

제어문

제어문이란?

• 제어문

프로그램의 흐름을 제어한다고 하여 제어문이라고 함

• 제어문의 종류

1. 조건문 : if, switch
2. 반복문 : for, while, do-while
3. 제어문 : break, continue

* 조건문, 반복문 안에서 또 다른 흐름을 제어하는 역할

제어문(조건문)

조건문이란?

- 조건문

특정 조건을 검사하여, 참(true)이면 코드를 실행하고 거짓(false)이면 실행하지 않음

if문

• if 문

어떠한 조건이 만족되었을 때 명령을 수행할 수 있도록 해주는 문법
개발자가 가장 자주 사용하는 제어문으로, 특정 범위나 여러가지 케이스에 대한 대응이 가능함

• if문

```
if(조건식) {  
    참(true)일 때 실행할 코드  
}
```

- 조건이 참(true)일 경우 {} 안의 코드가 실행
- 조건이 거짓(false)일 경우 실행되지 않음

• 예시

```
boolean isTrue = true;  
  
if(isTrue == true) {  
    System.out.println("조건식의 결과는 true 입니다.");  
}
```


if문 활용하기 - 1

- 설명

스캐너를 통해 입력된 값이 10보다 클 경우 "입력된 숫자는 10보다 큼니다"가 출력되고, 10보다 작거나 같을 경우 아무런 메시지도 출력되지 않음

```
Scanner sc = new Scanner(System.in);
System.out.println("숫자를 입력 하세요 : ");
int num = sc.nextInt();

if(num > 10) {
    System.out.println("입력된 숫자는 10보다 큼니다.");
}
```



숫자를 입력 하세요 :
11
입력된 숫자는 10보다 큼니다.

if-else if문, if-else문

• if~else if

```
if(조건식1) {  
    첫번째 조건식이 참(true)일 때 실행할 코드  
}  
else if(조건식2) {  
    두번째 조건식이 참(true)일 때 실행할 코드  
}  
else if(조건식3) {  
    세번째 조건식이 참(true)일 때 실행할 코드  
}
```

- 위에서부터 확인하여 조건이 참(true)일 경우 { } 안의 코드가 실행
- 모든 조건이 거짓(false)일 경우 실행되지 않음

• if~else if~else

```
if(조건식1) {  
    첫번째 조건식이 참(true)일 때 실행할 코드  
}  
else if(조건식2) {  
    두번째 조건식이 참(true)일 때 실행할 코드  
}  
else if(조건식3) {  
    세번째 조건식이 참(true)일 때 실행할 코드  
}  
else {  
    위의 모든 조건이 거짓(false)일 때 실행할 코드  
}
```

- 모든 조건이 거짓(false)일 경우 else가 실행

if문 활용하기 - 2

- 설명

입력된 값이 10일 경우, **첫번째 조건식에 만족하지 않아 else if의 조건식을 확인**하고 참(true)이기 때문에 "입력된 숫자는 10입니다."가 출력

```
if(num > 10) {  
    System.out.println("입력된 숫자는 10보다 큼니다.");  
} else if(num == 10) {  
    System.out.println("입력된 숫자는 10입니다.");  
}
```

숫자를 입력 하세요 :

10

입력된 숫자는 10입니다.

if문 활용하기 - 3

• 설명

입력된 값이 7일 경우, 위의 모든 조건식이 거짓이기 때문에 else에 있는 "입력된 숫자가 10보다 작습니다."가 출력

```
Scanner sc = new Scanner(System.in);
System.out.println("숫자를 입력 하세요 : ");
int num = sc.nextInt();

if(num > 10) {
    System.out.println("입력된 숫자는 10보다 큼니다.");
} else if(num == 10) {
    System.out.println("입력된 숫자는 10입니다.");
} else {
    System.out.println("입력된 숫자가 10보다 작습니다.");
}
```



숫자를 입력 하세요 :

7

입력된 숫자가 10보다 작습니다.

if문 활용하기 - 4

• 설명

입력된 값이 9일 경우 "1보다는 크다. 그리고 10보다는 작다"는 참(true)이기 때문에 "입력된 숫자는 1보다 크고 10보다 작습니다."가 출력

```
Scanner sc = new Scanner(System.in);
System.out.println("숫자를 입력 하세요 : ");
int num = sc.nextInt();

if(num > 1 && num < 10) {
    System.out.println("입력된 숫자는 1보다 크고 10보다 작습니다.");
} else if(num == 10) {
    System.out.println("입력된 숫자는 10입니다.");
} else {
    System.out.println("입력된 숫자가 10보다 작습니다.");
}
```

숫자를 입력 하세요 :

9

입력된 숫자는 1보다 크고 10보다 작습니다.

if문 활용하기 - 5

• 설명

입력된 값이 10 또는 11일 경우 첫번째 조건식에 만족하지 않고,
"10과 같다. 또는 11과 같다" 에 참(true)이기 때문에 해당 메시지 출력

```
Scanner sc = new Scanner(System.in);
System.out.println("숫자를 입력 하세요 : ");
int num = sc.nextInt();

if(num > 1 && num < 10) {
    System.out.println("입력된 숫자는 1보다 크고 10보다 작습니다.");
} else if(num == 10 || num == 11) {
    System.out.println("입력된 숫자는 10 또는 11과 같습니다.");
} else {
    System.out.println("입력된 숫자가 10보다 작습니다.");
}
```

숫자를 입력 하세요 :
10
입력된 숫자는 10 또는 11과 같습니다.

```
Scanner sc = new Scanner(System.in);
System.out.println("숫자를 입력 하세요 : ");
int num = sc.nextInt();

if(num > 1 && num < 10) {
    System.out.println("입력된 숫자는 1보다 크고 10보다 작습니다.");
} else if(num == 10 || num == 11) {
    System.out.println("입력된 숫자는 10 또는 11과 같습니다.");
} else {
    System.out.println("입력된 숫자가 10보다 작습니다.");
}
```

숫자를 입력 하세요 :
11
입력된 숫자는 10 또는 11과 같습니다.

if문 활용하기 - 6

- Q1. 입력 받은 값이 0 또는 12 이상일 경우,
"값이 너무 크거나 작습니다."를 출력 하시오.

1. 스캐너로 사용자로부터 숫자를 입력 받으세요.
2. 위의 예시를 활용하여 OR 연산자 또는 AND 연산자를 활용하여 문제를 풀어보세요.

if문 활용하기 - 7

- Q2. 입력 받은 월에 따라 계절을 출력 하시오.

1. 스캐너로 사용자로부터 몇 월인지 입력 받으세요.
2. 해당하는 계절과 출력 메시지는 아래와 같습니다.

3, 4, 5 : 봄

6, 7, 8 : 여름

9, 10, 11 : 가을

12, 1, 2 : 겨울

그 외 : 해당하는 계절이 없습니다.

ex) 1을 입력 받을 경우 "겨울" 출력

3. 위의 예시를 활용하여 OR 연산자 또는 AND 연산자를 활용하여 문제를 풀어보세요.

switch문

- switch문

특정 숫자, 문자에 대한 값을 받을 때 주로 사용함

ex) 메뉴 1~6번에서 누르세요.

- switch문

```
switch(값을 가져올 변수) {  
  case 1 :  
    숫자 1을 입력했을 때 실행할 코드  
    break;  
  case 2 :  
    숫자 2를 입력했을 때 실행할 코드  
    break;  
  case 3 :  
    숫자 3을 입력했을 때 실행할 코드  
    break;  
  default :  
    그 외 모든 값을 실행할 코드  
}
```

- 값을 가져올 변수에서 숫자 1, 2, 3일 경우에 대한 내용을 처리하고 그 외에 대한 값은 default 에서 처리함

switch문 예시

```
Scanner sc = new Scanner(System.in);
System.out.println("메뉴를 선택하세요.");
System.out.println("1 : 아이스 아메리카노");
System.out.println("2 : 카페라떼");
System.out.println("3 : 딸기요거트 스무티");
int num = sc.nextInt();
switch(num) {
    case 1 :
        System.out.println("아이스 아메리카노를 선택하셨습니다.");
        break;
    case 2 :
        System.out.println("카페라떼를 선택하셨습니다.");
        break;
    case 3 :
        System.out.println("딸기요거트 스무티를 선택하셨습니다.");
        break;
    default :
        System.out.println("해당하는 메뉴가 없습니다.");
}
```

메뉴를 선택하세요.

1 : 아이스 아메리카노

2 : 카페라떼

3 : 딸기요거트 스무티

1

아이스 아메리카노를 선택하셨습니다.

메뉴를 선택하세요.

1 : 아이스 아메리카노

2 : 카페라떼

3 : 딸기요거트 스무티

0

해당하는 메뉴가 없습니다.

break를 사용하는 이유

• 설명

switch문에서 break를 사용 하는 건, 조건에 해당할 경우 switch문을 종료하겠다는 뜻으로 break를 사용하지 않을 경우 fall-through가 발생함

- switch문에서의 fall-through(통과) : case를 실행한 후 그 다음 case로 실행되는(통과되는) 현상

즉, 조건에 만족 하는것과 별개로 다음 case가 실행되며 마지막 default의 경우 더 이상 실행 할 case가 없으므로 사용하지 않아도 됨

* 프로그램의 흐름을 제어하기 때문에 break 또한 제어문에 속함

switch문 활용하기

- Q1. 사용자로부터 1부터 7 사이의 숫자를 입력 받아 주말인지, 평일인지 출력하는 프로그램을 작성하시오.

1. 스캐너로 사용자로부터 숫자를 입력 받으세요.
2. 1~5를 입력 받으면 "평일입니다."를 출력하고, 6~7을 입력 받으면 "주말입니다."를 출력하시오.