

제4장

조건문(if, switch)



1. 조건문이란 무엇일까?

- 조건문은 어떤 조건으로 인하여, 프로그램이 실행하는 부분에서 분기가 일어나는 것을 의미한다.
- 조건문은 조건식과 실행될 블록{ }으로 구성된다.
- 자바에서는 if문과 switch문 두 가지 밖에 없다.
- 보통, 현업에서는 if문이 사용되어지며, switch문은 가독성이 좋으므로 정해진 값, 한정된 값에는 사용하는게 좋다.

중요 : 모든 switch문은 if문으로 변경이 가능하다. 하지만, 모든 if문은 switch문으로 변경할 수 없는 경우가 대다수이다.

```
if(score < 90) {  
    System.out.println("점수가 90보다 작습니다.");  
}
```

```
switch(num) {  
    case 1:  
        System.out.println("1번이 나왔습니다.");  
        break;  
}
```

2. if문이란?

- if문은 if, if-else, if-else if-else의 세가지 형태가 존재한다.
- 조건식의 결과는 반드시 true 또는 false이어야 한다.

```
if(조건식) {  
    // 조건식의 결과가 true일 때 수행될 문장들  
}
```

```
if(조건식) {  
    // 조건식의 결과가 true일 때 수행될 문장들  
} else {  
    // 조건식의 결과가 false일 때 수행될 문장들  
}
```

```
if(조건식1) {  
    // 조건식1의 결과가 true일 때 수행될 문장들  
} else if(조건식2) {  
    // 조건식2의 결과가 true일 때 수행될 문장들  
    // (조건식1의 결과는 false)  
} else if(조건식3) {  
    // 조건식3의 결과가 true일 때 수행될 문장들  
    // (조건식1과 조건식2의 결과는 false)  
} else {  
    // 모든 조건식의 결과가 false일 때 수행될 문장들  
}
```

```
if(score > 60) {  
    System.out.println("합격입니다.");  
}
```

```
if(score > 60)  
    System.out.println("합격입니다.");
```

```
if(score > 60) {  
    System.out.println("합격입니다.");  
} else {  
    System.out.println("불합격입니다.");  
}
```

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

```
if(score >= 90) {  
    System.out.println("A등급");  
} else if(score >= 80 && score < 90) { // 80<=score<90  
    System.out.println("B등급");  
} else if(score >= 70 && score < 80) { // 70<=score<80  
    System.out.println("C등급");  
} else { // score < 70  
    System.out.println("F등급");  
}
```

2. if문이란?

```
int i = 0;  
if( i % 2 == 0 ) { ..... }
```

```
char ch = ' ';  
if(ch >= '0' && ch <= '9' ) { ..... }
```

```
String str = "" ;  
if( str != null && !str.equals( "" ) {  
    ch = str.charAt(0);  
}
```

```
boolean power = false;
```

```
if( !power ) {  
    // power가 들어왔으면..  
}
```

3. 중첩 if문이란?

- 중첩 if문은 if문 안에 또 다른 if문을 중첩해서 넣을 수 있다.
- 중첩 if문은 개수가 제한이 없다.(하지만, 너무 많이 중첩을 하면, 가독성이 저해된다.)

```
if(score >= 90) {  
    // 중첩 if문(무제한적으로 중첩 if는 들어갈 수 있다.)  
    if(score >= 95)  
        grade = "A+";  
    else  
        grade = "A0";  
} |  
else {  
    if(score >= 85)  
        grade = "B+";  
    else  
        grade = "B0";  
}
```

4. switch문이란?

- if문의 조건식과 다르게, **조건식의 계산 결과가 int범위**

이하의 정수, 그리고 String만 값으로 들어감.

- 조건식의 결과와 일치하는 **case문으로 이동**
후, break문을 만날 때까지 실행한다.

단, break문이 없으면, switch문을 벗어나지
않고 계속 default까지 실행한다.

반드시, break문이 존재해야 한다.

- case문의 값으로 변수는 절대 사용할 수 없다.
(상수, 리터럴만 가능함)

```
char grade = 'B';  
switch(grade) {  
    case 'A':  
    case 'a':  
        System.out.println("우수 회원입니다.");  
        break;  
    case 'B':  
    case 'b':  
        System.out.println("일반 회원입니다.");  
        break;  
    //default는 옵션이다.  
    default:  
        System.out.println("손님입니다.");  
}
```

4. switch문이란?

- if문의 조건식과 다르게, **조건식의 계산 결과가 int범위**

이하의 정수, 그리고 String만 값으로 들어감.

- 조건식의 결과와 일치하는 **case문으로 이동**
후, break문을 만날 때까지 실행한다.

단, break문이 없으면, switch문을 벗어나지
않고 계속 default까지 실행한다.

반드시, break문이 존재해야 한다.

- case문의 값으로 변수는 절대 사용할 수 없다.
(상수, 리터럴만 가능함)

```
char grade = 'B';  
switch(grade) {  
    case 'A':  
    case 'a':  
        System.out.println("우수 회원입니다.");  
        break;  
    case 'B':  
    case 'b':  
        System.out.println("일반 회원입니다.");  
        break;  
    //default는 옵션이다.  
    default:  
        System.out.println("손님입니다.");  
}
```

5. 중첩 switch문이란?

- switch문의 안에 또 다른 switch문을 중첩해서 if문과 동일하게 넣을 수 있다.
- 아울러, 중첩 횟수는 제한이 없다.
- 또한, if문과 같이 섞어서 사용할 수도 있다.(단, 가독성 저해)

```
int num = 1;

switch(num) {
    case 1:
    case 7:
        switch(num) {
            case 1 :
                System.out.println("1이군요");
                break;
            case 7 :
                System.out.println("7이군요");
                break;
        }
}
```


6. if문과 switch문을 비교하자면?



- if문이 현업에서는 빈번하게 사용되어지며, 정형화된 수, 한정된 값 등 은 switch문을 사용하면 가독성이 좋다.
- 모든 switch문은 if문으로 변경 가능하지만, if문은 switch문으로 변경할 수 없는 경우가 대다수이다.
- 당연한 결과이겠지만, if문보다는 switch문이 가독성면에서는 더욱 좋다.
- 하지만, 현업에서는 딱 떨어지게끔 정해진 결과가 거의 없기 때문에 switch문은 찾아보기가 힘들다.

7. 자주 쓰는 메서드 설명 – Math.random()

* Math.random() 을 이용해서 특정 범위의 정수 난수(亂數) 얻기

- 원리 이해 (1~10까지의 임의의 정수 얻기)

```
0.0 <= Math.random() < 1.0
```

```
0.0 * 10 <= Math.random() * 10 < 1.0 * 10
```

```
(int) 0.0 * 10 <= (int) Math.random() * 10 < (int) 1.0 * 10  
0.0           (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)           10
```

```
(int) 0.0 * 10 + 1 <= (int) Math.random() * 10 + 1 < (int) 1.0 * 10 + 1  
1.0           ( 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)           11
```

- 공식

```
int num = (int) (Math.random() * n) + start
```

ex) 주사위 번호 뽑기

```
int num = (int)(Math.random() * 6) + 1
```

감사합니다.

