

제어문





1. 조건문



조건문

조건문

→ 조건에 따라서 수행되는 코드가 달라지는 구문

조건문 종류

1. if
 - if
 - if ~ else
 - if ~ else if
2. switch



조건문

if 문

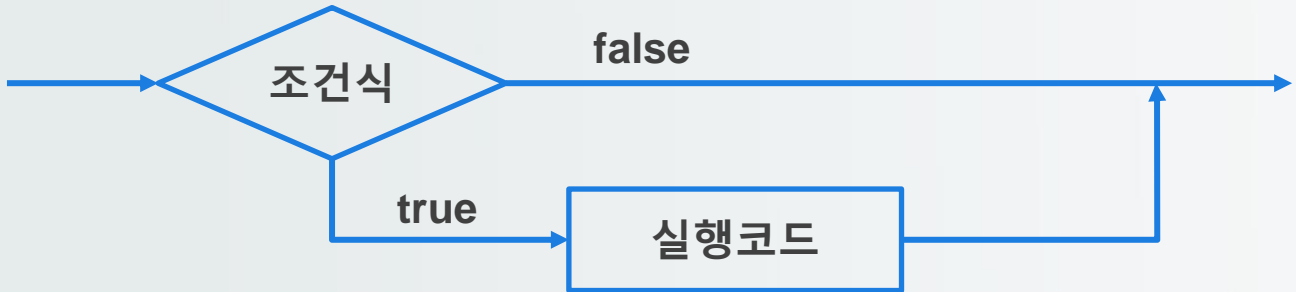
→ 조건식의 결과 값이 true면 {}안의 내용을 실행하고 false면 실행하지 않는 구문

if문 표현식

```
if(조건식){
```

```
    조건식이 true인 경우 수행 할 코드
```

```
}
```



조건문

if 문

예제 1)

```
Scanner sc = new Scanner(System.in);  
System.out.print("10을 입력해 보세요 : ");  
int number = sc.nextInt();  
if(number == 10){  
    System.out.println("잘했어요");  
}
```



조건문

if ~ else 문

→ 조건식의 결과 값이 true면 if의 { }안의 내용을 실행하고 false면 else의 { } 안의 내용을 실행

if문 표현식

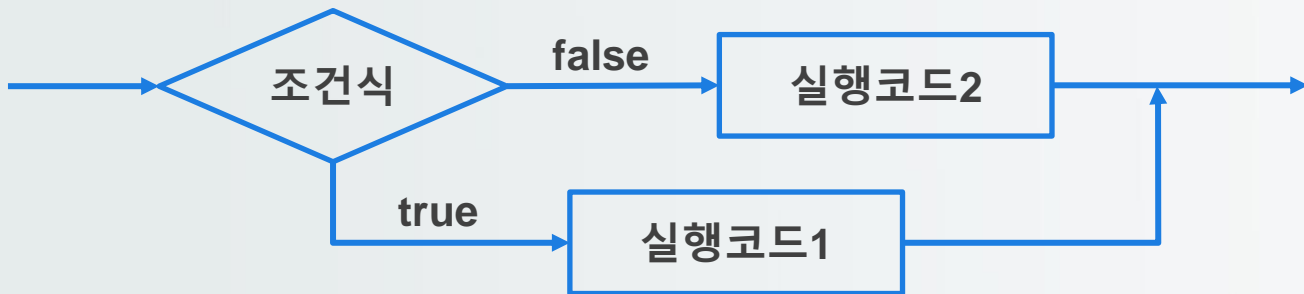
```
if(조건식){
```

```
    조건식이 true인 경우 수행 할 코드
```

```
}else{
```

```
    조건식이 false인 경우 수행 할 코드
```

```
}
```



조건문

if ~ else 문

예제 1)

```
Scanner sc = new Scanner(System.in);  
System.out.print("정수를 입력하세요 : ");  
int number = sc.nextInt();  
if(number%2 == 0){  
    System.out.println("짝수입니다.");  
}else{  
    System.out.println("홀수입니다.");  
}
```



조건문

if ~ else if 문

→ 조건식1의 결과 값이 true면 if의 { }안의 내용을 실행하고 false면 else if의 조건식을 확인하여 결과가 true이면 { } 안의 내용을 실행하고 false이면 else의 { }안의 내용을 실행

if문 표현식

```
if(조건식1){
```

조건식1이 true인 경우 수행 할 코드

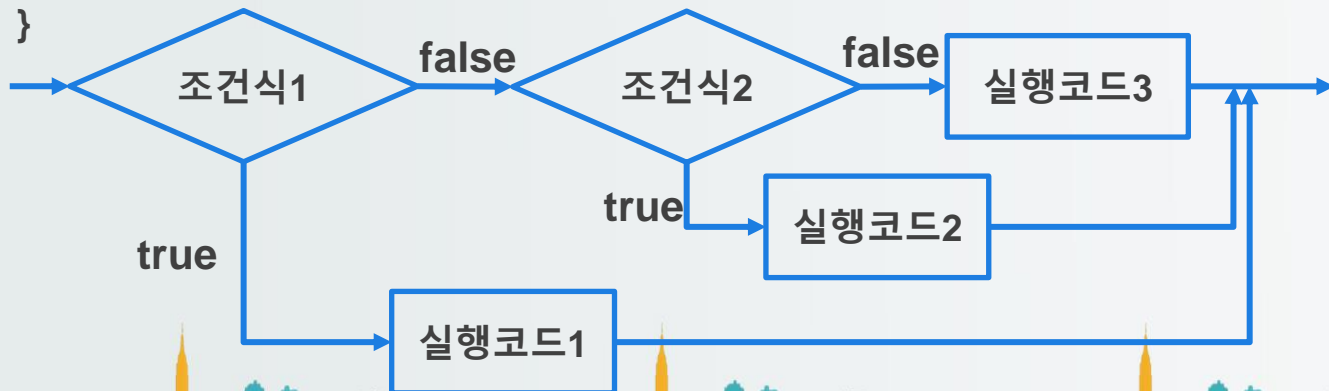
```
}else if(조건식2){
```

조건식1이 false이며 조건식2가 true인 경우 수행 할 코드

```
}else {
```

조건식1, 조건식2가 모두 false면 수행 할 코드

```
}
```



조건문

if ~ else if 문

예제 1)

```
Scanner sc = new Scanner(System.in);
System.out.print("1~3 사이 수를 입력하세요 : ");
int number = sc.nextInt();
if(number == 1){
    System.out.println("One");
}else if(number == 2){
    System.out.println("Two");
} else if(number == 3){
    System.out.println("Three");
}else{
    System.out.println("잘못입력했습니다");
}
```



조건문

switch 문

→ 정수, 문자, 문자열에 있는 값과 일치한 case문을 실행하고, break;를 이용해서 switch문을 빠져 나가는 구문(if ~ else if 문과 유사)

switch문 표현식

```
switch(변수 or 계산식){  
    case 리터럴값1 : 실행코드1;break;  
    case 리터럴값2 : 실행코드2;break;  
    case 리터럴값3 : 실행코드3;break;  
    case 리터럴값4 : 실행코드4;break;  
    default : 실행코드5;break;  
}
```



조건문

switch 문

예제 1)

```
Scanner sc = new Scanner(System.in);  
System.out.print("1~3 사이 수를 입력하세요 : ");  
int number = sc.nextInt();  
switch(number){  
    case 1 : System.out.println("1을 입력");break;  
    case 2 : System.out.println("2를 입력");break;  
    case 3 : System.out.println("3을 입력");break;  
    default: System.out.println("잘못입력하셨습니다");  
}
```





2. 반복문



반복문

반복문

→ 사용자가 원하는 만큼 명령들을 여러 번 반복해서 실행하는 구문

반복문 종류

1. for
2. while
 - while
 - do ~ while



반복문

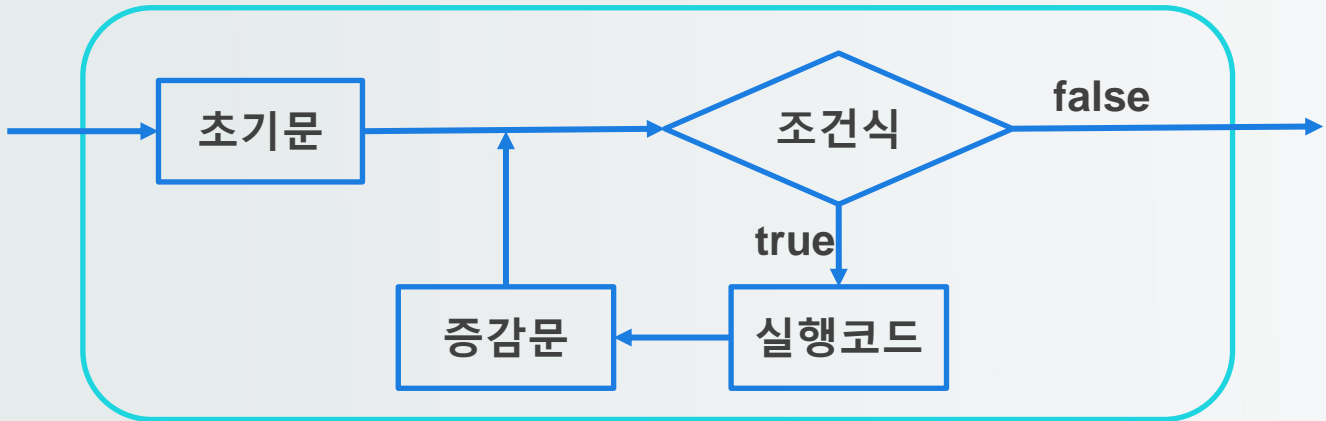
for

→ 가장 보편적인 반복문으로 가독성이 높고, 초기,조건,증감문을 한번에 선언해서 사용

for문 표현식

```
for(초기문;조건문;증감문){  
    실행문;  
}
```

for



반복문

for 문

예제 1)

```
for(int i=1;i<=5;i++){  
    System.out.println(i + “번째 반복해서 수행 중”);  
}
```



반복문

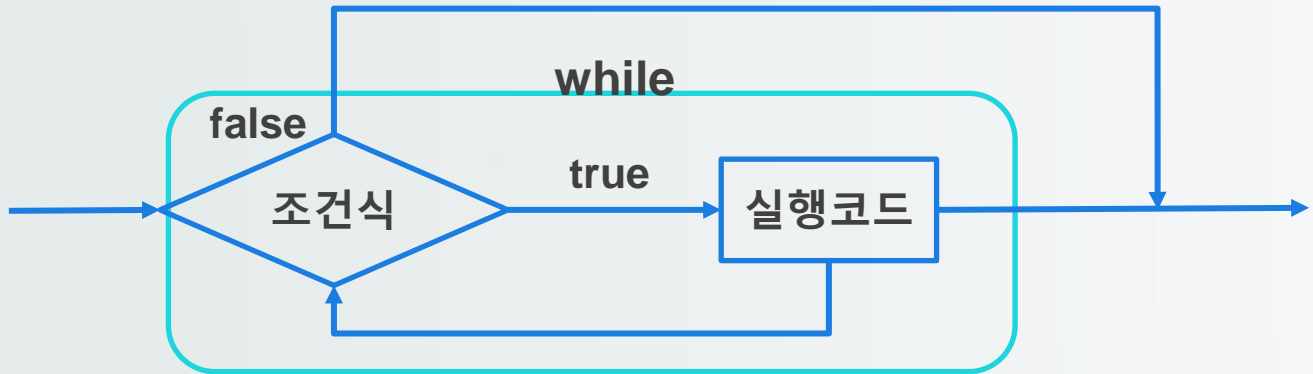
while

→ 특정 조건이 만족되는 동안 명령을 계속해서 반복적으로 수행함

→ for문과 다르게 초기식을 미리 작성하고, 증감문도 실행코드 내부에 직접 작성해야 함

while문 표현식

```
while(조건문){  
    실행문;  
}
```



반복문

while

예제 1)

```
int i =1;           //초기값 지정
while(i<=5){        //조건문
    System.out.println(i + “번째 반복해서 수행 중”);
    i++;             //실행코드에 증감문을 직접 작성
}
```



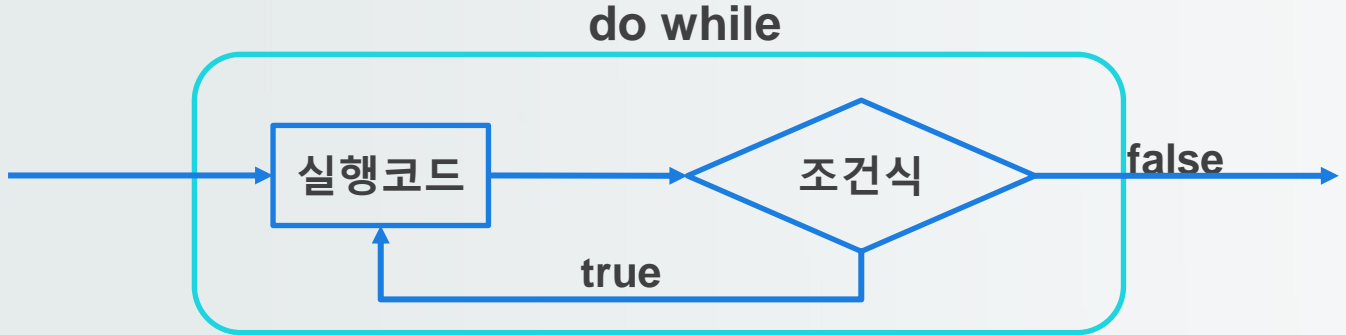
반복문

do while

→ while문과 유사하지만 조건문이 뒤에 위치하기 때문에 조건문에 맞지 않더라도 1번은 무조건 실행

do while문 표현식

```
do{  
    실행문;  
} while(조건문);
```



반복문

do while

예제 1)

```
Scanner sc = new Scanner(System.in);  
System.out.print ("정수 입력 : ");  
int num = sc.nextInt();  
do{  
    System.out.println(num%10);  
    num /= 10;  
}while(num>0);
```



반복문

중첩 for 문

→ for문안에 for문이 들어간 형태로 내부의 명령을 외부의 횟수만큼 반복하게 됨

예제 1)

```
for(int i=1;i<=5;i++){  
    for(int j=1;j<=3;j++){  
        System.out.println("i = "+i+" / j = "+j);  
    }  
}
```



반복문

무한루프(무한 반복문)

→ 반복 횟수가 미리 정해져 있지 않고, 무한히 반복되는 반복문

무한루프 생성 코드

1. for

```
for(;;){  
    실행코드;  
}
```

//초기,조건,증감을 비워 둔 채로 작성

2. while

```
while(true){  
    실행코드;  
}
```

//조건식 자리에 논리형 true값을 입력

※ 보편적으로 무한루프 사용시에는 while(true)방식 사용





3. 분기문



분기문

분기문

→ 반복문의 실행흐름을 제어하는 구문

분기문 종류

1. break

- 반복문을 빠져 나가는 구문
- 반복문 수행 중 break;구문을 만나는 경우 곧장 반복문을 빠져 나옴

2. continue

- 반복문 수행 중 continue;를 만나는 경우 아래를 실행하지 않고 반복문을 다시 실행
- for문의 경우 증감식으로 이동



분기문

break

예제 1)

```
Scanner sc = new Scanner(System.in);
System.out.print ("숫자 입력 : ");
int num = sc.nextInt();
int sum = 0;           //총 합계를 저장 할 변수
int i = 1;             //초기값
while(true){
    sum += i;           //합계에 현재 i값을 더함
    if(i == num){       //입력한 수와 i가 같으면
        break;          //반복문 종료
    }
    i++;                //증감식
}
```



분기문

continue

예제 1)

```
Scanner sc = new Scanner(System.in);
System.out.print ("숫자 입력 : ");
int num = sc.nextInt();
int sum = 0;          //총 합계를 저장 할 변수
for(int i=1;i<=num;i++){
    if(i%2 != 0){      //홀수인 경우
        continue;     //합계에 더하지 않고 증감으로
    }
    sum += i;
}
System.out.println(sum); //1부터 입력한 수까지 짝수의 합
```

