

# 제어문

# 목차

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

# Chap01. 조건문

## ▶ 조건문

프로그램 수행 흐름을 바꾸는 역할을 하는 제어문 중 하나로 조건에 따라 다른 문장이 수행되도록 함

### ✓ 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

### ✓ if

```
if(조건식) {  
    ①  
}
```

조건식의 결과 값이 true면 ① 안의 내용 실행  
false면 실행하지 않음

### ✓ if문 예시

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

## ▶ if문

### ✓ if~else

```
if(조건식) {  
    ㉠  
} else {  
    ㉡  
}
```

조건식의 결과 값이 true면 ㉠ 안의 내용 실행  
false면 ㉡ 안의 내용 실행

### ✓ if~else문 예시

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

## ▶ if문

### ✓ if~else if~else

```
if(조건식1) {
    ㉠
} else if(조건식2){
    ㉡
} else {
    ㉢
}
```

조건식1의 결과 값이 true면

㉠ 안의 내용 실행

조건식2의 결과 값이 true면

㉡ 안의 내용 실행

모두 false면

㉢ 안의 내용 실행

### ✓ 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.";
}
```

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

## ▶ 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 경보";
    } 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 경보";
    } else if (temperature >= 33) {
        season += " 폭염 주의보";
    }
} else if (month >= 9 && month <= 11) {
    season = "가을";
} else {
    season = "해당하는 계절이 없습니다.";
}
```



## ▶ switch문

조건식 하나로 많은 경우의 수를  
처리할 때 사용하며 이때 조건식의  
결과는 정수 또는 문자, 문자열

조건식의 결과 값과 일치하는  
case문으로 이동  
default문은 일치하는  
case문이 없을 때 수행(= else )

## ✓ 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";  
}
```

# Chap02.

## 반복문

## ▶ 반복문

프로그램 수행 흐름을 바꾸는 역할을 하는 제어문 중 하나로 특정 문장들을 반복해서 수행하도록 함

### ✓ 반복문의 종류

#### for문

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

#### while문

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

## ▶ 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

## ▶ while문

### ✓ while

```
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
```

## ▶ while문

### ✓ do ~ while

```
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
```



# 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");  
}
```



## 실습문제

문자열과 문자열에서 검색될 문자를 입력 받아 문자열에 그 문자가 몇 개이었는지 개수를 확인하는 프로그램을 작성하세요.

출력 예)

문자열 입력 : application

문자 입력 : p

포함된 갯수 : 2 개

=====

문자열 입력 : apple\_test123

문자 입력 : p

영문자가 아닙니다.





## 실습문제

5명의 회원정보를 입력 받아 저장하고 출력하는  
프로그램을 만드세요

입력 : 이름, 나이, 주소, 키, 몸무게, 연락처

출력 예)

===== 저장회원=====

홍길동 19세 서울시 강남구 170cm 68kg 010...

임걱정 20세 경기도 수원시 180cm 75kg 010...

.

.





## 실습문제

버전 업 : 등록된 회원의 평균나이, 평균 키, 평균  
몸무게를 계산하고 출력해보세요

출력에 )

===== 저장회원=====

홍길동 19세 서울시 강남구 170cm 68kg 010...

임걱정 20세 경기도 수원시 180cm 75kg 010...

.

.

평균나이 00세 / 평균 키 : 00CM / 평균 몸무게 : 00kg





## 실습문제

버전 업2

입력한 회원 수를 입력하고 그 수 만큼 회원을  
입력 받게 업그레이드 하세요

출력에 )

===== 저장회원 =====

홍길동 19세 서울시 강남구 170cm 68kg 010...

임걱정 20세 경기도 수원시 180cm 75kg 010...

·  
·

평균나이 00세 / 평균 키 : 00CM / 평균 몸무게 : 00kg





## 실습문제

5개의 상품을 입력 받아 출력하는 프로그램을 만드세요

입력 : 상품id, 종류, 상품명, 가격

출력 예)

===== 저장상품 =====

p001 핸드폰 노트8 120만원

p002 핸드폰 v10 98만원

c001 컴퓨터 엘지xnote 100만원

e001 가전 전자레인지 30만원

e002 가전 세탁기 80만원





## 실습문제

몬스터 사냥하기 게임의 사냥하는 기능을 만드시오.

출력 예)

사용법

0. 사냥시작 1. 오크사냥 2. 용사냥 3. 사람사냥 99.끝내기

선택 :

사냥을 시작합니다.

입력 : 1

오크사냥완료 +1 exp

입력 : 2

용 사냥 완료 +15exp

입력 : 99

사냥을 마쳤습니다.

사냥한 몹의 수는 2마리, 획득한 경험치는 16exp입니다.



# 실습문제

## 분식집 주문 받는 프로그램을 만들어 보세요.

```
김밥류=====
1.원조김밥=====1000원
2.치즈김밥=====1200원
3.참치김밥=====2000원
라면=====
4.그냥라면=====1200원
5.치즈라면=====1500원
6.짬뽕라면=====2000원
기타=====
7.튀김=====1200원
8.순대=====2500원
9.오밀=====1000원
10.물로수=====800원
메뉴선택 : 3
수량선택 : 5
5개 주문하셨습니다.
추가주문하시겠습니까?(y/n)
```

\*주문하신 정보는 다음과 같습니다.\*

=====

원조김밥 : 5개 - 5000원

그냥라면 : 5개 - 6000원

짬뽕라면 : 2개 - 4000원

=====

총 가격 : 15000원

1. 메뉴선택
2. 수량선택
3. 추가주문선택
4. y면 1번부터 다시 시작  
단 데이터 누적
5. n면 주문결과서 출력

