명령문2;
    ...
    명령문n;
}
```



for 문

이중 for문

- for문 안에 for문이 들어간 형태로 내부의 명령을 외부의 횟수만큼 반복하게 된다.

- 이중 for문의 사용 예

```
for(초기값; 조건; 증/감값){
    for(초기값; 조건; 증/감값){
        명령문1;
        명령문2;
        ...
        명령문n;
    }
}
```

- 내부 명령은 외부의 반복 횟수만큼 반복 수행하게 된다.