JAVA 3강 조건문

강사 박주병

1.if문

조건문이란 조건식이 참, 거짓에 따라 실행흐름이 분기 되는 것을 말한다.
조건문은 크게 if문과 switch문 2가지가 있다. 그중에서 먼저 if문을 보도록 하자.

```
int age = 30;


if(age>65)
{
    System.out.println("퇴직연금 수령나이입니다.");
}
```

if(조건식)
{
}
형태로 쓰이며 괄호안에 들어가는 조건식은 true, false 가 나와야 한다. true일 경우 중괄호 영역의 코드가 실행된다. false라면 중괄호 영역의 코드는 건너뛰게 된다.

위 그림의 예시대로라면 age의 값이 65보다 작아 false가되어 실행되지 않는다. 또한 중괄호 안의 내용은 tab으로 들여쓰기 하는 것을 권장한다.

1.1 if문의 권장사항

하지 않으면 에러는 아니지만 따르는 것이 가독성 측면에서 좋다. 아래의 그림을 참조하자.

if문의 내용이 한줄 이면 중괄호를 생략하여도 되지만 아래의 경우처럼 혼동이 생길 수 있으니 띄어쓰기와 줄넘김을 확실히 하자.

```
if(age>=0 && age<20 )
    System.out.println("미성년자입니다.");
System.out.println("if문에 포함된건가??");
```

If문에 포함된 문장이 아니다 따라서 무조건 실행된다. 주의하도록 하자!

1.2 if else

if문의 조건식이 false 일 때 실행된다. 아래의 경우 a가 13이라면 else문만 실행될 것이다. 그리고 else문 역시 한줄이라면 중괄호를 생략해도 된다.

```
int a = 10;

if(a >=10)
{
    System.out.println("10보다 같거나 크다.");
}
else
{
    System.out.println("10보다 작다");
}
```

그림5는 else 문에 또 다시 if 조건문을 준 형태이다 이처럼 첫 번째 if문이 false가 될 경우 else문에서 다시 조건식을 사용 할 수 있다. 이렇게 else if를 무한정 뒤에 붙여서 조건을 늘릴 수 있다.

```
int a = 10;

if(a>=10)
    System.out.println("10보다 같거나 크다.");
else if(a<10 && a>=5)
    System.out.println("5~9");
else if(a<5 && a>=0)
    System.out.println("0~4");
else
    System.out.println("음수입니다.");
```

그림 5

그림5의 조건을 자세히 보면 두 번째인 else if의 a<10 조건은 사실상 없어도 되는 조건이다.
왜냐하면 이미 첫 번째 if문에서 10보다 큰 경우는 실행되면서 제외되었기 때문이다.
즉 두 번째 else if 문이 실행된다는 것은 10보다 작아야지만 실행되는 것이다.

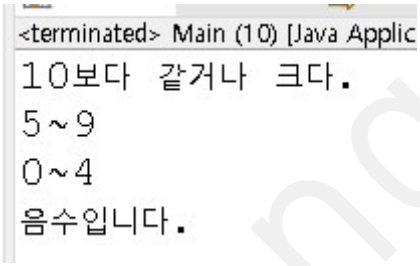
else if문은 엄연히 첫 번째 if문이 false 일 때 실행된다는 것을 알아둬야 한다.
그림6처럼 두 가지 코드는 서로 전혀 다른 결과를 출력한다.
(내용은 신경쓰지말자 그저 else가 있고 없고의 차이를 보여주기 위함이다.)

```java
int a = 10;

if(a>=10)
    System.out.println("10보다 같거나 크다.");
else if(a>=5)
    System.out.println("5~9");
else if(a>=0)
    System.out.println("0~4");
else
    System.out.println("음수입니다");
```

=

```java
int a = 10;

if(a>=10)
    System.out.println("10보다 같거나 크다.");
if(a>=5)
    System.out.println("5~9");
if(a>=0)
    System.out.println("0~4");
if(a<0)
    System.out.println("음수입니다");
```

그림 6



1.3 if문의 중첩

중괄호로 묶어놓는 문법들은 웬만하면 그 내부에 다시 중첩으로 해당 문법을 사용 할 수 있다.
if문 역시 마찬가지이다.

```java
if(score>10) //조건식1
{
    System.out.println("조건식1이 true인경우 ");
    if(score ==3) //조건식2
    {
        System.out.println("조건식2가 true인경우 ");
    }else
    {
        System.out.println("조건식1은 true이며 조건식2는 false인경우 ");
    }
}
else
{
    System.out.println("조건식1이 false인경우 ");
}
```

if문 역시 중첩으로 내부에 다시 if문과 else 혹은 else if 문을 얼마든지 사용 가능하다. 중첩의 개수에는 제한이 없다. 중첩을 쓸 경우 그림10 처럼 혼돈을 야기하는 코드는 주의를 하도록 하자. 이럴 경우 확실하게 중괄호를 적어주는 것 또한 좋다.

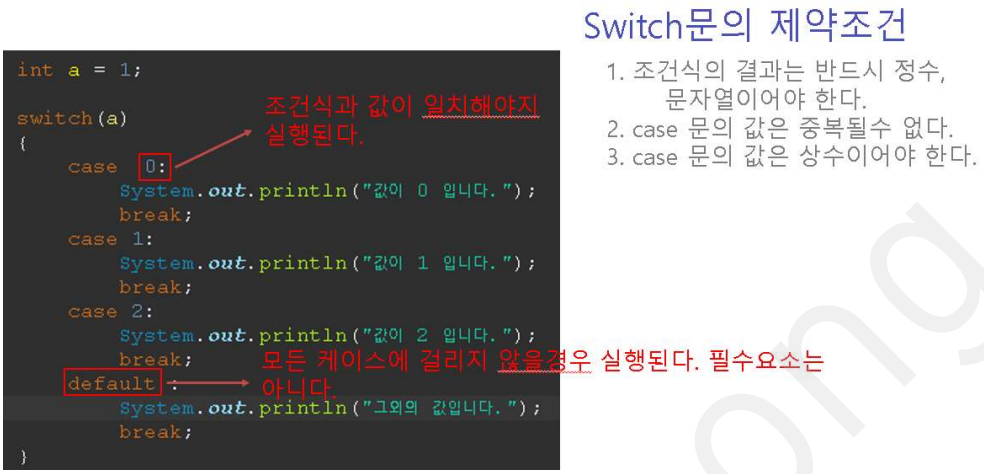
```java
if(score>10) //조건식1
    if(score ==3) //조건식2
        System.out.println("조건식2가 true인경우 ");
else
    System.out.println("조건식1이 false인경우 ");
```

중괄호가 없으면 항상 가장 가까운 if문의 else가 된다.

그림 10

2.switch문

switch문은 if문과 마찬가지로 실행 흐름을 분기하는 역할을 한다. 다만 if문의 조건식의 경우 true, false 가 나오는 식을 넣어야 했다면 switch는 오직 정수, 문자열만 사용가능하다. 그리고 해당 조건식의 값과 일치하는 case문 쪽으로 실행흐름이 넘어 가게 된다. case문에서 역시 true, false가 아닌 정수, 문자열만 올수 있으며 switch의 조건식에 들어갔던것과 같은 타입이여야 한다. 즉 조건식에 정수가 들어갔다면 case에도 정수가 들어가야 한다.
default는 모든 케이스가 실행되지 않을 때 실행된다.




```
int a = 1;

switch(a)
{
    case  0:
        System.out.println("값이 0 입니다.");
        break;
    case 1:
        System.out.println("값이 1 입니다.");
        break;
    case 2:
        System.out.println("값이 2 입니다.");
        break;
    default :
        System.out.println("그외의 값입니다.");
        break;
}
```

만약 break문을 뺀다면 a가 1일 경우 case1부터 case2가 실행되며 break문을 만날 때 까지 실행된다.

그림12처럼 case1의 break문을 제거 한다면 “값이 1입니다” ,“값이 2입니다.” 2가지가 실행될 것이다.

```
int a = 1;

switch(a)
{
    case  0:
        System.out.println("값이 0 입니다.");
        break;
    case 1:
        System.out.println("값이 1 입니다.");
        break;
    case 2:
        System.out.println("값이 2 입니다.");
        break;
    default :
        System.out.println("그외의 값입니다.");
        break;
}
```

break; 문을 제거하고 실행해보자

그림 12

이런 것을 일부러 이용 할 때도 있다. 아래처럼 특정값 이상이면 순차적으로 아래 단계의 코드들 모두 실행해야 될 경우 일부러 break문을 제거하기도 한다.

```
int weaponLevel = 3;
int damage=100;


switch(weaponLevel)
{
    case 3:
        damage +=100;
    case 2:
        damage +=10;
    case 1:
        damage +=1;
}
```

단계적으로 코드를 실행해야 할때 일부러 break를 빼서 활용하기도 한다.