

Java - 조건문과 반복문

컴소과 동아리 REFERENCE

제작자: 김명준

목 차

- 조건문
- 반복문

조건문

조건문

if문, switch문이 존재

처리해야 할 경우의 수가 많을 때는 if문보다 switch문이 효율적이지만, 제약이 많음

if문

```
if(조건식) {  
    // 조건식이 참(true)일 때 수행될 문장들을 적는다.  
}
```

조건식은 일반적으로 비교연산자와 논리연산자로 구성

블럭 {}은 가능하면 생략하지 않고 사용 -> **제어문은 다음 문장 하나만 실행**

조건문

if-else문

```
if(조건식) {  
    // 조건식이 참(true)일 때 수행될 문장들을 적는다.  
} else {  
    // 조건식이 거짓(false)일 때 수행될 문장들을 적는다.  
}
```

if-else if문

```
if(조건식1) {  
    // 조건식1이 참(true)일 때 수행될 문장들을 적는다.  
} else if(조건식2) {  
    // 조건식1이 거짓(false)이고 조건식2가 참(true)일 때 수행될 문장들을 적는다.  
} else {  
    // 위의 어느 조건식도 거짓(false)일 때 수행될 문장들을 적는다.  
}
```

조건문

switch문

```
switch(조건식) {  
    case 값1:  
        // 조건식의 결과가 값1인 경우 수행될 문장들  
        break;  
    case 값2:  
        // 조건식의 결과가 값2인 경우 수행될 문장들  
        break;  
    ...  
    default:  
        // 조건식의 결과와 일치하는 case문이 없을 때 수행될 문장들  
        break;  
}
```

switch문의 조건식 결과는 정수 또는 문자열이어야 함

case문의 값은 정수 상수만 가능하며, 중복되지 않아야 함

반복문

반복문

어떤 작업이 반복적으로 수행되도록 할 때 사용

반복문의 종류

- for문
- while문
- do-while문

반복문

for문

```
for(초기화; 조건식; 증감식) {  
    // 조건식이 참일 때 수행될 문장들을 적는다.  
}
```

순서: 초기화 -> 조건식 -> 수행될 문장 -> 증감식 -> 조건식 -> ...

향상된 for문

```
for(타입 변수명 : 배열 또는 컬렉션) {  
    // 반복할 문장  
}
```

반복문

while문

```
while(조건식) {  
    // 조건식의 연산 결과가 참(true)인 동안, 반복될 문장들을 적는다.  
}
```

while문의 조건식은 생략 불가

do-while문

```
do {  
    // 조건식의 연산 결과가 참(true)일 때 수행될 문장들을 적는다.  
} while(조건식);
```

최소한 한 번은 수행될 것을 보장

그 외

break문

자신과 제일 가까운 제어문을 빠져 나옴

continue문

자신과 제일 가까운 제어문의 끝으로 이동 -> **continue문 다음 문장들은 수행 X**

감사합니다

컴소과 동아리 REFERENCE