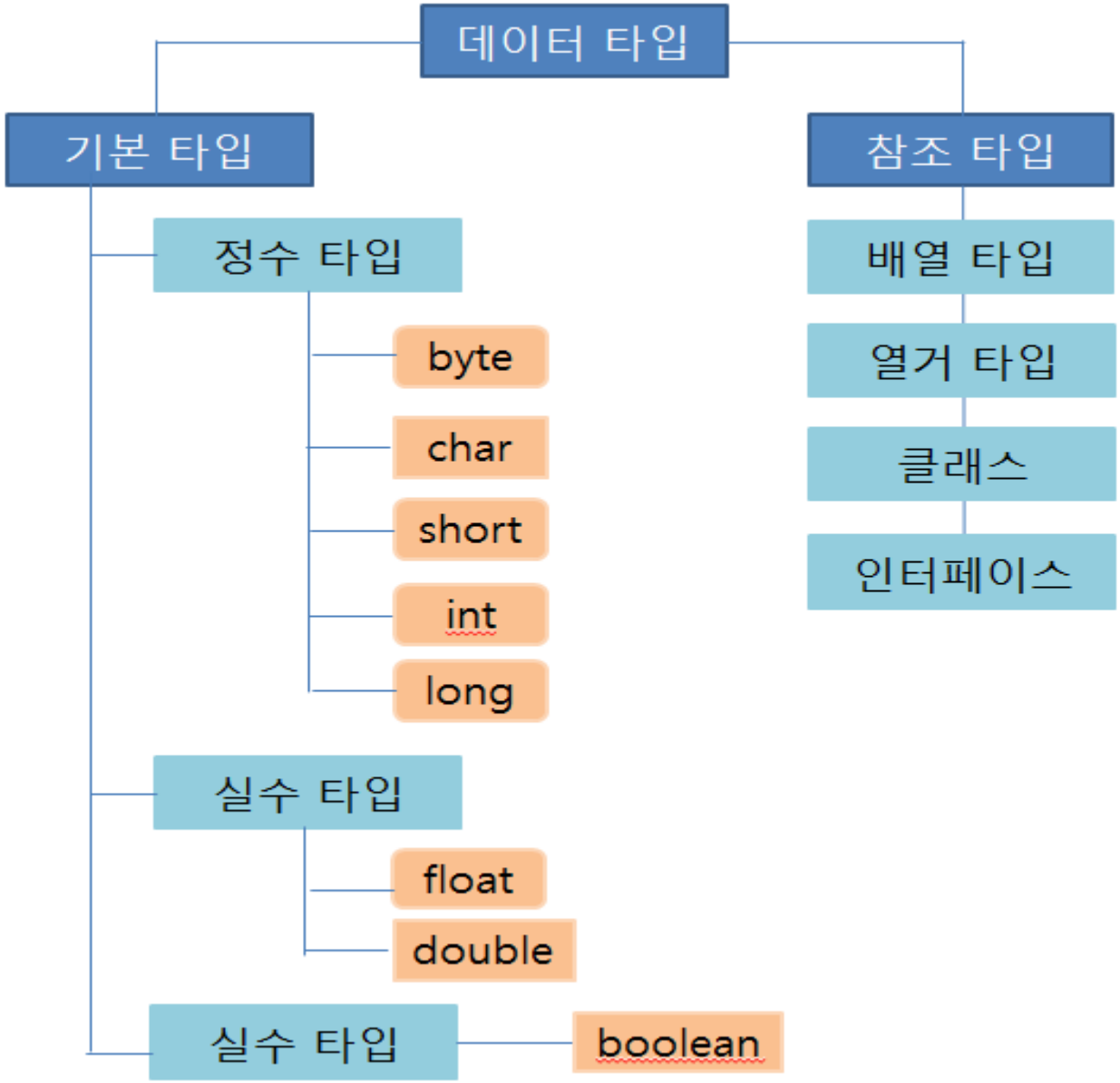
자바 1~2주차 최종

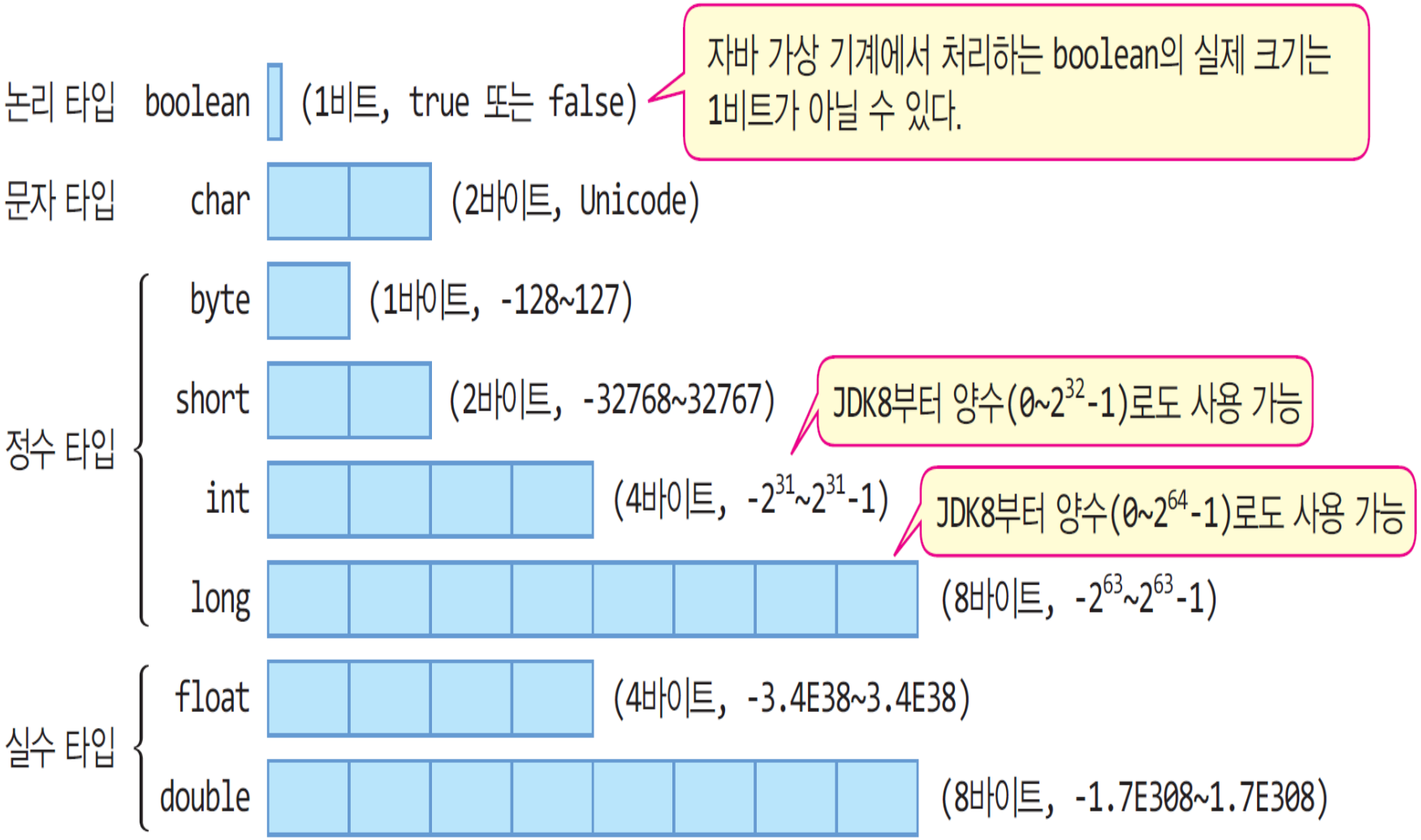
자바 프로그램의 키워드 용도

기본데이터형	byte, short, char, int, float, double, void
흐름제어문	if, else, do, while, for, break, continue, return, case, switch, default
영역 결정	public, protected, private
특성	abstract, final, synchronized, native, transient, volatile
예외처리	throws, try, throw, catch, finally
상하위 클래스 관련	this, super
패키지 관련	package, import
객체 생성	new
데이터형 관련	instanceof

데이터형


데이터 타입
기본 타입
참조 타입
정수 타입
byte
char
short
int
long
실수 타입
float
double
실수 타입
boolean
배열 타입
열거 타입
클래스
인터페이스

데이터형


논리 타입 boolean (1비트, true 또는 false)
자바 가상 기계에서 처리하는 boolean의 실제 크기는 1비트가 아닐 수 있다.
문자 타입 char (2바이트, Unicode)
정수 타입
byte (1바이트, -128~127)
short (2바이트, -32768~32767)
JDK8부터 양수(0~2^32-1)로도 사용 가능
int (4바이트, -2^31~2^31-1)
JDK8부터 양수(0~2^64-1)로도 사용 가능
long (8바이트, -2^63~2^63-1)
실수 타입
float (4바이트, -3.4E38~3.4E38)
double (8바이트, -1.7E308~1.7E308)

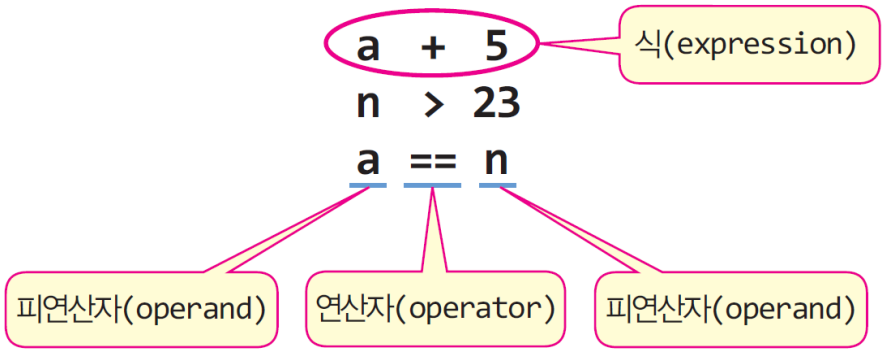
- 메모리의 동작
  - 메모리 관리는 자바 가상 머신이 수행
  - 하지만, 내부적으로 메모리 동작을 이해해야지만 좋은 프로그래밍이 가능함

```
int x = 10;
int y = x;
x = x * 2;
String str = "메모리 사용";
```

연산자

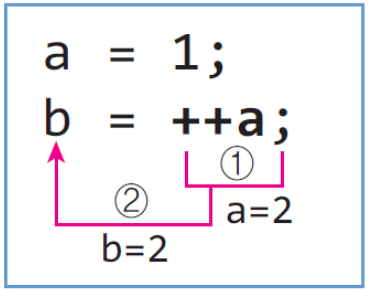
연산자종류	연산자	사용법	설명
산술연산자	+	a+b	a와 b의 합
	-	a-b	a와 b의 차
	*	a*b	a와 b의 곱
	/	a/b	a를 b로 나눈 몫
	%	a%b	a를 b로 나눈 나머지
대입연산자	+=	a+=b	a= a+b
	-=	a-=b	a= a-b
	*=	a*=b	a = a*b
	/=	a/=b	a= a/b
	%=	a%=b	a= a%b
	&=	a&=b	a= a&b
	|=	a|=b	a= a|b
	^=	a^=b	a= a^b
증감연산자	++	i++	i=i+1
	--	i--	i=i-1
비교연산자	>	a > b	a가 b보다 큰 경우 true
	>=	a>=b	a가 b보다 크거나 같을 경우 true
	<	a<b	a가 b보다 작을 경우 true
	<=	a<=b	a가 b보다 작거나 같을 경우 true
	==	a==b	a와 b가 같을 경우 true
	!=	a!=b	a와 b가 같지 않을 경우 true
비트연산자	&	a&b	비트단위의 AND
	|	a|b	비트단위의 OR
	^	a^b	비트단위의 XOR
	~	a~b	비트단위의 보수
	>>	a>>b	a를 b만큼 오른쪽으로 이동 (빈 자리는 양수는 0 음수는 1)
	>>>	a>>>b	a를 b만큼 오른쪽으로 이동 (빈자리는 항상 0)
	<<	a<<b	a를 b만큼 왼쪽으로 이동
논리연산자	&&	a && b	a와 b가 모두 true 인 경우 true
	||	a||b	a 또는 b가 true 인 경우 true
	!	!a	a가 true 면 false , false 면 true

연산자

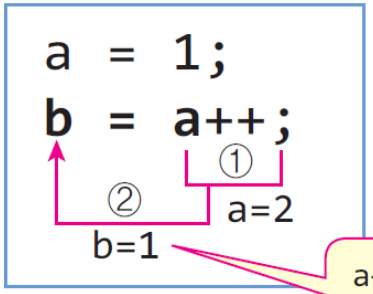


연산의 종류	연산자	연산의 종류	연산자
증감	++ --	비트	& | ^ ~
산술	+ - * / %	논리	&& || ! ^
시프트	>> << >>>	조건	? :
비교	> < >= <= == !=	대입	= *= /= += -= &= ^= |= <<= >>= >>>=

연산자



(a) 전위 연산자



(b) 후위 연산자

a++의 연산은 증가 전의 값 1을 반환한다.

연산자	내용	연산자	내용
a++	a를 1 증가하고 증가 전의 값 반환	++a	a를 1 증가하고 증가된 값 반환
a--	a를 1 감소하고 감소 전의 값 반환	--a	a를 1 감소하고 감소된 값 반환

연산자

```
int a = 1, b = 3;
a = b;          // b 값을 a에 대입하여 a=3
a += b;         // a = a + b의 연산이 이루어져, a=6. b는 3 그대로
```

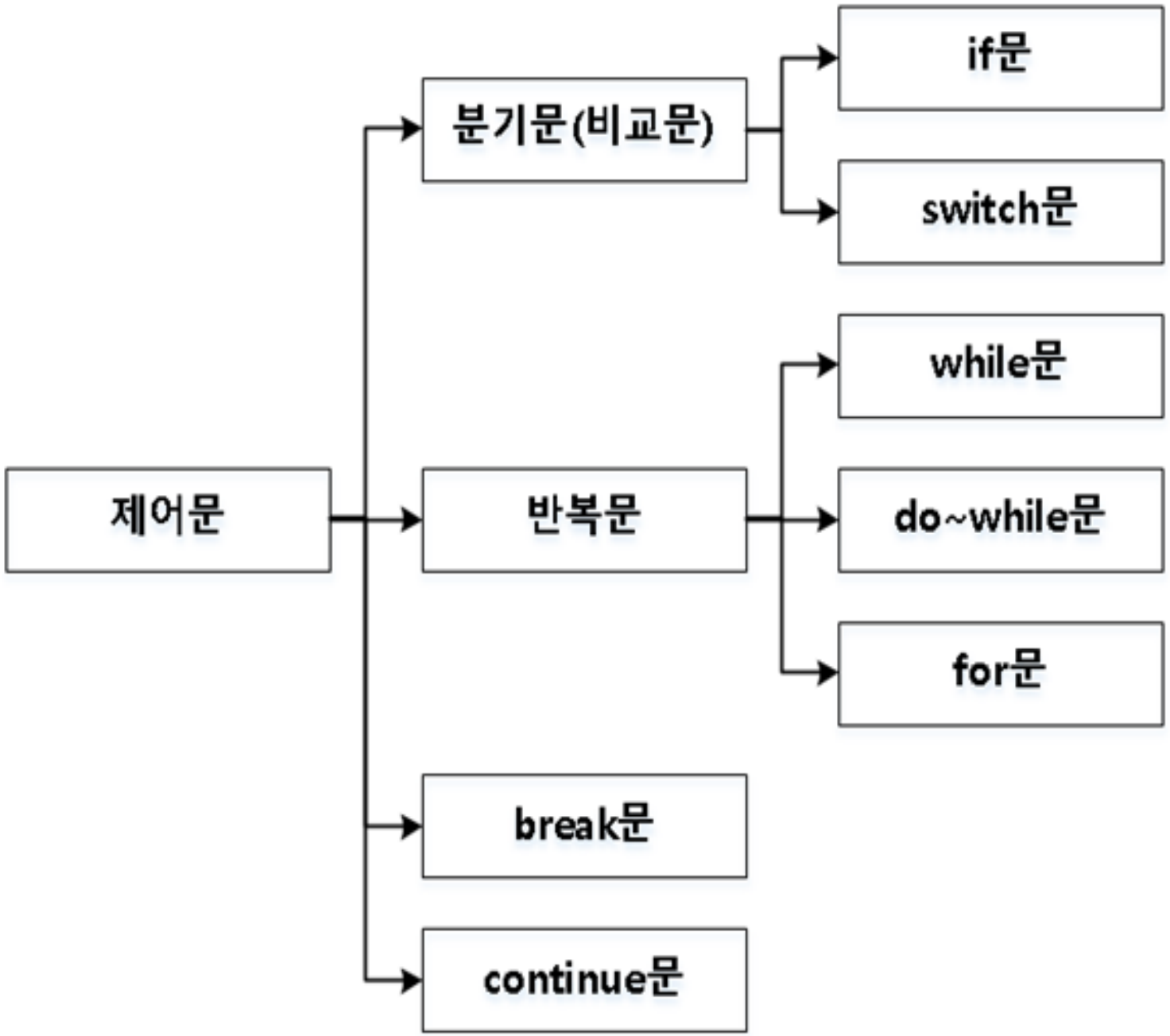
대입 연산자	내용	대입 연산자	내용
a = b	b의 값을 a에 대입	a &= b	a = a & b와 동일
a += b	a = a + b와 동일	a ^= b	a = a ^ b와 동일
a -= b	a = a - b와 동일	a |= b	a = a | b와 동일
a *= b	a = a * b와 동일	a <<= b	a = a << b와 동일
a /= b	a = a / b와 동일	a >>= b	a = a >> b와 동일
a %= b	a = a % b와 동일	a >>>= b	a = a >>> b와 동일

연산자

연산자	내용	예제	결과
a < b	a가 b보다 작으면 true	3<5	true
a > b	a가 b보다 크면 true	3>5	false
a <= b	a가 b보다 작거나 같으면 true	1<=0	false
a >= b	a가 b보다 크거나 같으면 true	10>=10	true
a == b	a가 b와 같으면 true	1==3	false
a != b	a가 b와 같지 않으면 true	1!=3	true

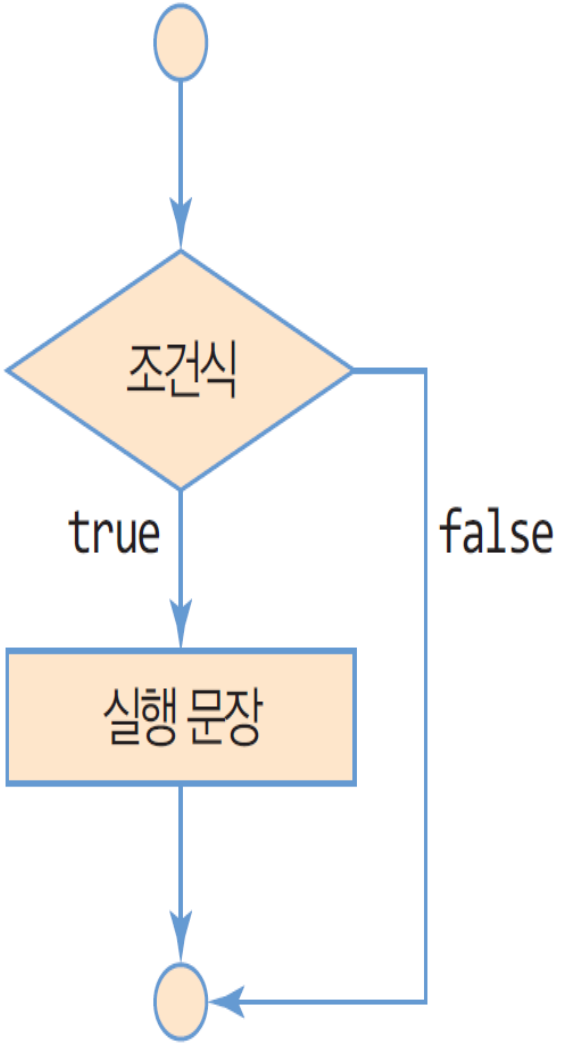
연산자	내용	예제	결과
! a	a가 true이면 false, false이면 true	!(3<5)	false
a || b	a와 b의 OR 연산. a와 b 모두 false인 경우에만 false	(3>5)||(1==1)	true
a && b	a와 b의 AND 연산. a와 b 모두 true인 경우에만 true	(3<5)&&(1==1)	true
a ^ b	a와 b의 XOR 연산. a와 b가 서로 다를 때 true	(3>5)^(1==1)	true

제어문


제어문
분기문(비교문)
if문
switch문
반복문
while문
do~while문
for문
break문
continue문

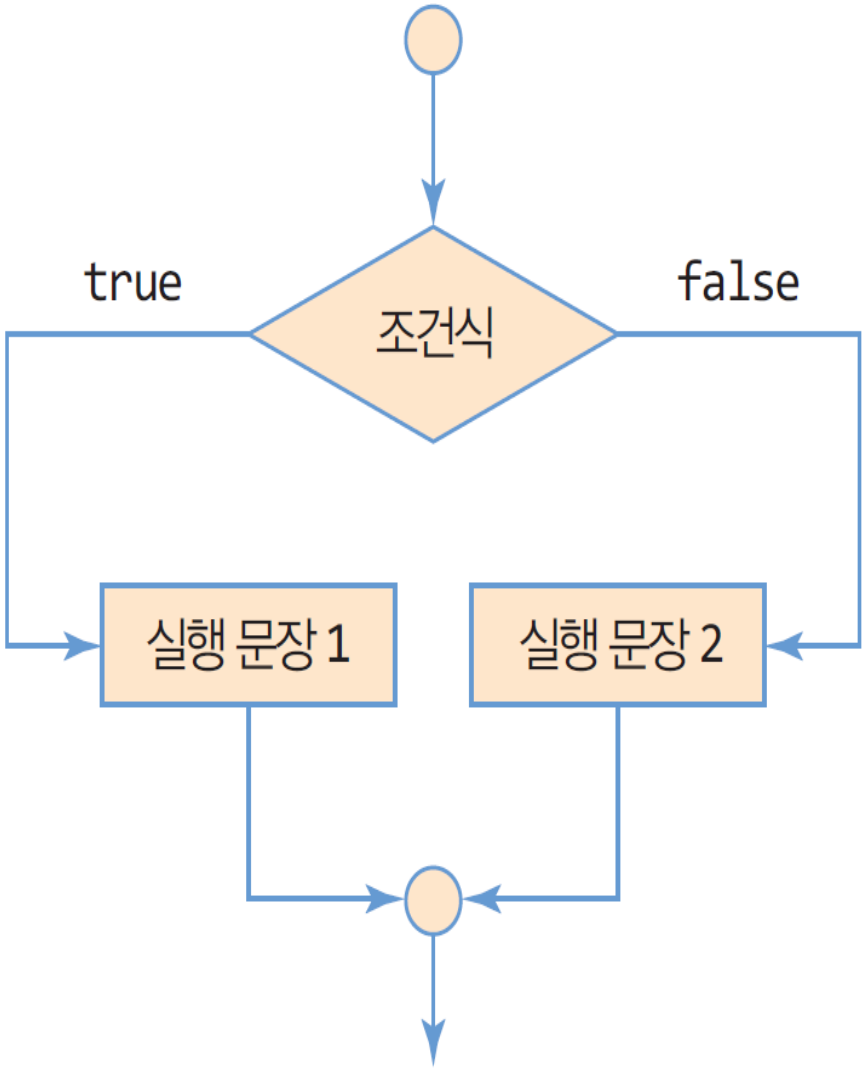
제어문

```
if (조건식) {
    ...실행 문장...  // 조건식이 참인 경우
}
```



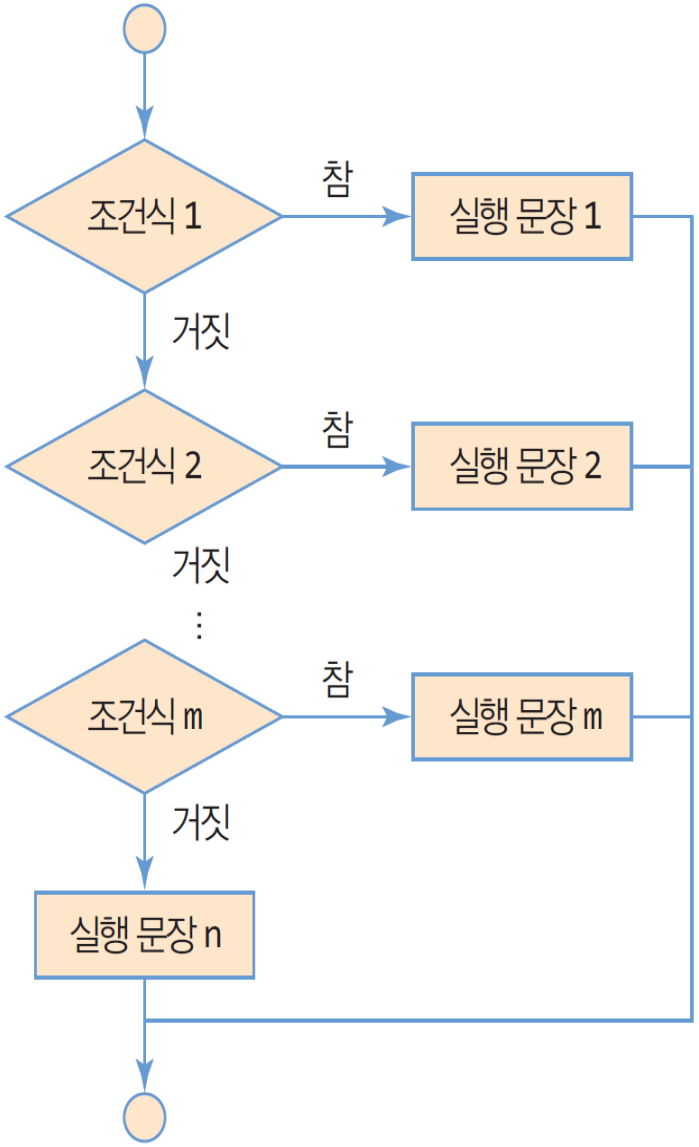
제어문

```
if (조건식) {
    ...실행 문장 1...
}
else {
    ...실행 문장 2...
}
```



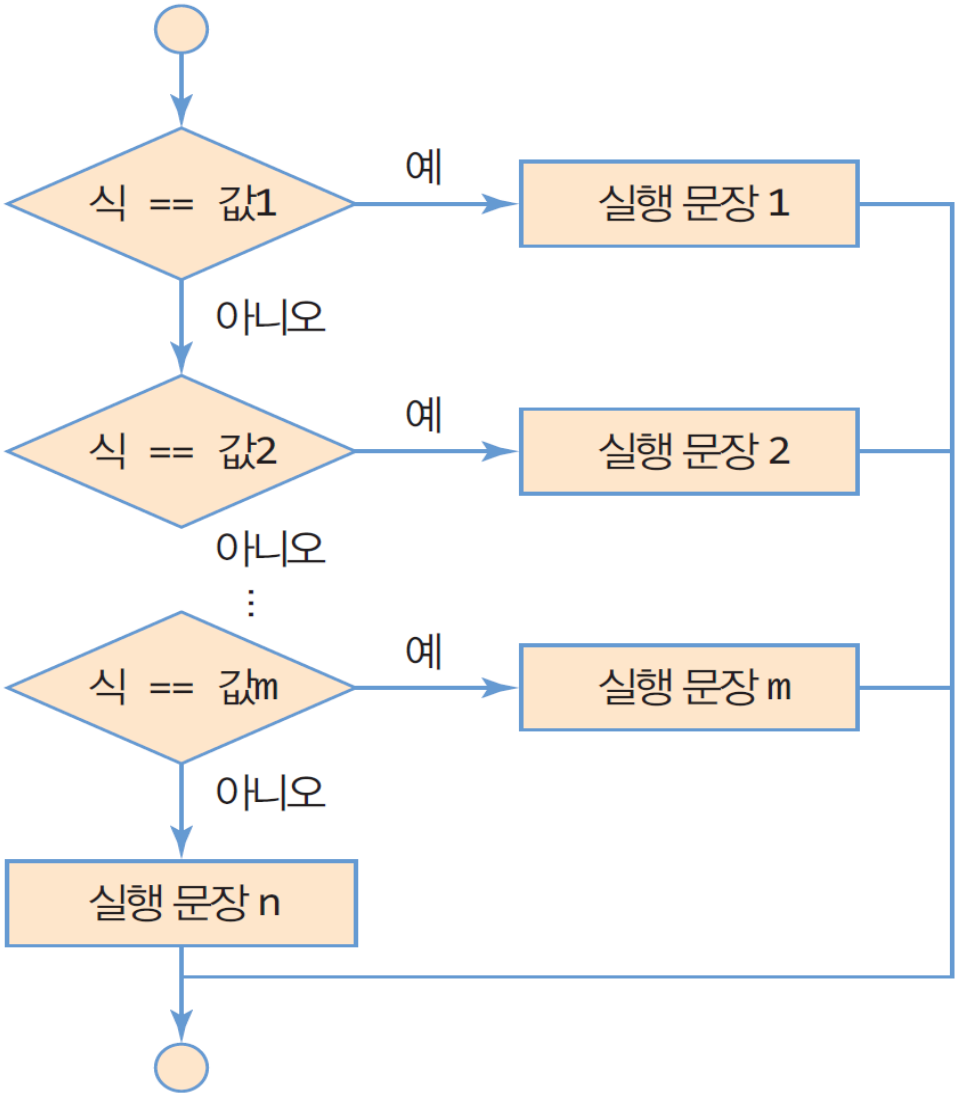
제어문

```
if (조건식 1) {
   실행 문장 1; // 조건식 1이 참인 경우
}
else if (조건식 2) {
   실행 문장 2; // 조건식 2가 참인 경우
}
else if (조건식 m) {
   ..... // 조건식 m이 참인 경우
}
else {
   실행 문장 n; // 앞의 모든 조건이 거짓인 경우
}
```

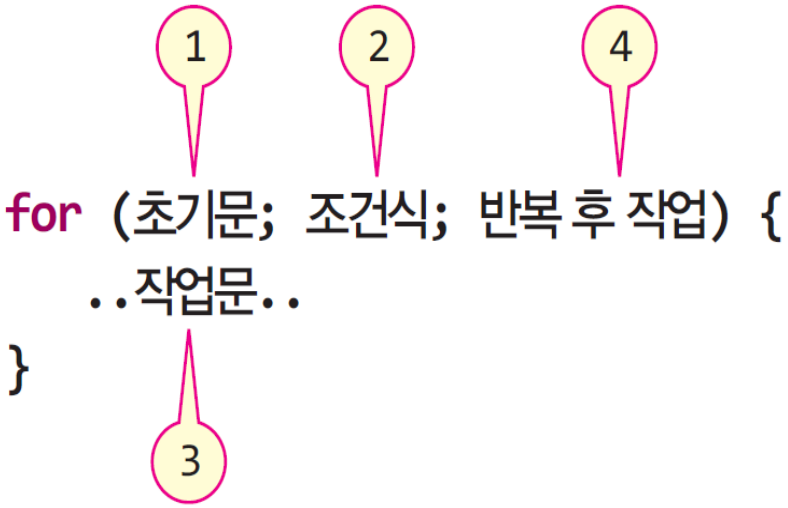


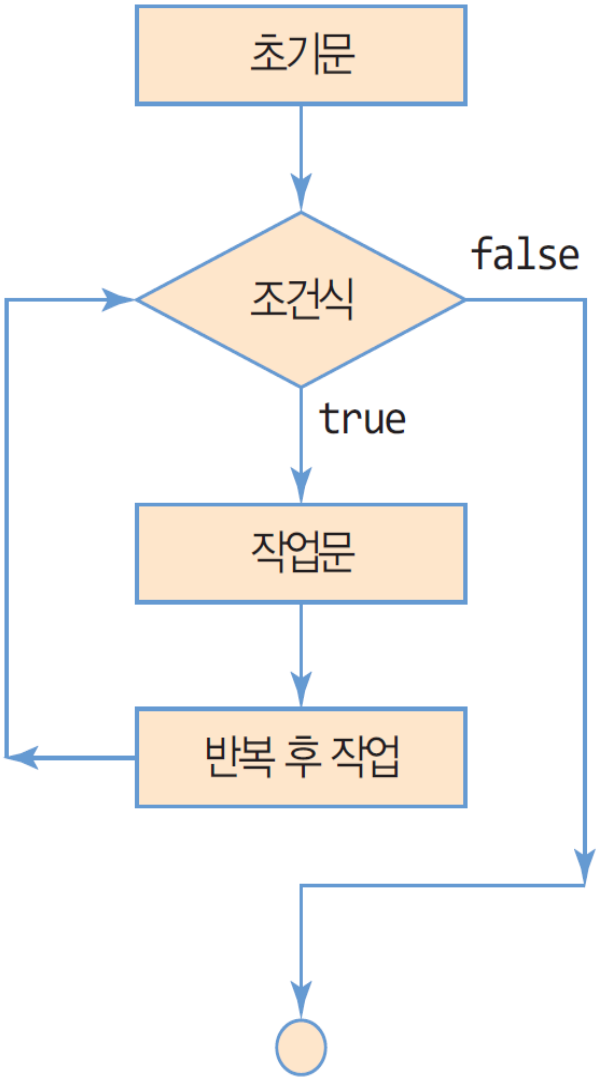
제어문

```
switch (식) {
    case 값1:
        실행 문장 1;
        break;
    case 값2:
        실행 문장 2;
        break;
    ...
    case 값m:
        실행 문장 m;
        break;
    default:
        실행 문장 n;
}
```

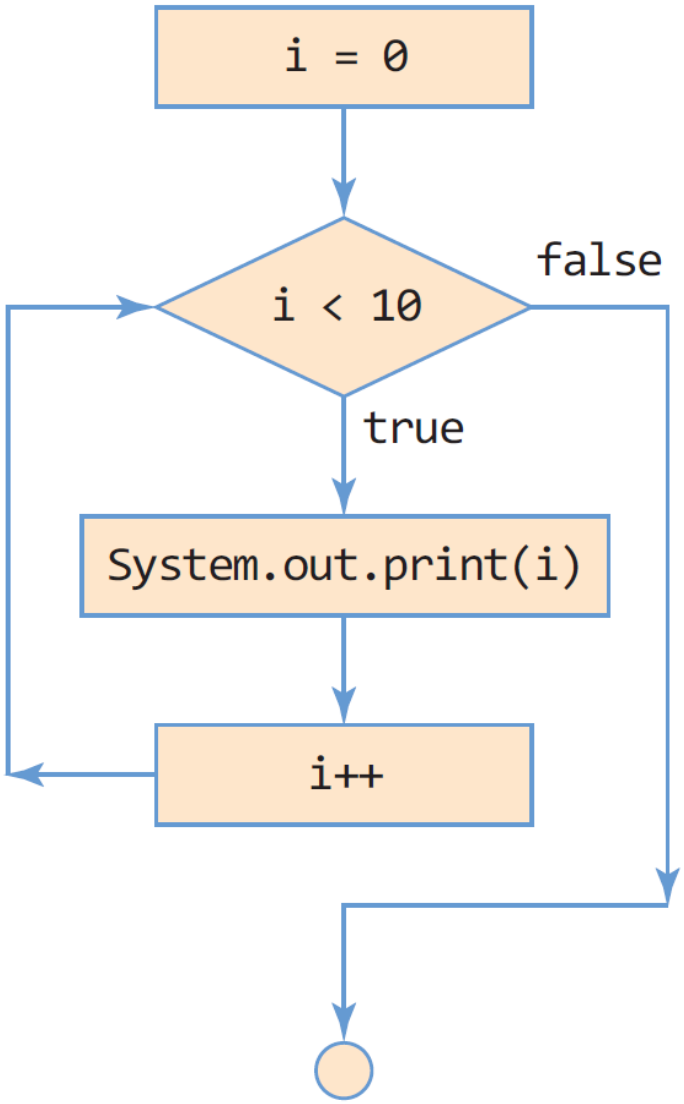


제어문





제어문




```
for(i=0; i<10; i++) {
    System.out.print(i);
}
```

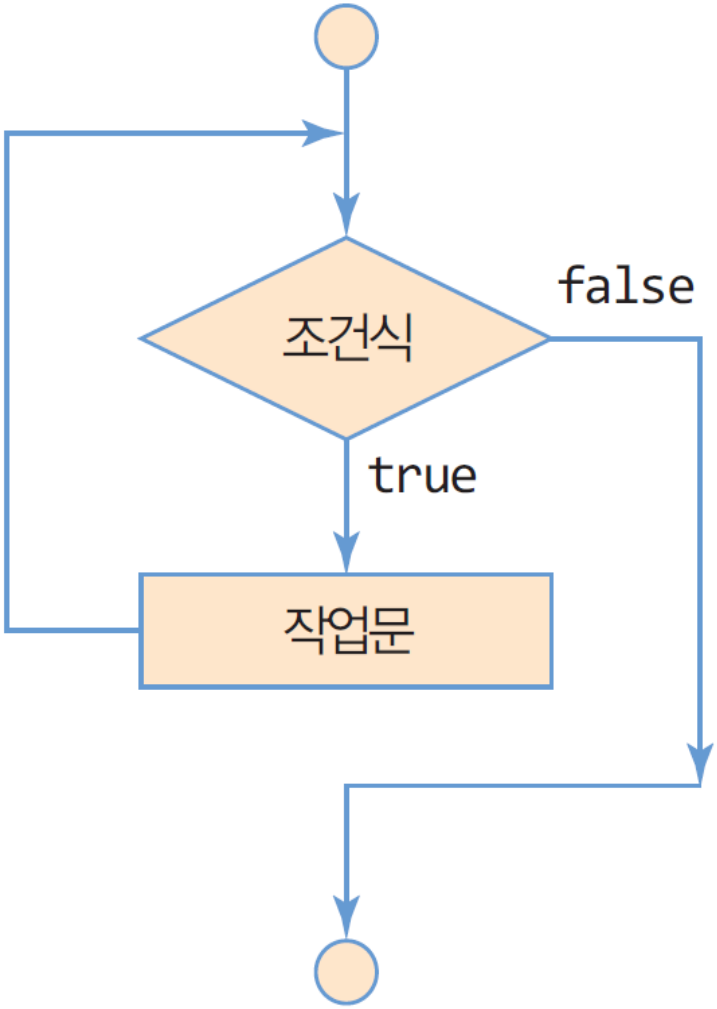
```
0123456789
```

제어문

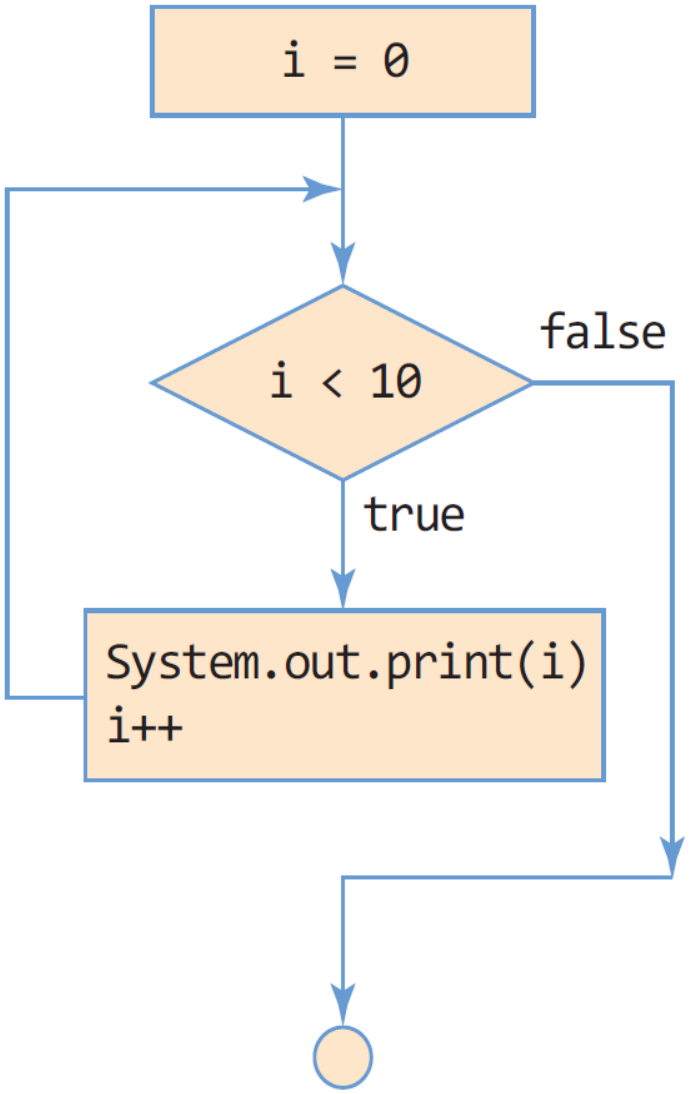
```
while (조건식) {
    ..작업문..
}
```

1

2



제어문




```
i = 0;
while(i<10) {
    System.out.print(i);
    i++;
}
```

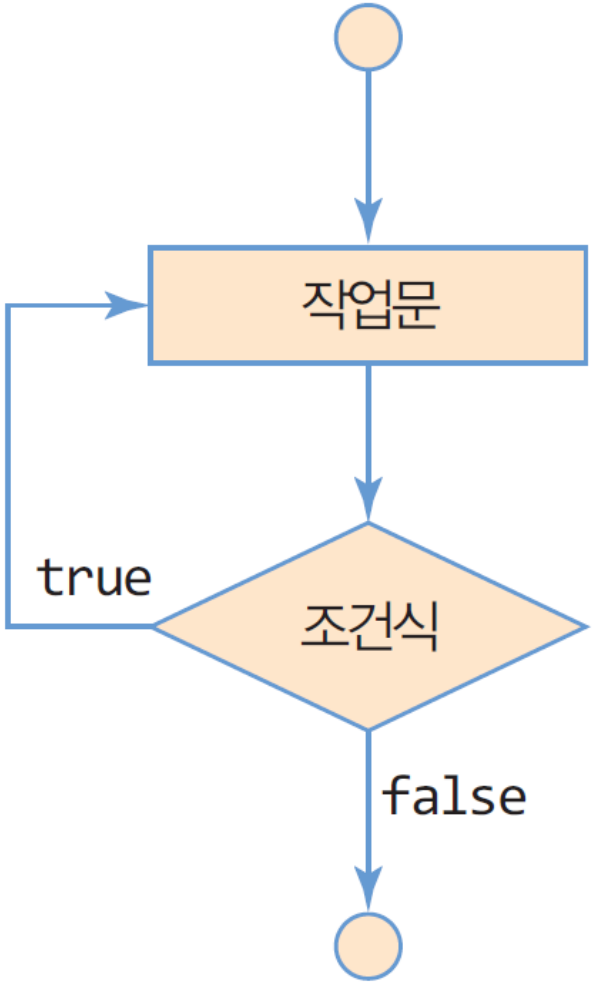
```
0123456789
```

제어문

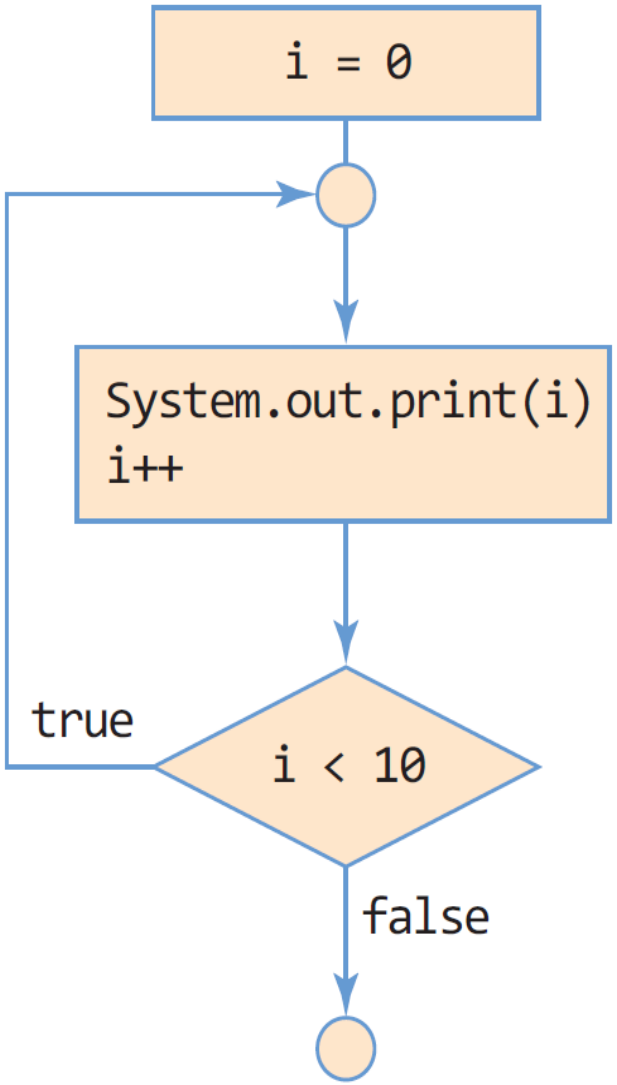
```
do {
    ..작업문..
} while(조건식);
```

1

2



제어문




```
i = 0;
do {
    System.out.print(i);
    i++;
} while(i<10);
```

```
0123456789
```

제어문

```
for(초기문; 조건식; 반복 후 작업) {
    .........
    continue;
    .........
}
```

```
while(조건식) {
    .........
    continue;
    .........
}
```

```
do {
    .........
    continue;
    .........
} while(조건식);
```

```
for(초기문; 조건식; 반복 후 작업) {
    ..................
    break;
    ..................
}
..................
```

(a) 현재 반복문 벗어나기

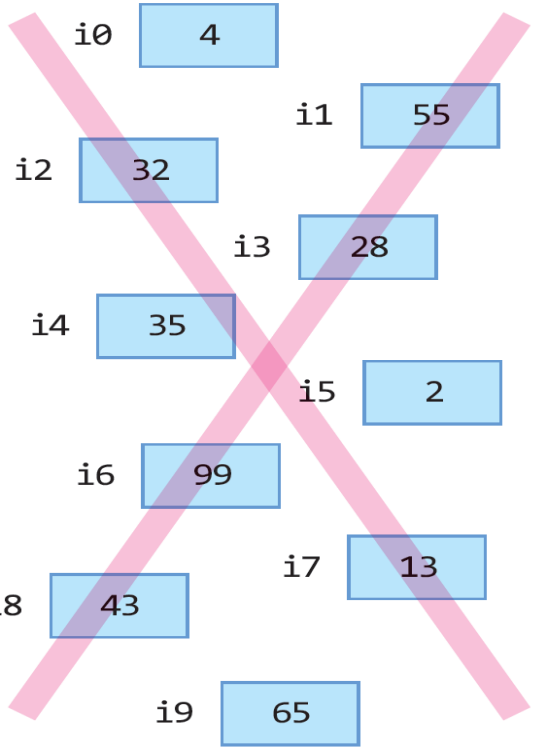
```
for(초기문; 조건식; 반복 후 작업) {
    while(조건식) {
        ..................
        break;
        ..................
    }
    ..................
}
..................
```

(b) 중첩 반복에서 안쪽 반복문만 벗어나는 경우

배열

(1) 10개의 정수형 변수를 사용하는 경우

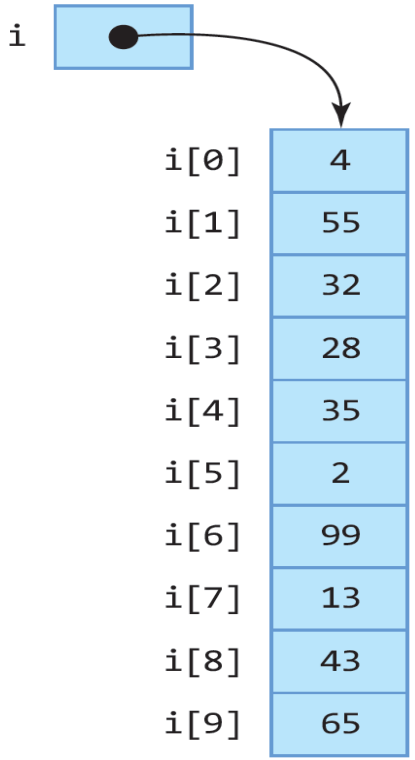
```
int i0, i1, i2, i3, i4, i5, i6, i7, i8, i9;
```




```
sum = i0+i1+i2+i3+i4+i5+i6+i7+i8+i9;
```

(2) 10개의 정수로 구성된 배열을 사용하는 경우

```
int i[] = new int[10];
```




```
for(sum=0, n=0; n<10; n++)
    sum += i[n];
```

배열

(1) 배열에 대한 레퍼런스 변수 intArray 선언

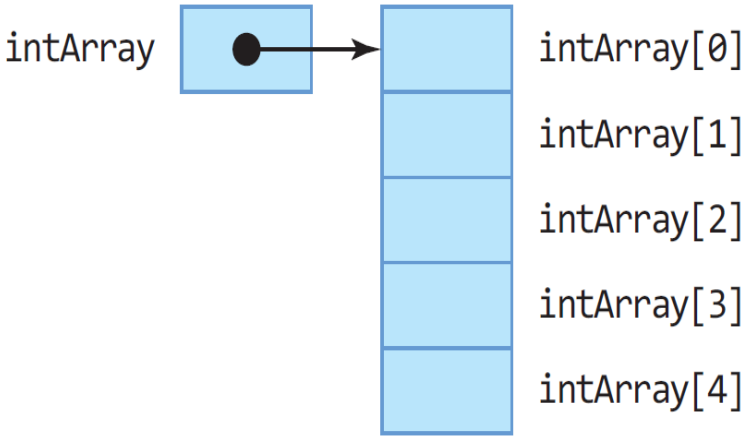
```
int intArray [];
```

배열 타입 / 배열에 대한 레퍼런스 변수 / 배열 선언

(2) 배열 생성

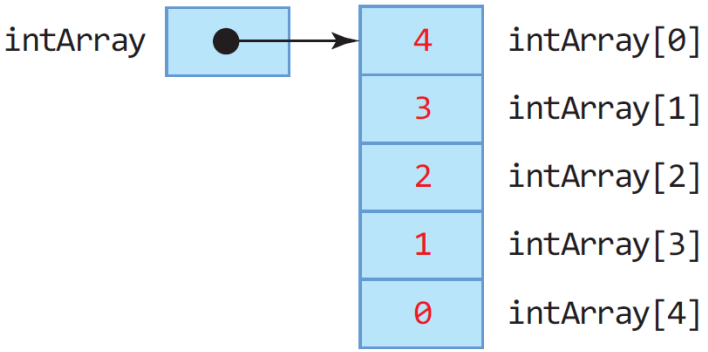
```
intArray = new int [5];
```

배열에 대한 레퍼런스 변수 / 배열 생성 / 타입 / 원소 개수

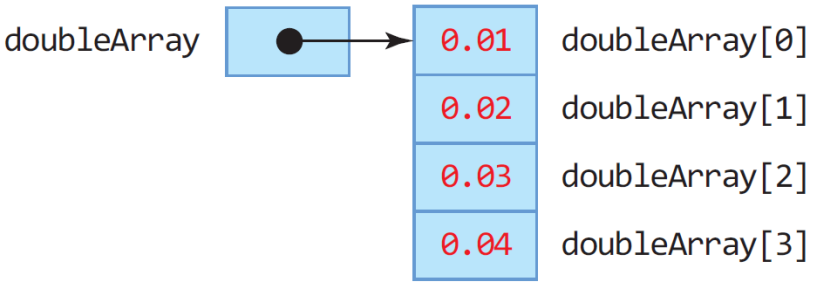


배열

```
int intArray[] = {4, 3, 2, 1, 0};
```




```
double doubleArray[] = {0.01, 0.02, 0.03, 0.04};
```



제어문

- for-each 문
  - 배열이나 나열(enumeration)의 각 원소를 순차적으로 접근하는데 유용한 for 문

```
int[] num = { 1,2,3,4,5 };
int sum = 0;
for (int k : num) // 반복될 때마다 k는 num[0], num[1], ..., num[4] 값으로 설정
    sum += k;
System.out.println("합은 " + sum);
```

합은 15

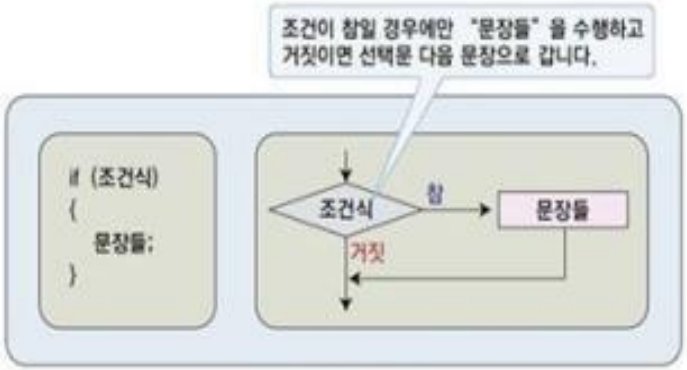
```
String names[] = { "사과", "배", "바나나", "체리", "딸기", "포도" } ;
for (String s : names) // 반복할 때마다 s는 names[0], names[1], ..., names[5] 로 설정
    System.out.print(s + " ");
```

사과 배 바나나 체리 딸기 포도

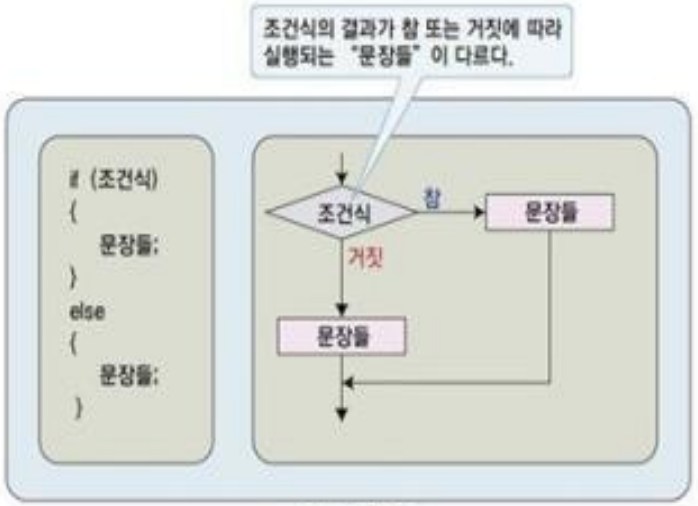
```
enum Week { 월, 화, 수, 목, 금, 토, 일 }
for (Week day : Week.values()) // 반복될 때마다 day는 월, 화, 수, 목, 금, 토, 일로 설정
    System.out.print(day + "요일 ");
```

월요일 화요일 수요일 목요일 금요일 토요일 일요일

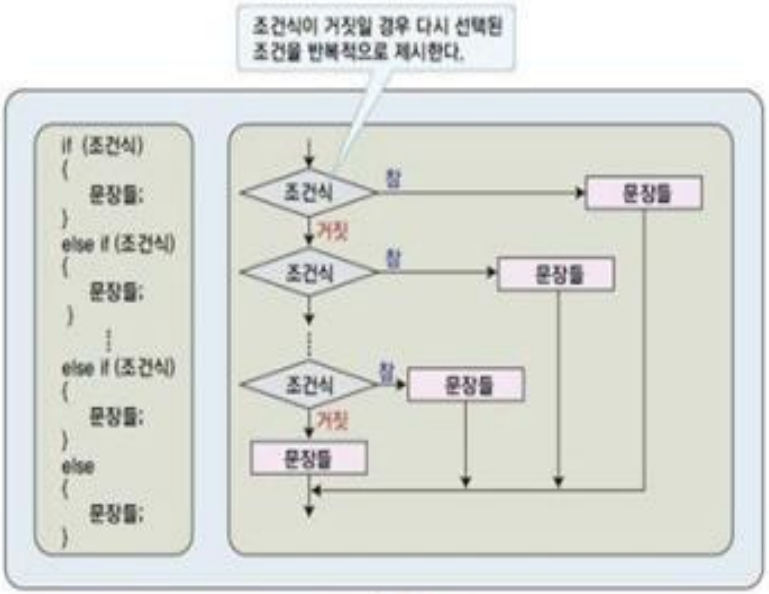
제어문



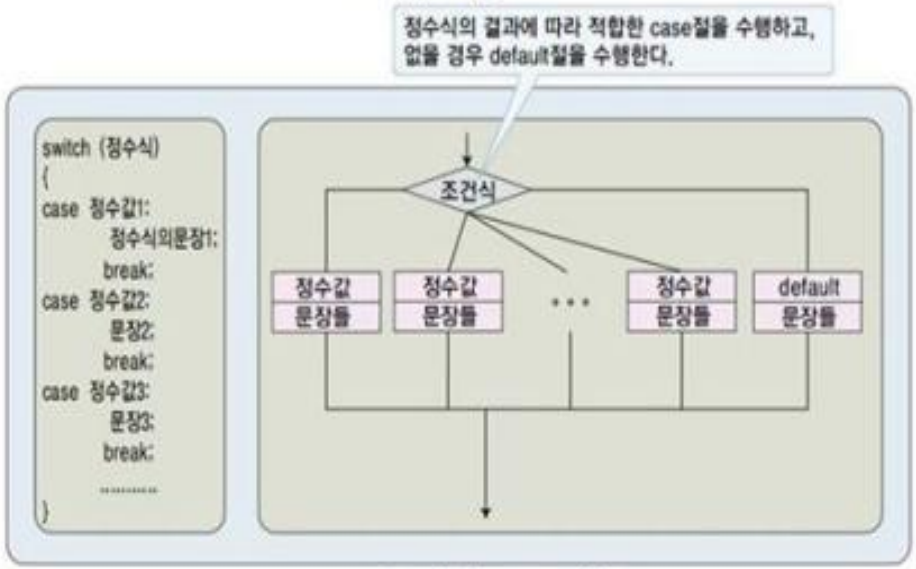
<단순 if 문>



<이중 if 문>

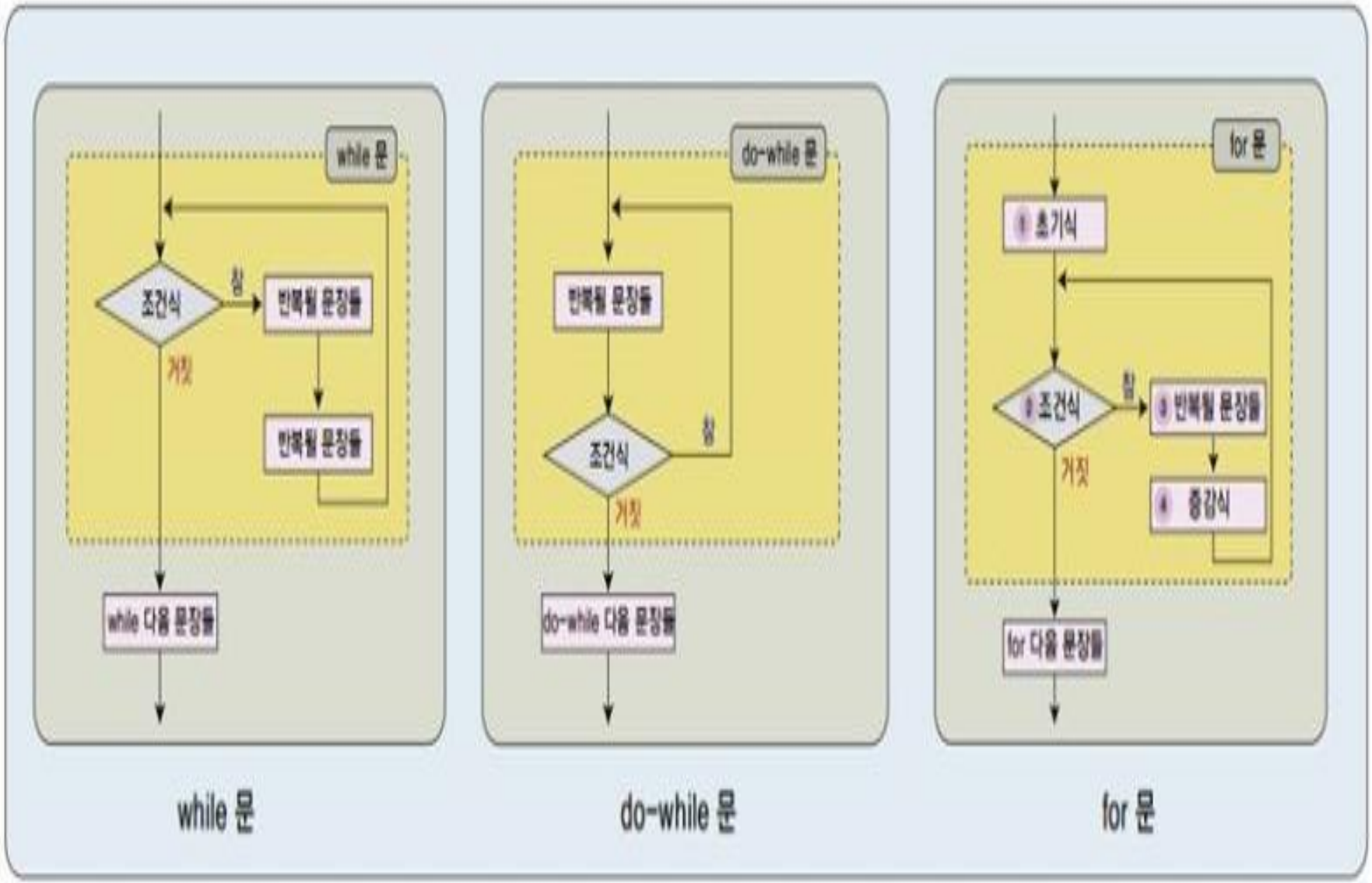


<복합 if 문>



<switch ~ case 문>

제어문


while 문
조건식
참
반복될 문장들
거짓
반복될 문장들
while 다음 문장들
do-while 문
반복될 문장들
조건식
참
거짓
do-while 다음 문장들
for 문
① 초기식
② 조건식
참
③ 반복될 문장들
④ 증감식
거짓
for 다음 문장들
while 문
do-while 문
for 문