

# 제어문

# ▶ 제어문

프로그램 수행의 흐름을 결정(제어)하는 데 사용되는 문장

프로그램은 항상 위에서 부터 아래로 실행되는데 조건에 따라 원하는 순서로 실행되도록 흐름을 조정 함

## ✓ 제어문의 종류

조건문 : if문, switch문

반복문 : while문, do-while문, for문

분기문 : continue , break , return

# 목차

- ✓ Chap01. 조건문
- ✓ Chap02. 반복문
- ✓ Chap03. 분기문

# Chap01. 조건문

# ▶ 조건문

조건식을 제시하고 조건식의 결과에 따라 다음 코드의 수행 여부를 결정하는 문장

## ✓ 조건문의 종류

### if문

```
if(조건식1) {  
    수행될 문장;  
} else if(조건식2) {  
    수행될 문장;  
} else if(조건식3) {  
    수행될 문장;  
} else {  
    수행될 문장;  
}
```

### switch문

```
switch(조건식) {  
    case 값1:  
        수행될 문장;  
        break;  
    case 값2:  
        수행될 문장;  
        break;  
    default:  
        수행될 문장;  
}
```

# ▶ if문

조건이 참일 때 다음에 오는 하나의 명령을 실행

조건이 참일 때 실행 할 내용이 한 줄 이상일 경우 { } 로 영역지정

## ✓ if문 사용법

```
if(조건식)  
    명령;
```

조건식의 결과 값이 true면 명령 실행  
false면 실행하지 않음

```
if(조건식){  
    명령1;  
    명령2;  
}
```

조건식의 결과 값이 true면 명령1,2 실행  
false면 실행하지 않음

# ▶ if문

## ✓ if문 예시

```
if(num > 0) {  
    System.out.println("양수입니다.");  
}
```

## ✓ 예시 실행 결과

num = 10 인 경우

-----

양수입니다.

num = -5 인 경우

-----

(출력결과 없음)

# ▶ if-else문 → else if

if 문의 조건식의 결과가 true 일 경우와 false일 경우 수행할 문장을 각각 지정 조건이 참일 때 if 다음 명령을 실행하고 거짓일 때 else 다음 명령을 실행

## ✓ if-else문 사용법

```
if(조건식) {  
    명령1;  
} else {  
    명령2;  
}
```

조건식의 결과 값이 true면 명령1 실행  
false면 명령2 실행



# ▶ if-else문

## ✓ if~else문 예시

```
if(num % 2 == 0) {  
    System.out.println("짝수");  
} else {  
    System.out.println("홀수");  
}
```

## ✓ 예시 실행 결과

num = 10 인 경우

-----

짝수

num = 5 인 경우

-----

홀수

# ▶ if – else if - else 문

조건을 2개 이상 주고 각 조건에 따라 다른 내용을 실행

## ✓ if-else if-else 문 사용법

<code>if(조건식1) {</code>	조건식1의 결과 값이 true면 명령1 실행
<code>    <b>명령1;</b></code>	
<code>} else if(조건식2){</code>	false면 조건식2 확인
<code>    <b>명령2;</b></code>	조건식2의 결과 값이 true면 명령2 실행
<code>} else {</code>	false면 명령3 실행
<code>    <b>명령3;</b></code>	
<code>}</code>	

\* if는 true, false와 상관 없이 조건절 실행,  
if~else if~else는 조건문이 true면 이후 조건은 실행하지 않음

# ▶ if – else if - else 문

## ✓ if~else if~else문 예시

```
if(num > 0) {  
    System.out.println("0보다 큰 수");  
} else if (num < 0){  
    System.out.println("0보다 작은 수");  
} else {  
    System.out.println("0과 같은 수");  
}
```

## ✓ 예시 실행 결과

num = 10 인 경우

-----

0보다 큰 수

num = -5 인 경우

-----

0보다 작은 수

num = 0 인 경우

-----

0과 같은 수

# ▶ if – else if - else 문

## ✓ if~else if~else문 예시

```
if(month == 1 || month == 2 || month == 12) {  
    season = "겨울";  
} else if(month >= 3 && month <= 5) {  
    season = "봄";  
} else if(month >= 6 && month <= 8) {  
    season = "여름";  
} else if(month >= 9 && month <= 11) {  
    season = "가을";  
} else {  
    season = "해당하는 계절이 없습니다.";  
}
```

# ▶ if문

## ✓ 중첩 if

```

if (조건식1) {
    if (조건식2) {
        if (조건식3) {
            수행될 문장;
        } else if (조건식4) {
            수행될 문장;
        } else {
            수행될 문장;
        }
    } else {
        수행될 문장;
    }
} else if (조건식5) {
    수행될 문장;
} else {
    수행될 문장;
}

```

## ✓ 중첩 if문 예시

```

if (month == 1 || month == 2 || month == 12) {
    season = "겨울";
    if (temperature <= -15) {
        season += " 한파 경고";
    } else if (temperature <= -12) {
        season += " 한파 주의보";
    }
} else if (month >= 3 && month <= 5) {
    season = "봄";
} else if (month >= 6 && month <= 8) {
    season = "여름";
    if (temperature >= 35) {
        season += " 폭염 경고";
    } else if (temperature >= 33) {
        season += " 폭염 주의보";
    }
} else if (month >= 9 && month <= 11) {
    season = "가을";
} else {
    season = "해당하는 계절이 없습니다.";
}

```

## ▶ if문

-. 놀이 공원 프로그램 만들기

=====출력=====

##놀이 공원 프로그램##

입장 하실 인원은 총 몇 명입니까? 3

어른은 몇 명입니까? (인원당 1만5천원) 1

아이는 몇 명입니까? (인원당 5천원) 2

지불하실 총 금액은 25000원 입니다

# ▶ if문

-. 계산기 만들기(if문 사용)

=====출력=====

## 계산기 ##

1. 더하기

2. 빼기

3. 나누기

4. 곱하기

선택 :

## ▶ switch문

조건 값에 해당하는 경우를 찾아 선택하여 실행 하는 것  
상수 값(정수 또는 문자, 문자열)의 결과를 가지는 조건을 입력 받아 동일  
한 case 조건 이후 내용부터 모두 실행

조건식의 결과 값과 일치하는 case문으로 이동

default문은 일치하는 case문이 없을 때 수행(= else )



# ▶ switch문

## ✓ switch문 사용법

```
switch(조건식) {  
  case 조건값1 :  
      명령1;  
  case 조건값2 :  
      명령2;  
  case 조건값3 :  
      명령3;  
  default :  
  }
```

조건식의 결과값과 조건값 중 일치하는  
case 를 찾아 이후 내용 모두 실행

# ▶ switch문

## ✓ switch문 예시

```
Scanner scan = new Scanner(System.in);
System.out.print("정수를 입력하세요 : ");
int select = scan.nextInt();
switch(select)
{
    case 1 : System.out.println("1 입력");
    case 2 : System.out.println("2 입력");
    case 3 : System.out.println("3 입력");
    case 4 : System.out.println("4 입력");
    default : System.out.println("기타입력");
}
```

# ▶ switch문

## ✓ switch문 예시

```
Scanner scan = new Scanner(System.in);
System.out.print("정수를 입력하세요 : ");
int select = scan.nextInt();
switch(select)
{
    case 1 : System.out.println("1 입력");
             break;
    case 2 : System.out.println("2 입력");
             break;
    case 3 : System.out.println("3 입력");
             break;
    case 4 : System.out.println("4 입력");
             break;
    default : System.out.println("기타입력");
}
```

# ▶ switch문 break switch, while 가

## ✓ switch문 예시

```
switch(num % 5) {  
  case 1:  
    team = "1조";  
    break;  
  case 2:  
    team = "2조";  
    break;  
  case 3:  
    team = "3조";  
    break;  
  case 4:  
    team = "4조";  
    break;  
  default:  
    team = "다시";  
}
```

## ▶ switch 문

-. 계산기 만들기(switch문 사용)

=====출력=====

## 계산기 ##

1. 더하기
2. 빼기
3. 나누기
4. 곱하기

선택 :

# Chap02. 반복문

# ▶ 반복문

문장들을 반복해서 여러 번 수행되게 할 때 사용하는 구문  
조건을 만족하는 동안 명령을 반복하여 실행  
구문 상에 반복되는 구간을 루프(LOOP) 라고 함

## ✓ 반복문의 종류

### while문

```
[초기식;]
while(조건식) {
    수행될 문장;
    [증감식 or 분기문;]
}
```

### for문

```
for(초기식; 조건식; 증감식)
{
    수행될 문장;
}
```

# ▶ while문

## ✓ while

```
[초기식;]
while(조건식) {
    수행될 문장;
    [증감식 or 분기문];
}
```

조건식이 true일 때 문장 수행  
문장 수행이 끝나면 조건식  
다시 확인 후 true면 수행,  
조건식이 false가 될 때까지 수행  
조건식이 false가 되면 반복문 종료

\* {} 안에 조건을 벗어나게 할 연산(증감식, 분기문) 필요

## ✓ while문 예시

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

## ✓ 실행 결과

```
1 출력
2 출력
...
9 출력
10 출력
```



# ▶ do-while문

## ✓ do ~ while

```
[초기식;]
do {
    수행될 문장;
    [증감식 or 분기문;]
} while(조건식);
```

do 안의 내용 먼저 실행  
조건식 확인 후 true면 문장 수행,  
false면 종료  
while 뒤에 ; 꼭 필요

\* while과 do~while의 차이점 :

do~while은 조건문이 true가 아니더라도  
무조건 한 번 이상 수행

\* {} 안에 조건을 벗어나게 할 연산(증감식, 분기문) 필요

## ✓ do ~ while문 예시

```
int i = 1;
do {
    System.out.println(i + "출력");
    i++;
} while(i <= 10);
```

## ✓ 실행 결과

1 출력  
2 출력  
...  
9 출력  
10 출력

# ▶ for문

## ✓ for

```
for(초기식; 조건식; 증감식) {  
    수행될 문장;  
}
```

1회전: 초기식 확인 후 조건식 확인

조건식이 true면 문장 수행

조건식이 false면 수행하지 않음

2회전: 증감식 연산 후 조건식 확인

조건식이 true면 문장 수행

조건식이 false면 수행하지 않음

\* 2회전 이상부터는 모두 **2회전과 동일**하고  
**조건식이 false가 나올 때까지** 문장 수행

## ✓ for문 예시

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

## ✓ 실행 결과

```
1 출력  
2 출력  
...  
9 출력  
10 출력
```

# ▶ 중첩 반복문

## ✓ 표현식

```
for(초기값1; 조건식1; 증감식1) {  
    수행될 문장1;  
    for(초기값2; 조건식2; 증감식2) {  
        수행될 문장2;  
    }  
    수행될 문장3;  
}
```

for문에 진입하면 수행될 문장1을 먼저 수행하고 두 번째 for문에 진입하면 조건식2가 false가 될 때까지 수행될 문장2를 수행 후 나오면 수행될 문장3을 수행하고 조건식1로 돌아와 true면 다시 반복

# ▶ 중첩 반복문

## ✓ 표현식

```
for(초기값1; 조건식1; 증감식1) {  
    수행될 문장1;  
    for(초기값2; 조건식2; 증감식2) {  
        수행될 문장2;  
        break;  
    }  
    수행될 문장3;  
    [break;]  
}
```

두 번째 for문에 break를 만날 경우 반복문을 나가 수행될 문장3을 수행 후  
다시 첫 번째 for문을 실행하지만  
마지막 break가 있다면 수행될 문장3을 수행 후 for문을 완전히 빠져나감

# Chap03. 분기문

# ▶ 분기문

## ✓ break

반복문에서는 break문 자신이 포함된 가장 가까운 반복문을 빠져나가는 구문

## ✓ break문 예시

```
for(int i = 1;; i++) {  
    System.out.println(i + " 출력");  
  
    if(i >= 10) {  
        break;  
    }  
}
```

# ▶ 분기문

## ✓ continue

반복문 내에서만 사용 가능하며 반복문 실행 시 continue 아래 부분은 실행하지 않고 반복문 다시 실행

for문의 경우 증감식으로 이동,

while(do~while)문의 경우 조건식으로 이동

전체 반복 중에 특정 조건을 만족하는 경우를 제외하고자 할 때 유용

## ✓ continue문 예시

```
for(int i = 1; i <= 10; i++) {  
    if(i % 2 == 0) {  
        continue;  
    }  
    System.out.println(i + " 출력");  
}
```