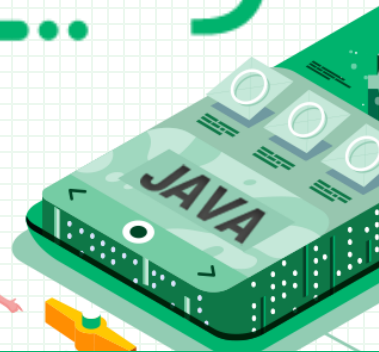




안드로이드 프로그래밍을 위한 자바기초 _...



④ 조건문과 반복문의 기본 문법 이해하기



학습목표

- 제어문을 이해하고 프로그래밍에 적용할 수 있다.
- 조건문을 이해하고 프로그래밍에 적용할 수 있다.
- 반복문을 이해하고 프로그래밍에 적용할 수 있다.



학습내용

- 제어문 이해하기
- 조건문 이해하기
- 반복문 이해하기

④ 조건문과 반복문의 기본 문법 이해하기

제어문 이해하기

▶ 제어문의 개념

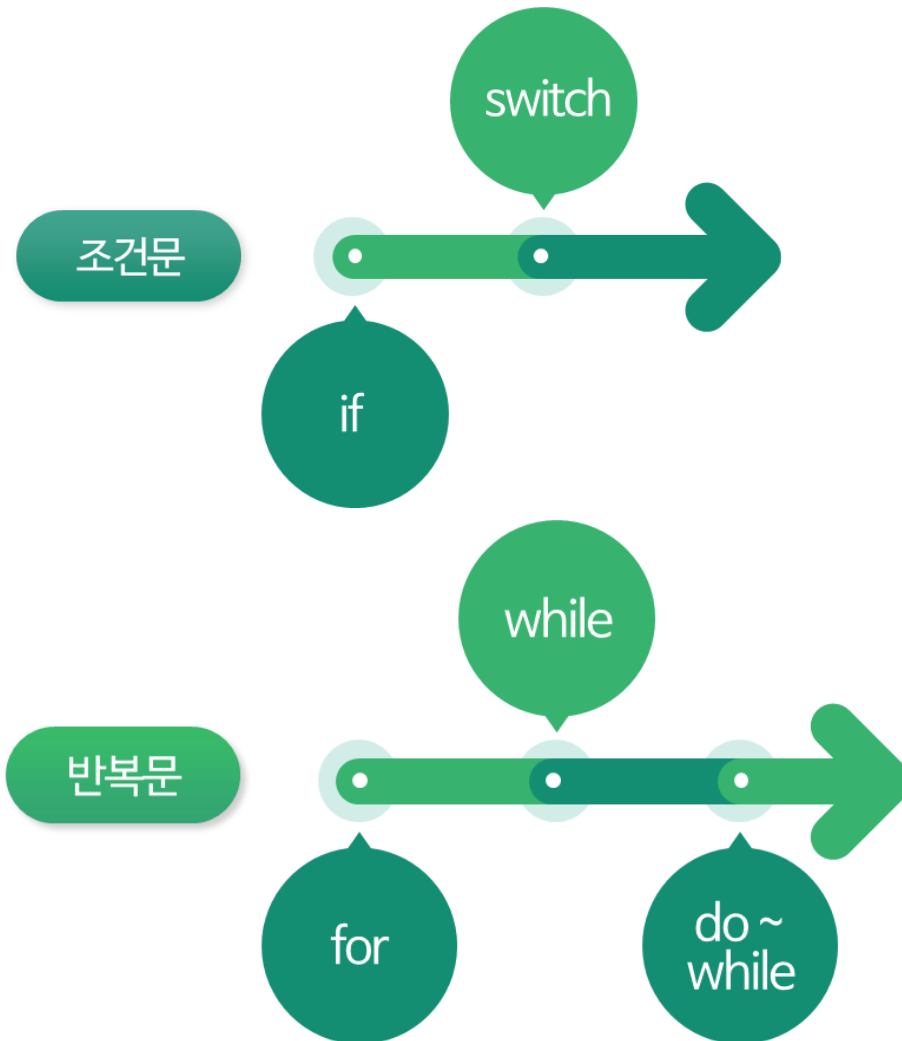
1) 제어문이란?

- 프로그램 명령이 순서를 제어하는 명령문

2) 제어문의 분류

- 조건문 - 조건이 참인지 거짓인지 따라서 실행 명령을 분리하는 것
- 반복문 - 조건이 참인 동안 명령들을 반복 수행하는 것

3) 제어문의 종류



④ 조건문과 반복문의 기본 문법 이해하기

제어문 이해하기

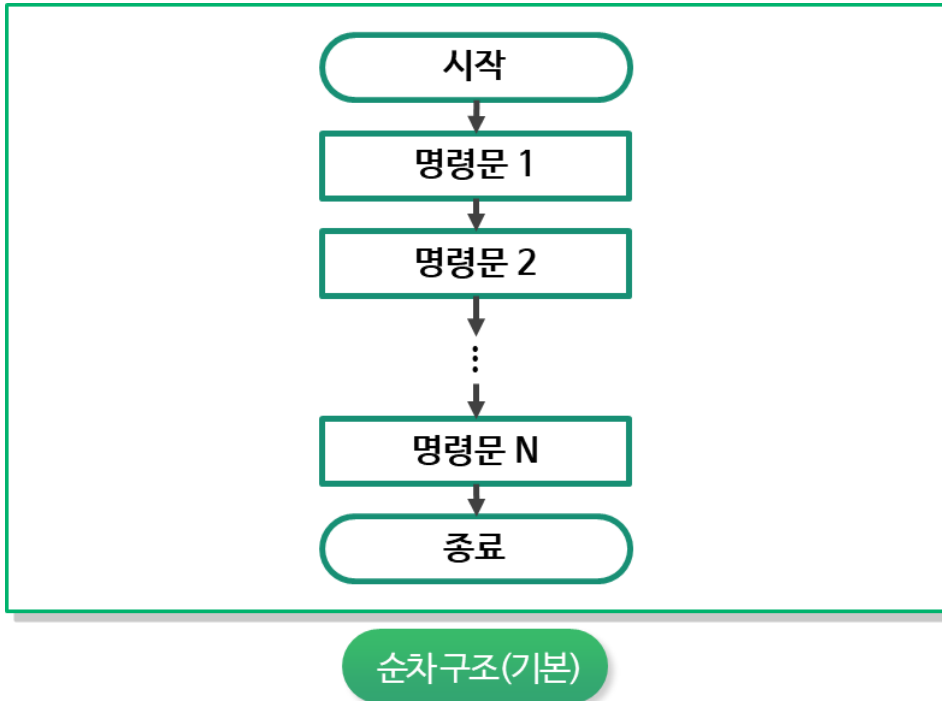
➤ 제어문의 구조

1) 제어문 구조의 종류

- 순차구조
- 선택구조
- 반복구조

(1) 순차구조

- ✓ 기본 구조로 명령문 하나씩 위에서 아래로 순차적으로 실행

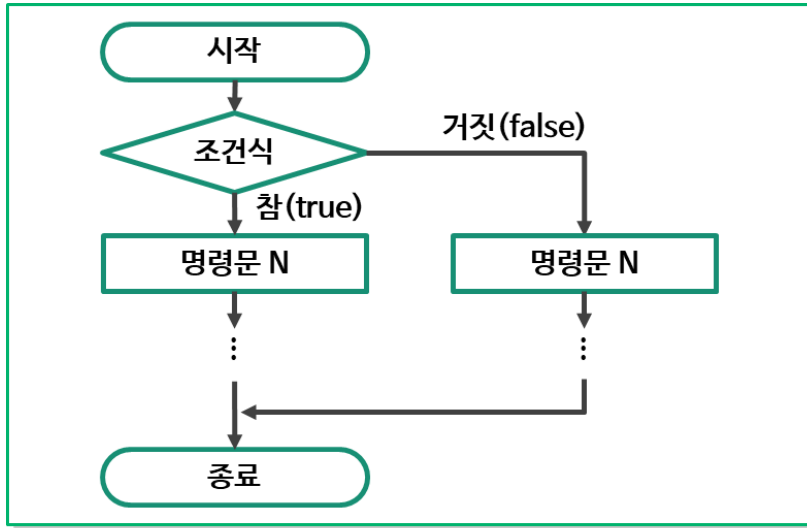


④ 조건문과 반복문의 기본 문법 이해하기

제어문 이해하기

(2) 선택구조

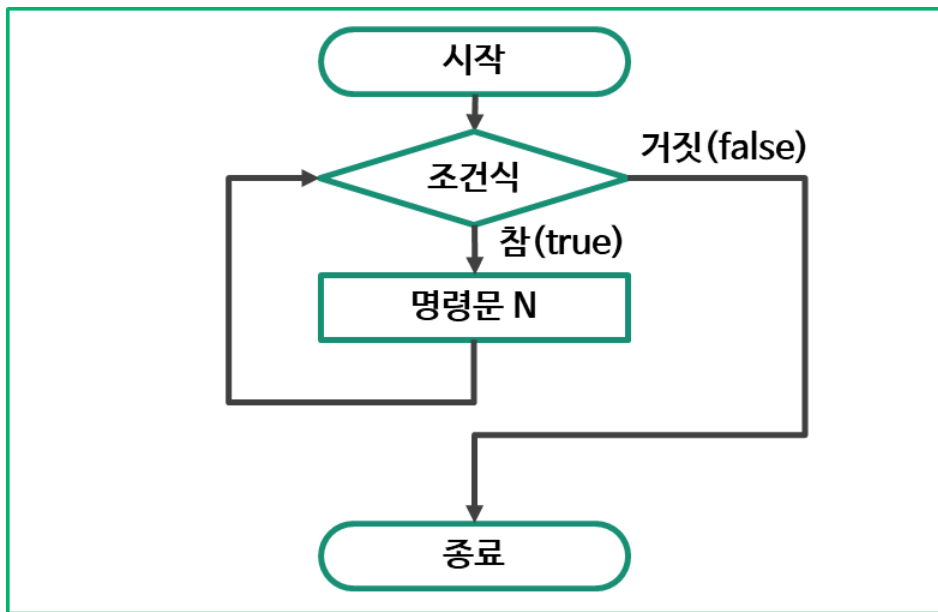
- ✓ 조건식의 참과 거짓에 따라 명령문들을 분리하여 실행



선택구조(조건문)

(3) 반복구조

- ✓ 조건식이 참인 동안 명령문들을 실행



반복구조(반복문)

④ 조건문과 반복문의 기본 문법 이해하기

조건문 이해하기

▶ if 조건문 이해하기

1) if 조건문 규칙

- (1) 참과 거짓을 구분하여 실행하는 명령어 블록은 중괄호 ({}) 로 표시
- (2) 참과 거짓을 구분하여 실행하는 명령어가 한 개일 경우는 중괄호({}) 생략 가능
- (3) if 조건문의 명령어 블록 내부에 또다시 if 조건문을 사용할 수 있음(중첩 가능)
- (4) 중괄호({}) 내부에 선언된 변수는 중괄호({})안에서만 사용가능
- (5) 중괄호({}) 외부에 선언된 변수는 내부에서 사용할 수 있음
- (6) 조건식에는 세미콜론(;)을 붙이지 않음

2) if 조건문 종류

단순 if() 문

```
if(조건식)
{
    :
}
```

if() ~ else 문

```
if(조건식)
{
    :
}
else
{
    :
}
```

if() ~ else if() 문

```
if(조건식)
{
    :
}
else if(조건식)
{
    :
}
```

if() ~ else if()
~ else문

```
if(조건식)
{
    :
}
else if(조건식)
{
    :
}
else
{
    :
}
```

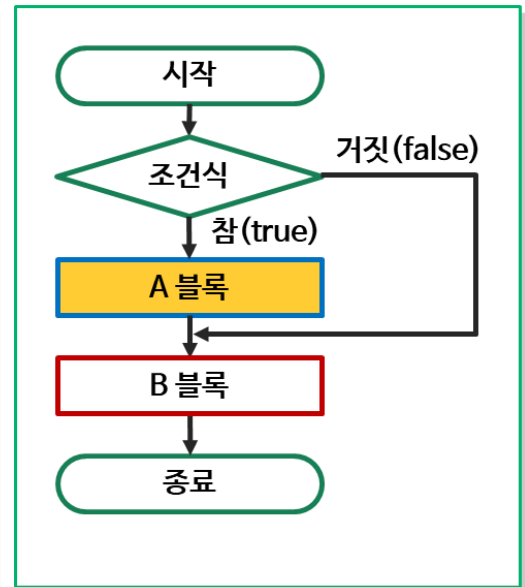
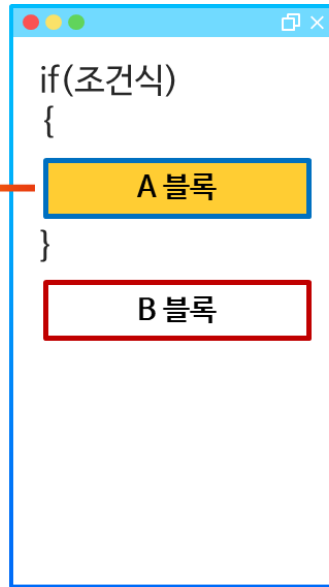
④ 조건문과 반복문의 기본 문법 이해하기

조건문 이해하기

3) 단순 if()문

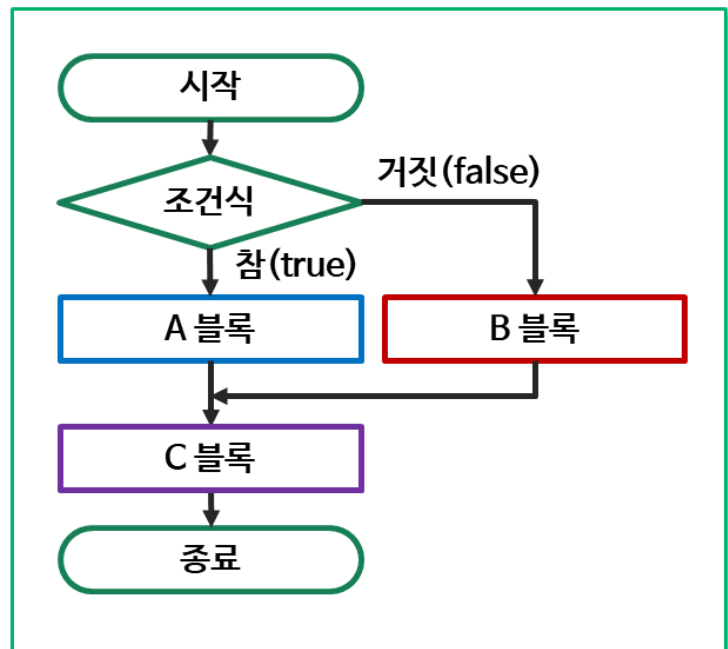
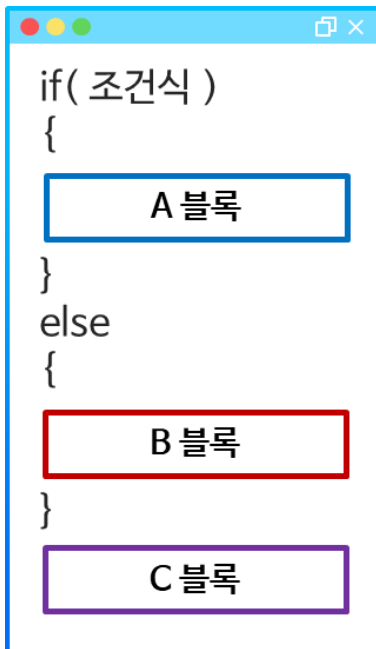
- 구조 : 조건식이 참인 경우 실행되는 블록을 가지는 구조

명령어가 한개만 있다면
중괄호 생략가능



4) if() ~ else 문

- 구조 : 조건식이 참인 경우와 거짓인 경우 각각 실행 블록을 가지고 있는 구조

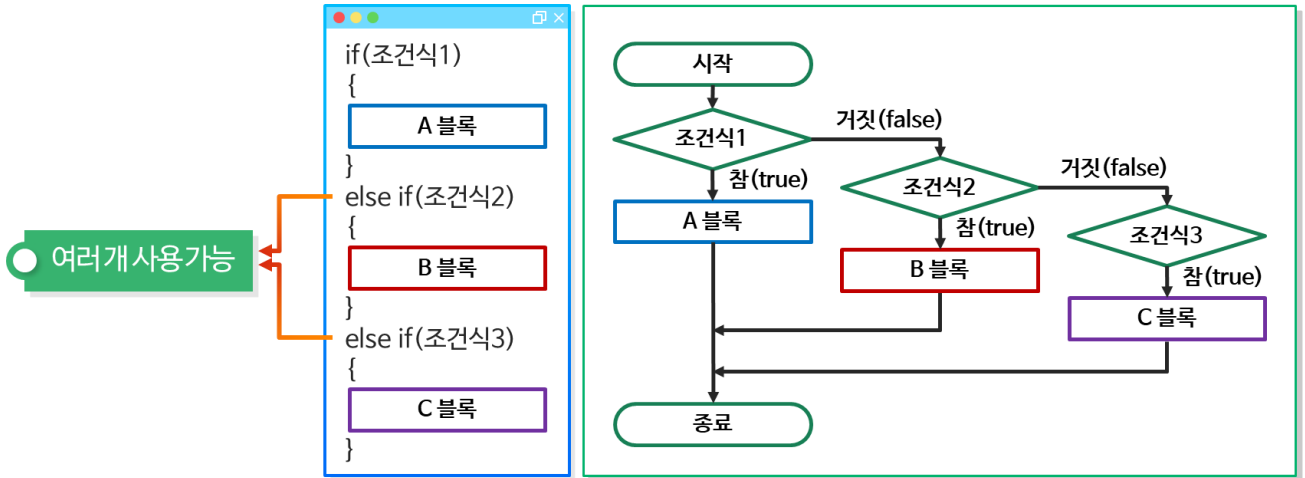


④ 조건문과 반복문의 기본 문법 이해하기

조건문 이해하기

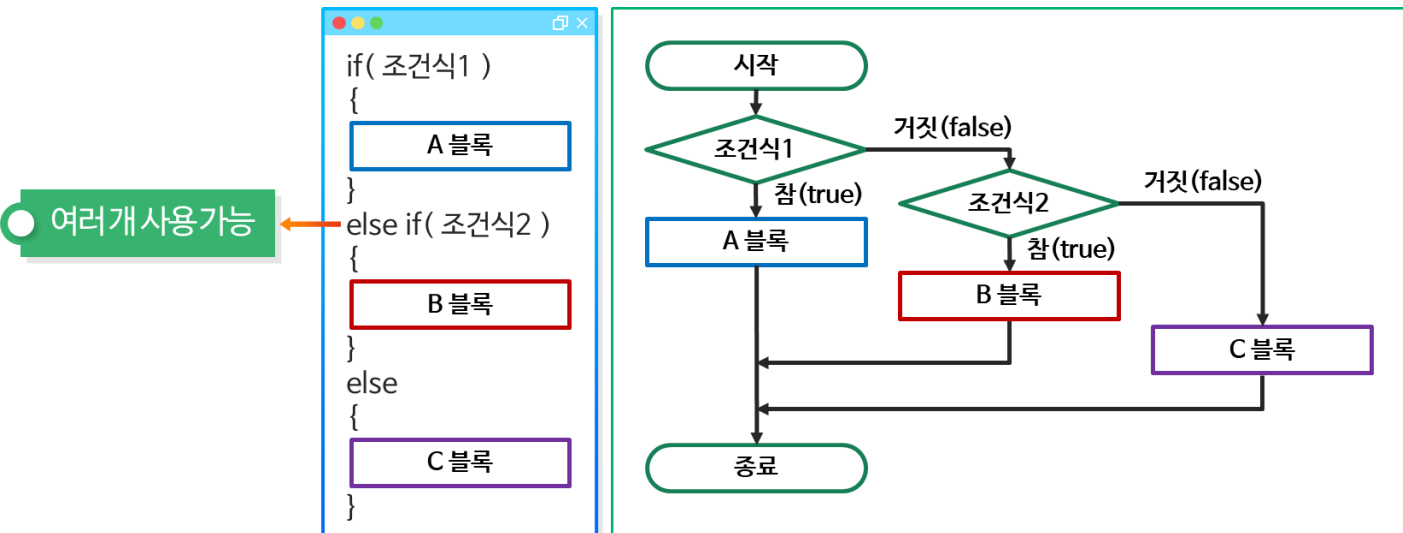
5) if() ~ else if() 문

- 구조 : 여러 개의 조건식이 참이면 실행되는 블록들을 가지는 구조

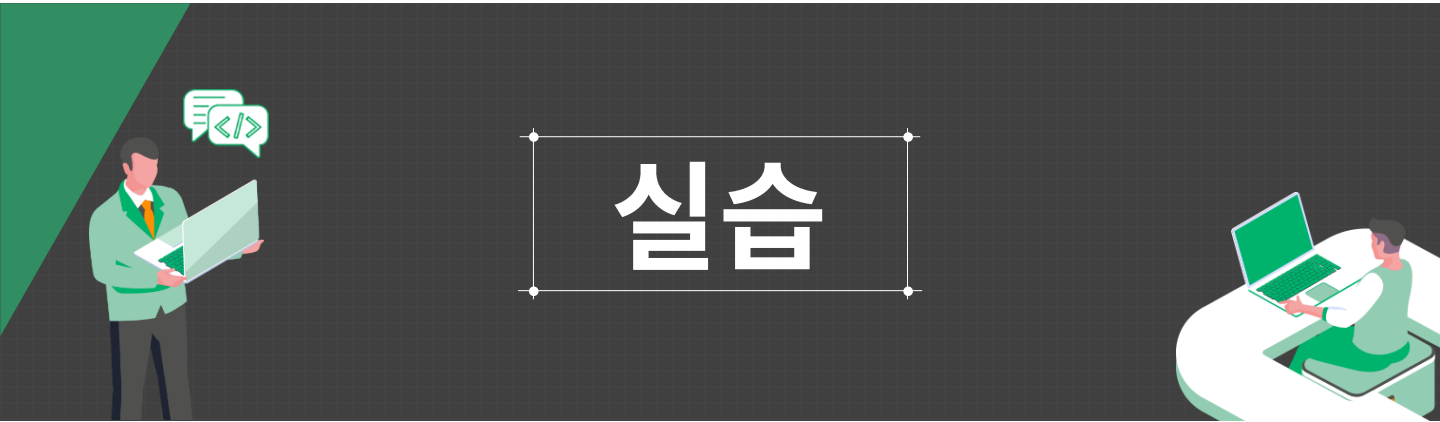


6) if() ~ else if() ~ else 문

- 구조 : 여러 개의 조건식이 참이면 실행되는 블록들과 모든 조건식이 거짓인 경우 실행되는 블록을 가짐



④ 조건문과 반복문의 기본 문법 이해하기



if 조건문 실습



실행 화면

[단순 if()문 연습] 숫자를 입력하세요 : 8
 계산된 값은 : 11
 [if() ~else()문 연습] 숫자를 입력하세요 : 98
 계산된 값은 : 99
 [if() ~else if()문 연습] 숫자(점수)를 입력하세요 :
 90
 90 점은 A 학점입니다.
 [if() ~else if() ~else문 연습] 숫자(점수)를
 입력하세요 : 50
 50 점은 F 학점입니다.

- 소스 파일명 : [conIf.java]
- 자세한 내용은 실습 영상을 확인해보세요.

④ 조건문과 반복문의 기본 문법 이해하기

조건문 이해하기

➤ switch 조건문 이해하기

1) switch 조건문이란?

- (1) if()~else if()문 또는 if()~else if()~else문의 복잡성의 단순화
- (2) case문, break문, default문으로 구성됨

```
if( 조건식1 ) {  
    명령문들..  
} else if (조건식2) {  
    명령문들..  
} else if (조건식3) {  
    명령문들..  
} else {  
    명령문들..  
}
```

단순화

```
switch( 변수 또는 수식 )  
{  
    case 상수1:  
        명령문들..  
        break;  
    case 상수2:  
        명령문들..  
        break;  
    default :  
        명령문들..  
        break;  
}
```

④ 조건문과 반복문의 기본 문법 이해하기

조건문 이해하기

2) switch 조건문 구조

```

switch( 변수 또는 수식 )
{
    case 상수1 :
        A 블록
        break;
    case 상수2 :
        B 블록
        break;
    default :
        C 블록
        break;
}

```

● 세미콜론(;) 없음

- 변수는 정수형, 문자형, 문자열 변수 사용
- 수식은 정수형 또는 문자형 결과값이 나오는 수식 사용
- 상수 1값과 "변수 또는 수식"의 값이 동일한 경우 A 블록 실행
- switch문의 외부로 제어가 이동
- break문이 없으면 B 블록 실행
- 상수 2값과 "변수 또는 수식"의 값과 동일한 경우 B 블록 실행
- 동일하지 않으면 default문 실행
- switch문의 외부로 제어가 이동
- break문이 없으면 C 블록 실행
- case문은 switch문 내부에 1개 이상 사용
- 모든 상수값들과 "변수 또는 수식"의 값이 동일하지 않으면 C 블록 실행
- 마지막에 1개만 사용
- switch문의 외부로 제어가 이동
- break문 생략 가능
- case문과 default문의 마지막에 콜론(:) 사용

④ 조건문과 반복문의 기본 문법 이해하기

조건문 이해하기

3) switch 조건문과 if 조건문의 비교

```

switch( 변수 또는 수식 )
{
    case 상수1 :
        A 블록
        break;
    case 상수2 :
        B 블록
        break;
    default :
        C 블록
        break;
}

```

상호 변환 가능

```

if( 변수 또는 수식 == 상수1 )
{
    A 블록
}
else if( 변수 또는 수식 == 상수2 )
{
    B 블록
}
else
{
    C 블록
}

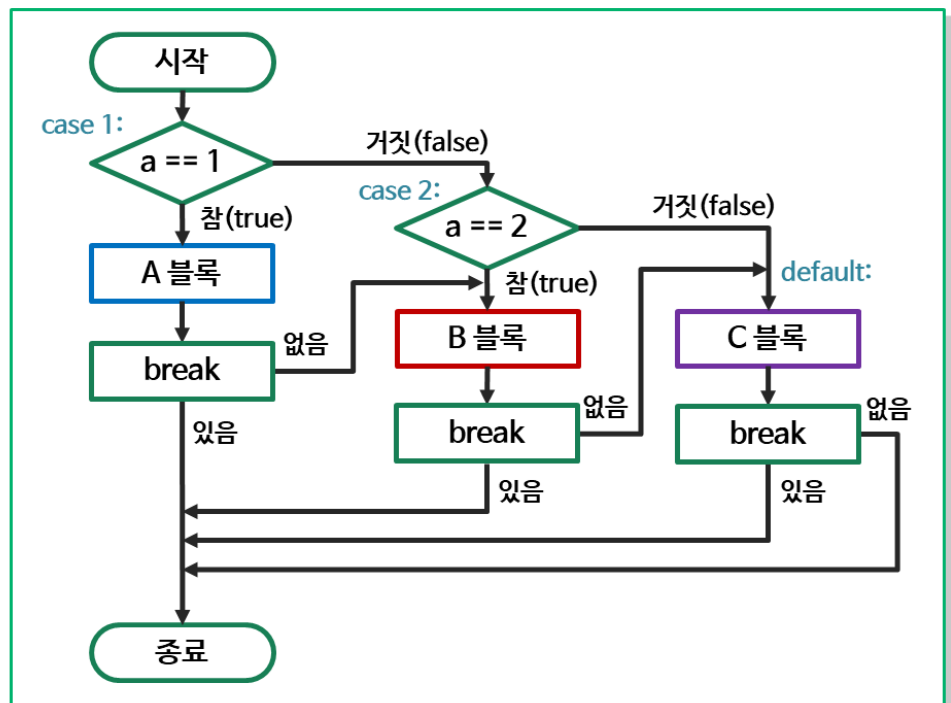
```

4) switch 조건문과 순서도

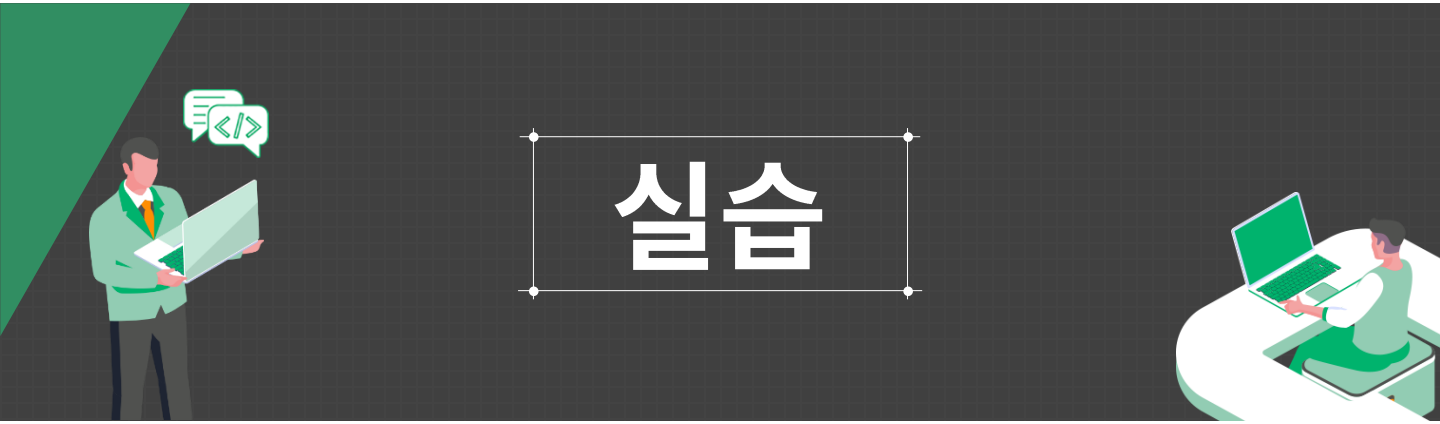
```

switch( a )
{
    case 1 :
        A 블록
        break;
    case 2 :
        B 블록
        break;
    default :
        C 블록
        break;
}

```



④ 조건문과 반복문의 기본 문법 이해하기



switch 조건문 실습



실행 화면

[숫자를 문자로 변경] 숫자를 입력하세요(1~3) : 2
TWO
[홀수와 짝수 구분하기] 숫자를 입력하세요 : 56
56는 짝수입니다.

- 소스 파일명 : [conSwitch.java]
- 자세한 내용은 실습 영상을 확인해보세요.

④ 조건문과 반복문의 기본 문법 이해하기

반복문 이해하기

▶ 반복문의 개념

1) 반복문이란?

- 반복적으로 명령문을 실행시킬 수 있는 문장 (중복성 제거, 간결해짐)
- 반복문 내부에 또는 다른 반복문을 사용할 수 있음 (중첩 사용)

2) 반복문의 예

- "안녕하세요?" 라는 문장을 100번 화면에 출력하는 경우

반복문을 사용하지 않는 경우

동일한 명령문
(System.out.println("안녕하세요");)을
100번 사용해야함



중복성, 비효율성

반복문을 사용하는 경우

```
for (inti=0;i<100;i++)
    System.out.println("안녕하세요");
```



중복성 회피, 간결해짐

3) 반복문의 종류

- 반복 횟수가 정해져 있는 경우 : for문
- 반복 횟수가 정해지지 않는 경우 : while문, do~while문

4) 무한반복

- 반복문이 종료되지 않고 지속적으로 반복되는 현상
 - ✓ break - 반복을 종료하는 명령문
 - ✓ continue - 반복을 건너뛰는 명령문

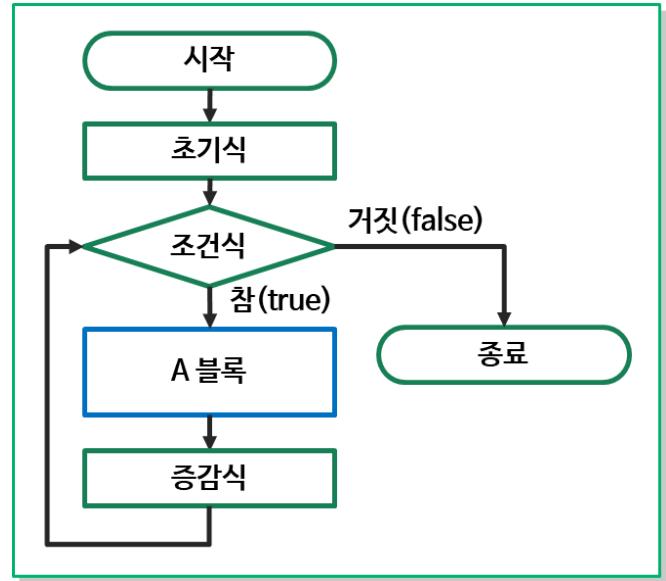
④ 조건문과 반복문의 기본 문법 이해하기

반복문 이해하기

▶ for 반복문 이해하기

1) for 반복문 구조

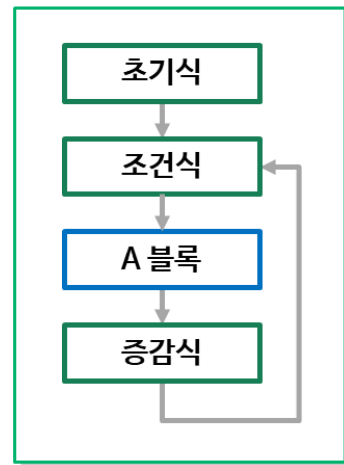
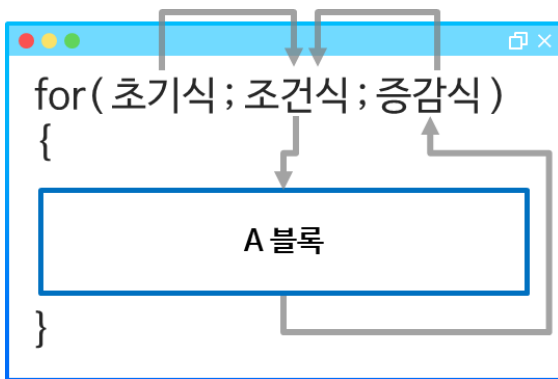
- 반복 전에 초기식 설정 수식
- 반복 여부를 판단하는 조건식
- 반복할 때마다 수행하는 증감식
- 반복할 명령어들
- 명령어가 하나인 경우 중괄호 ({}) 생략 가능
- 초기식, 조건식, 증감식은 세미콜론(;)으로 구분
- for문 마지막에 세미콜론(;) 없음



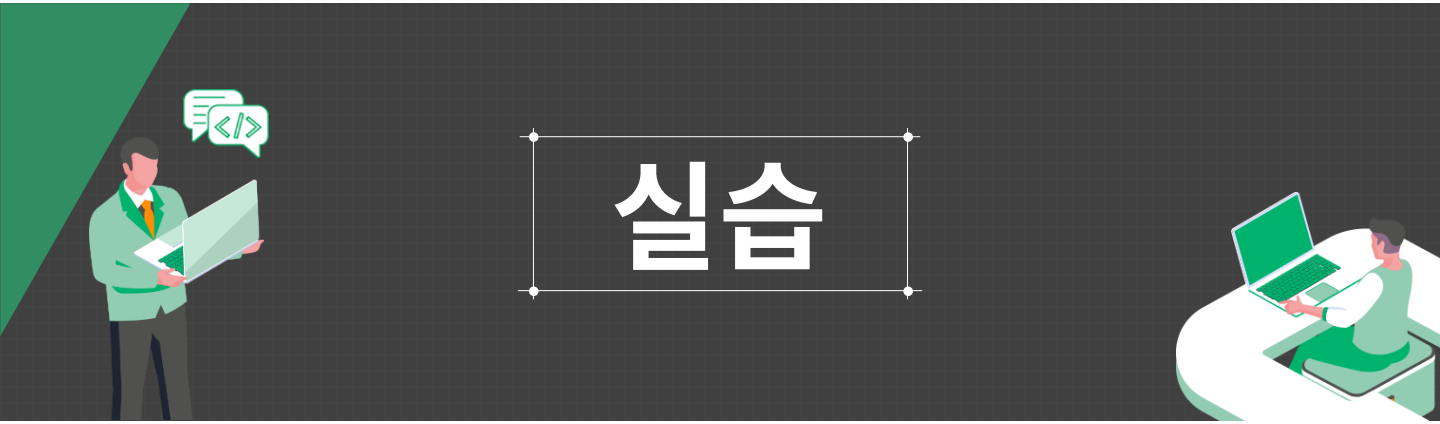
```

for( 초기식; 조건식; 증감식 )
{
    A 블록
}
  
```

2) for 반복문 동작



④ 조건문과 반복문의 기본 문법 이해하기



for 반복문 실습



실행 화면

[동일한 문장 3번 반복하기]

Hello

Hello

Hello

[1에서 10까지 출력, 합계출력]

1 2 3 4 5 6 7 8 9 10 1~10까지의 합계는 55

[10에서 1까지 출력]

10 9 8 7 6 5 4 3 2 1

[1에서 10까지 짝수만 출력]

2 4 6 8 10

[1에서 10까지 홀수만 출력]

1 3 5 7 9

[구구단 2단 출력] ...

- 소스 파일명 : [conFor.java]
- 자세한 내용은 실습 영상을 확인해보세요.

④ 조건문과 반복문의 기본 문법 이해하기

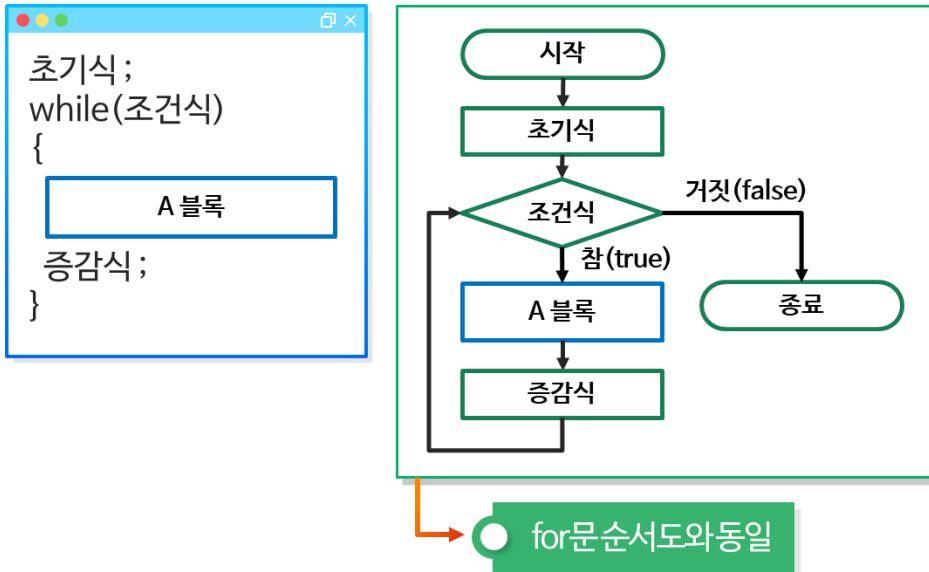
반복문 이해하기

▶ while 반복문 이해하기

1) while 반복문이란?

- 반복횟수가 미리 정해지지 않은 경우 주로 사용
- 조건식이 참인 동안 실행되는 반복문
- 초기식은 while 반복문이 시작되기 전에 선언
- 증감식은 while 반복문의 종괄호({})의 마지막에 선언
- 초기식과 증감식은 필요에 따라 생략가능

2) while 반복문 구조

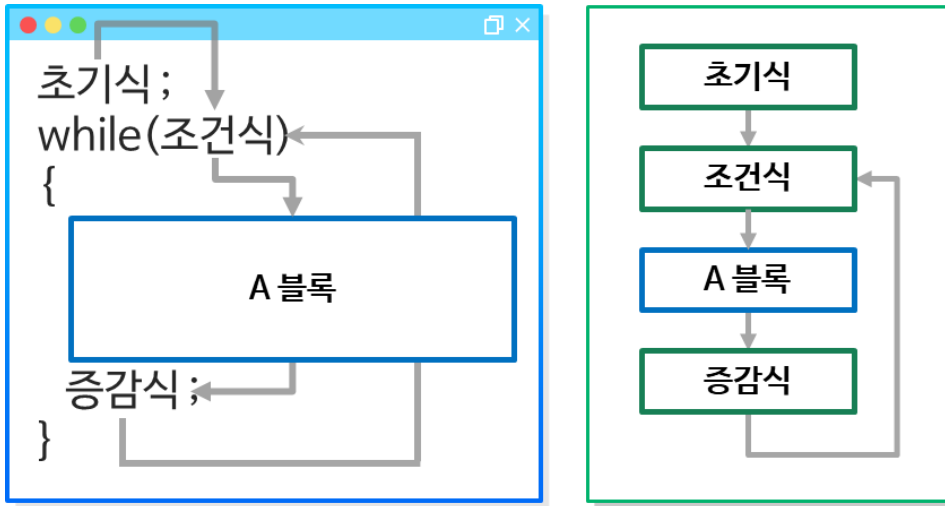


- 반복 전 초기식 설정 수식
- 반복 여부를 판단하는 조건식
- 반복할 때 마다 수행하는 증감식
- 반복할 명령어들
- 종괄호 ({}) 내부에 명령어가 하나인 경우 종괄호 ({}) 생략 가능
- 초기식과 증감식은 필요에 따라 생략 가능
- while문 마지막에는 세미콜론 사용하지 않음

④ 조건문과 반복문의 기본 문법 이해하기

반복문 이해하기

3) while 반복문의 동작



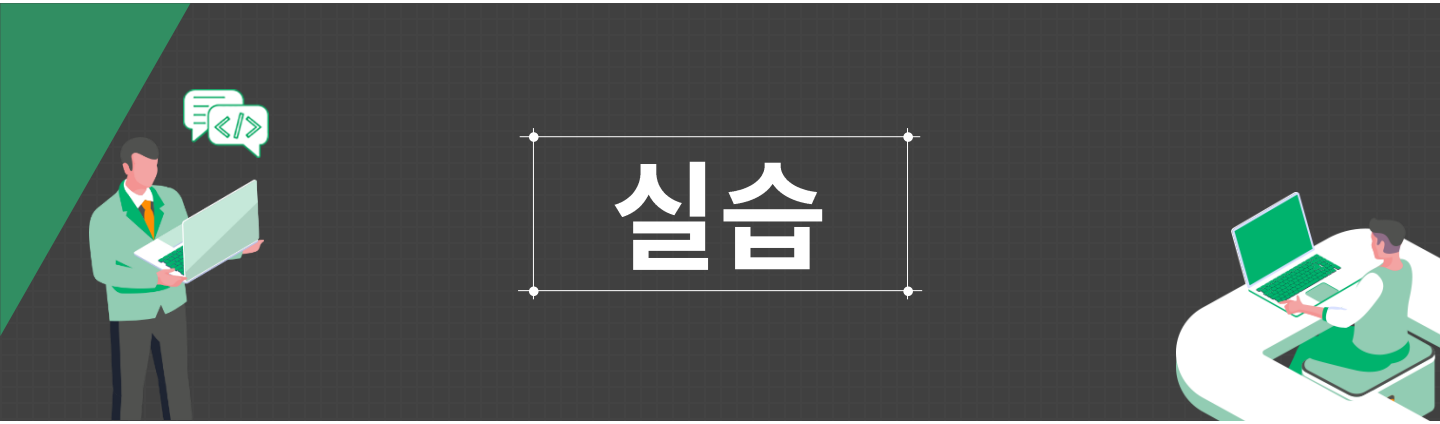
4) break 제어문이란?

- (1) 반복문(for, while, do~while)에서 사용
- (2) 반복문의 중괄호 ({})를 벗어나 반복문을 종료
- (3) 중첩된 반복문에서는 가장 가까운 중괄호 ({})를 벗어남

5) continue 제어문이란?

- (1) 반복문(for, while, do~while)에서 사용
- (2) 반복문 내부의 이후 명령문들은 건너뛴

④ 조건문과 반복문의 기본 문법 이해하기



while 반복문 실습



실행 화면

[1에서 10까지 출력하고, 합계 출력하기]
 1 2 3 4 5 6 7 8 9 10 누적합계는 55 입니다.
 [1에서 100까지 누적합계가 100이 넘어가는 숫자 출력하기]
 count = 14 sum = 105
 [정수값을 반복해서 입력받아 화면에 출력(0입력시 종료)]
 숫자를 입력하세요(0은 종료) : 5
 입력받은 값은 = 5
 숫자를 입력하세요(0은 종료) : 0
 [입력받은 값이 짝수이면 누적 합계, 홀수이면 다시 입력 받기(0입력시 종료)]
 숫자를 입력하세요(0은 종료) : 4...

- 소스 파일명 : [conWhile.java]
- 자세한 내용은 실습 영상을 확인해보세요.

④ 조건문과 반복문의 기본 문법 이해하기

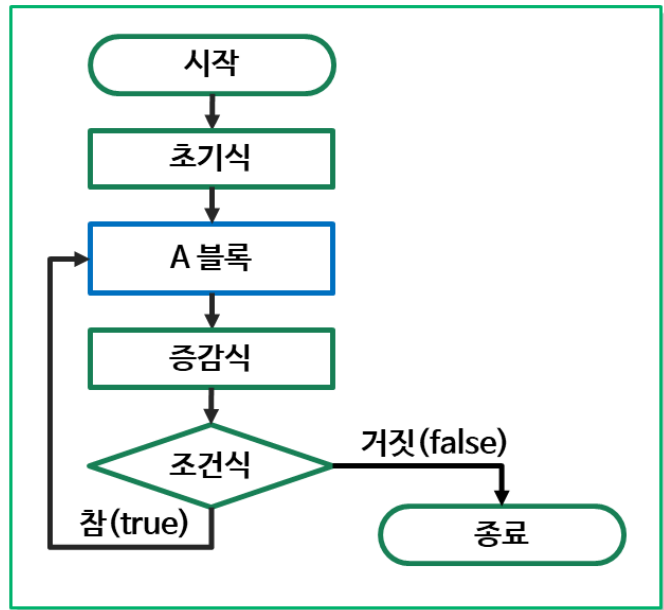
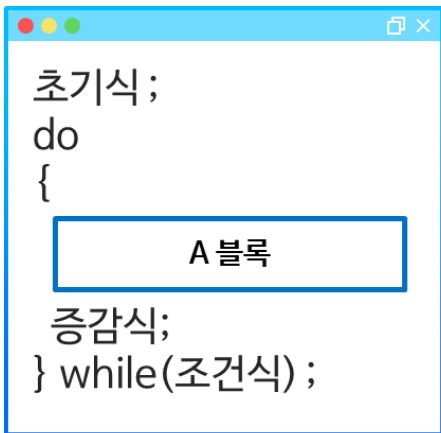
반복문 이해하기

▶ while 반복문 이해하기

1) do~while 반복문이란?

- (1) 반복 횟수가 미리 정해지지 않은 경우 주로 사용
- (2) 초기식은 do~while 반복문이 시작되기 전에 선언
- (3) 증감식은 do~while 반복문의 종괄호({})의 마지막에 선언
- (4) 초기식과 증감식은 필요에 따라 생략 가능
- (5) 조건식과 상관없이 종괄호({}) 내부의 명령문들이 먼저 한번 실행된 후 조건식이 실행됨(while 반복문과의 차이점)

2) do~while 반복문 구조

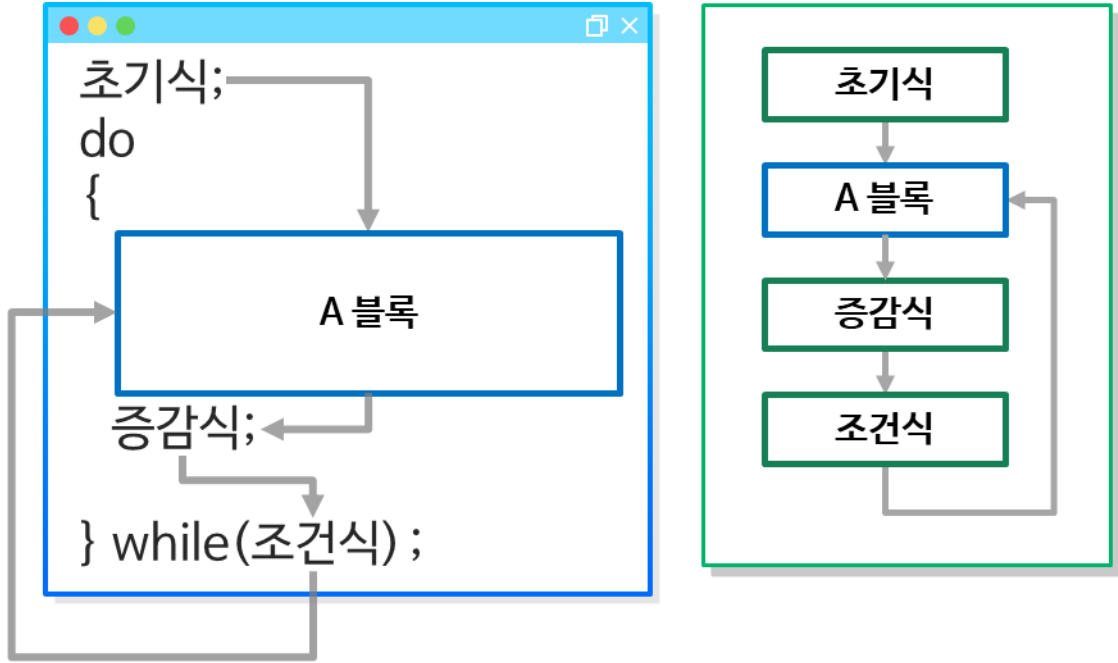


- 반복 전에 초기식 설정 수식
- 반복할 명령어들
- 반복시 수행하는 증감식
- 종괄호 ({}) 내부에 명령어가 하나인 경우 종괄호({}) 생략 가능
- 반복 여부를 판단하는 조건식
- 초기식과 증감식은 필요에 따라 생략 가능

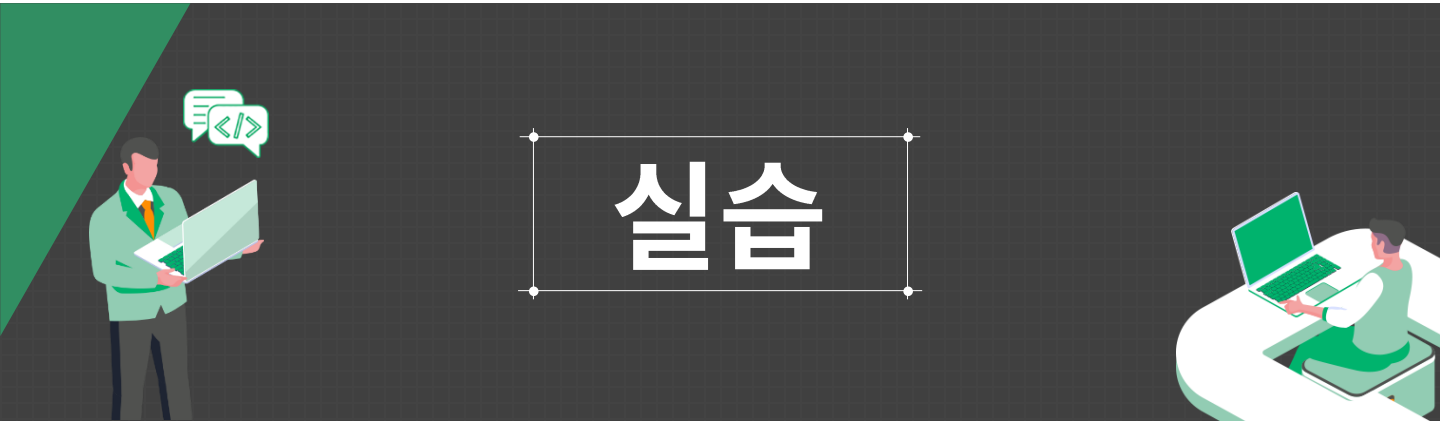
④ 조건문과 반복문의 기본 문법 이해하기

반복문 이해하기

3) do~while 반복문의 동작



④ 조건문과 반복문의 기본 문법 이해하기



do~while 반복문 실습



실행 화면

[1에서 10사이의 값을 출력, 누적 합계 출력]

1 2 3 4 5 6 7 8 9 10 sum = 55

[숫자를 반복해서 입력, 0입력시 종료]

숫자를 입력하세요 : 5

숫자를 입력하세요 : 4

숫자를 입력하세요 : 0

[구구단을 입력받아 출력하는 프로그램, 0입력시 종료]

단을 입력하세요(0:종료) => 3

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

단을 입력하세요(0:종료) => 4

4*1=4 4*2=8 4*3=12 4*4=16 4*5=20...

- 소스 파일명 : [conDowhile.java]
- 자세한 내용은 실습 영상을 확인해보세요.



정리하기

■ 제어문 이해하기

- 제어문이란?
 - 프로그램 명령이 실행되는 순서를 제어하는 명령문
- 제어문의 분류
 - 조건문 : 조건이 참인지 거짓인지에 따라서 실행 명령을 분리하는 것
 - 반복문 : 조건이 참인 동안 명령들을 반복 수행하는 것
- 제어문 구조의 종류
 - 순차구조 : 기본 구조로 명령문 하나씩 위에서 아래로 순차적으로 실행
 - 선택구조 : 조건식이 참과 거짓에 따라 명령문들을 분리하여 실행
 - 반복구조 : 조건식이 참인 동안 명령문들을 실행



정리하기

■ 조건문 이해하기

• if 조건문 이해하기

- 참과 거짓을 구분하여 실행하는 명령어 블록은 중괄호({})로 표시
- 참과 거짓을 구분하여 실행하는 명령어가 한 개일 경우는 중괄호({}) 생략 가능
- if 조건문의 명령어 블록 내부에 또다시 if 조건문을 사용할 수 있음(중첩 가능)
- 중괄호({}) 내부에 선언된 변수는 중괄호({})안에서만 사용 가능
- 중괄호({}) 외부에 선언된 변수는 내부에서 사용할 수 있음
- 조건식에는 세미콜론(;)을 붙이지 않음

• switch 조건문 이해하기

- if()~else if()문 또는 if()~else if()~else문의 복잡성의 단순화
- case문, break문, default문으로 구성됨



정리하기

■ 반복문 이해하기

• 반복문이란?

- 반복적으로 명령문을 실행시킬 수 있는 문장(중복성 제거, 간결해 짐)
- 반복문 내부에 또 다른 반복문을 사용할 수 있음(중첩 사용)
- 반복문의 종류
 - for문 : 반복 횟수가 정해져 있는 경우
 - while문, do~while문 : 반복 횟수가 정해지지 않는 경우

• for 반복문 이해하기

- 초기식 : 반복하고자 하는 블록을 반복하기 전에 초기식을 설정하는 수식
- 조건식 : 반복여부를 판단하는 수식으로 조건식이 참인 경우 블록을 실행하고, 거짓인 경우 반복을 종료
- 증감식 : 반복 블록을 한 번 실행한 후에 실행되는 증감 수식



정리하기

■ 반복문 이해하기

• while 반복문 이해하기

- 반복횟수가 미리 정해지지 않은 경우 주로 사용
- 조건식이 참인 동안 실행되는 반복문
- 초기식은 while 반복문이 시작되기 전에 선언
- 증감식은 while 반복문의 종괄호({})의 마지막에 선언
- 초기식과 증감식은 필요에 따라 생략 가능

• do~while 반복문이란?

- 반복횟수가 미리 정해지지 않은 경우 주로 사용
- 초기식은 do~while 반복문이 시작되기 전에 선언
- 증감식은 do~while 반복문의 종괄호({})의 마지막에 선언
- 초기식과 증감식은 필요에 따라 생략 가능
- 조건식과 상관없이 종괄호({}) 내부의 명령문들이 먼저 한번 실행된 후 조건식이 실행됨(while 반복문과의 차이점)