

22.11.16 - continue , break

복습

```
// 조건 변수 -> 정수, 문자, 문자열

char bar = 'a';

switch (bar) {
// 현재 bar 변수안에 들어 있는 값에 따라 선택하는 문장을 작성한다.

case 'a' :
    system.out.println(1);
    break;
case 'b' :
    system.out.println(2);
    break;
case 'c' :
    system.out.println(3);

default:
    system.out.println(4);

}
```

break文

(하나의 실행문, 조건식과 관계 x)

1

```
// continue break
// continue : 계속하다
// break : 멈추다 , 탈출하다

// ※ break 문을 만나면 실행흐름을 멈추고 첫 번째 반복문을 만날 때 까지 위로 올라간다.
```

```
// ※ 만나면 그 반복문을 탈출한다

Scanner scan = new Scanner(System.in);

int inputValue = 0;

while (true) {

    inputValue = scan.nextInt();

    if (inputValue > 0)
        break;           // 여기서 만나면 포함되어 있는 제일 상위 반복문으로
    System.err.println("양수만 입력하세요");

}
System.out.println(inputValue);
```

2 - 양수 사용 break 문

```
// ※ break 문을 만나면 실행흐름을 멈추고 첫 번째 반복문을 만날 때 까지 위로 올라간다.
// ※ 만나면 그 반복문을 탈출한다

// 조건이 많아지면 while loop 사용

Scanner scan = new Scanner(System.in);

int inputValue = 0;

for (;;) {
    while (true) {

        inputValue = scan.nextInt();

        if (inputValue > 0)
            break;
        System.err.println("양수만 입력하세요");

    }
}

System.out.println(inputValue);
```

3 - if / else 사용 break 문

```
Scanner scan = new Scanner(System.in);
```

```

int inputValue = 0;

while (true) {

    inputValue = scan.nextInt();

    if (inputValue > 0)
        break;

    // 잘못된 입력 값 처리

    String Msg = "잘 못된 입력 값입니다. 현재 값";
    if(inputValue == 0)
        Msg += "0";
    else
        Msg += "음수 입니다.";

    System.err.println("양수만 입력하세요");

}

System.out.println(inputValue);

```

4 - Loop and break Test

```

for (;;) {
    int foo = 0;
    int bar = 2;

    while (true) {

        if (foo > 2)
            break; // 탈출

        for (;;) {
            if (bar > 4)
                break; // 탈출

            bar++;
            System.out.println(1);
        }

        foo++;
        System.out.println(2);
    }

    System.out.println(3);
}

```

```

for (;;) {
    int foo = 0;
    int bar = 2;

    while (true) {

        if (foo > 2)
            break; // 탈출

        for (;;) {
            if (bar > 4)
                break; // 탈출

            bar++;
            System.out.println(1);
        }

        foo++;
        System.out.println(2);
    }

    System.out.println(3);
}

```

出力

1

1

1

2

2

2

3

ずっと繰り返
される

5 - continue / continue 아래에 있는 문장을 실행하지 않는다

```
for (int i = 0; i < 10; i++) {  
    if (i % 2 == 0)  
        continue;  
    System.out.println(i);  
}
```

6 반복문은 유지하지만 특정 조건에 해당 반복문 안을 실행 시키지 않고 싶을 때

```
// 1. 실행을 멈추고 위로 올라가 첫 번째 만나는 반복문을 찾는다  
// 2. 계속 반복 실행  
// 특정 조건의 해당문장이 실행 되지 않게 할 때 자주 사용  
// continue 아래에 있는 문장을 실행하지 않는다  
  
// 3, 4, 6, 8 단은 제외  
  
for (int i = 2; i < 10; i++) {  
    for(int j =1 ; j <10 ; j++) {  
  
        if (i == 3 || i == 4 || i == 6 || i == 8 )  
            continue;  
        System.out.println(i + "x" + j + "=" + i * j);  
    }  
}
```

7

```
// a = 97  
char letter = 97 ;  
  
// 반복문 작성  
// 97에서 3을 더하면 출력 하지 마라
```

```
for (char index = letter; index <= 122; index += 3) {  
    System.out.print(index);  
}  
  
for (char index = 'a'; index <= 'z'; index += 3) {  
    System.out.print(index);  
}  
  
// A = 65 ~ Z = 90  
  
char big = 90;  
for (char indexa = 'Z'; indexa >= 'A'; indexa-=1) {  
    System.out.print(indexa);  
}
```