

1주차(2)

제어문(control statement)

프로그램의 흐름을 제어하는 문장.

분기문과 반복문으로 나뉜다.

분기문 : if, switch

반복문 : for, while, do-while

if문

단순 if문 :

```
int n = 50;
```

```
String str = null;
```

```
//String은 처음이지? 일단 똑같이 쓰고 쌍따옴표 안에 문자열 넣는거야
```

```
//String의 기본값은 null, String은 기본자료형이 아니라는것만 알려두자.
```

```
if(n == 50){
```

```
    str = "n은 50입니다.";
```

```
}
```

```
//괄호안의 값이 true일 경우엔 그 아래쪽 연산을 수행
```

```
//괄호안의 값이 false일 경우엔 아래쪽 연산을 수행하지 않음
```

```
//괄호안에는 무조건 true나 false의 결과를 가지는 연산이 들어와야 함
```

```
//위에 결과 보여주고 if추가, n값 51로 변경후 다시 보여주기
```

```
if(n != 50){
```

```
    str = "n은 50이 아닙니다.";
```

```
}
```

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

if문

if ~ else문 :

```
int n = 49;
String str = null; //
```

```
if(++n >= 50){
    str = "n은 50이상의 수";
}
else{
    str = "n은 50미만의 수";
}
```

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

문제 :

변수 age에 나이를 대입하고, 30세 이상이면...
“드실만큼 드셨네요”, 그렇지 않으면 “더 드세요”를 출력하는 if문을 구현 한 후
마지막으로 “감사합니다”라는 문장을 출력해보자.

풀이 :

```
int age = 30;

if(age >= 30){

    System.out.println("드실만큼 드셨네요ㅋ");
}else{

    System.out.println("좀 더 드셔야겠어요ㅋ");
}

System.out.println("감사합니다.");
```

if문

다중 if문 (else if) : 여러개의 조건 비교가 필요할 경우

```
int num = 75;
String str = "";
```

```

if(num >= 90)
    str = "수";
else if(num >= 80)
    str = "우";
else if(num >= 70)
    str = "미";
else if(num >= 60)
    str = "양";
else
    str = "가";

System.out.println(str);

```

Switch - case문

if문과 비슷하지만 if문은 괄호안에 인자값이 true, 혹은 false로 결정되는 조건식이 들어가야 하고

Switch문은 인자값으로 조건이 아닌 비교할 값이 들어가야 한다.

특정 값을 바로 찾아 들어가기 때문에 if문에 비해 처리속도가 빠르다.

```
int n = 1;
```

```
switch (n) { //인자로 비교할 값이 들어와야 한다.
```

```
case 1: //인자와 비교할 조건값이 들어온다.
```

```
System.out.println("1. 게임하기");
```

```
break;
```

```
case 2:
```

```
System.out.println("2. 게임소개");
```

```
break;
```

```
case 3:
```

```
System.out.println("3. 종료");
```

```
break;
```

```
default:
```

```
System.out.println("메뉴선택이 올바르지 않습니다.");
```

```
break;
```

```
}
```

한 개 더!!

```
String str = "미";
```

```
String result;
```

```
switch (str) { //인자로 비교할 값이 들어와야 한다.
```

```
case "수"://인자와 비교할 조건값이 들어온다.
```

```
result = "90 ~ 100점";
```

```
break;
```

```
case "우"://콜론이다. 세미콜론 아니다.
```

```
result = "80 ~ 89점";
```

```
break;
```

```
case "미":
```

```
result = "70 ~ 79점";
```

```
break;
```

```
case "양":
```

```
result = "60 ~ 69점";
```

```
break;
```

```
case "가":
```

```
result = "59점 이하";
```

```
break;
```

```
default:
```

```
result = "제대로 입력하지지";
```

```
break;
```

```
}
```

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

-----예제 1

일단 Scanner객체를 이용해 키보드에서 값을 받는 예제를 먼저 보여주자.

이 예제 이후에 직접 Scanner객체를 이용하여 아래의 계산기 문제를 출제

```
public class ScannerEx {
```

```
public static void main(String[] args) {
```

```
// 키
```

보드의 값을 입력받기 위한 Scanner클래스.

```
// 클
```

래스 개념은 곧 배울테니 이렇게 있다는 것만 알고가자

//System.in을 통해 키보드의 입력값을 처리한다.

//Scanner를 통해 키보드에서 정수를 입력받자

```
System.out.println("정수를 입력하세요 : ");
```

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

```
int n  
= scan.nextInt();
```

```
System.out.println("n = " + n);
```

```
//문  
자를 입력하는데, 띄어쓰기를 포함한 문자를 입력해보자.
```

```
System.out.println("문자를 입력하세요 : ");
```

```
Scanner scan2 = new Scanner(System.in);
```

```
String str = scan2.next();
```

```
System.out.println("str = " + str);
```

```
//문
```

자를 입력하는데, 띄어쓰기를 포함한 문자를 입력해보자.

```
System.out.println("문자를 입력하세요 : ");
```

```
Scanner scan3 = new Scanner(System.in);
```

```
String str2 = scan3.nextLine();
```

```
System.out.println("str2 = " + str2);
```

```
//next()와 nextLine()설명
```

```
}
```

문제 :

Scanner와 Switch를 이용한 계산기(같이 만든다.)

키보드에서 정수 두 개를 입력받는다.

그리고 연산자를 입력받는다.

마지막으로 입력받은 연산자를 (switch-case문으로) 구별하여
입력받은 정수의 연산을 수행하는 코드를 작성해보자.

실행결과 :

첫 번째 숫자 입력 : 10

두 번째 숫자 입력 : 20

연산자 입력 : +

결과 : 10 + 20 = 30

풀이 :

```
int num1, num2;
```

```
String str = "";
```

```
String result = "";
```

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

```
System.out.print("첫번째 숫자 입력 : ");
```

```
num1 = scan.nextInt();
```

```
System.out.print("두번째 숫자 입력 : ");
```

```
num2 = scan.nextInt();
```

```
System.out.print("연산자 입력 : ");
```

```
Scanner scan2 = new Scanner(System.in);
```

```
str = scan2.next();
```

```
switch (str) {
```

```
case "+":
```

```
result = num1 + " + " + num2 + " = " + (num1 + num2);
```

```
//(num1 + num2)괄호로 안묶으면 결과값이 더해지지 않고 문자열로 붙어서 나온다.
```

```
break;
```

```
case "-":
```

```
result = num1 + " - " + num2 + " = " + (num1 - num2);
```

```
break;
```

```
case "*":
```



```
result = num1 + " * " + num2 + " = " + (num1 * num2);
```

```
break;
```

```
case "/":
```

```
result = num1 + " / " + num2 + " = " + (num1 / num2);
```

```
break;
```

```
default:
```

```
result = "올바른 연산자가 아닙니다.";
```

```
break;
```

```
}
```

```
System.out.println("결과 : " + result);
```

반복문

특정 수행문을 여러번 반복할 수 있도록 해주는 제어문

for문 : 특정 명령을 원하는 만큼 반복적으로 처리할 때 사용한다.

```
for(int i = 0; i <= 3; i++){
```

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

```
//for문 돌아가는 구조 설명해주기.
```

문제 : 1부터 10까지 반복하는 for문을 작성해보자.

```
for(int i = 1; i <= 10; i++){
```

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

```
}
```

문제 : 10부터 1까지 감소하는 for문을 작성해보자.

```
for(int i = 10; i >= 1; i--){
```

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

```
}
```

문제 - 1부터 10까지 반복하는 문장에서 3의 배수만 출력하는 제어력을 갖추어 보자!

```
for(int i=1; i<=10 ; i++){
```

```
    if(i%3 == 0)
```

```
        System.out.println(i + "은(는) 3의 배수입니다.");
```

```
    }//for의 끝!
```

```
// 보너스 난수발생
```

```
// 예) 1부터10까지 수 중 난수를 발생시키기  
// new Random().nextInt( (큰 수- 작은 수) + 1 ) + 작은 수  
// new Random().nextInt( (10 - 1) + 1 ) + 1  
int num = new Random().nextInt(10) + 1;  
System.out.println(num);
```

예제2 - 난수를 사용한 구구단 출력

```
//2 ~ 9사이의 난수를 입력받아 구구단 출력하기
```

```
int random = new Random().nextInt(8) + 2;
```

```
for(int i = 1; i <= 9; i++){
```

```
    System.out.println(random + " * " + i + " = " + (random * i));
```

```
}
```

난수로 알파벳 대문자들 중 하나를 출력하는 문제!!

```
//A - 65
```

```
//Z - 90
```

```
//결과:
```

```
//문자 : C
```

->방법1(아스키 코드를 이용하여 출력)

```
int random = new Random().nextInt(26) + 65;  
System.out.println("문자 : " + (char)random);
```

->방법2(문자를 이용하여 출력)

```
int random = new Random().nextInt('Z' - 'A' + 1) + 'A';  
System.out.println("문자 : " + (char)random);
```

구구단 문제

Scanner를 사용하여 2 ~ 9중 한개를 입력 받고

입력받은 숫자에 해당하는 구구단을 만들어서 출력해보자

단, 2 ~ 9 사이의 숫자가 입력되지 않았을 경우에는 “2 ~9 사이의 숫자를 입력하세요”라는 문장이 출력되도록 한다.

결과 :

단 입력 : 1

2 ~ 9 사이의 숫자를 입력하세요

단 입력 : 3

3 * 1 = 3

3 * 2 = 6

3 * 3 = 9

.

.

3 * 9 = 27

풀이 :

```
int dan;
```

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

```
System.out.print("단 입력 : ");
```

```
dan = scan.nextInt();
```

```
if(dan <= 1 || dan >= 10){
```

```
System.out.println("2 ~ 9 사이의 숫자를 입력 하세요");
```

```

}else{

for(int i = 1; i <= 9; i++){

System.out.println(dan + " * " + i + " = " + (dan * i));

}

}

```

다중 for문 : for문 안에 for문이 있는 경우

// 예) 결과를

// 1 2 3 4

// 1 2 3 4

// 1 2 3 4

```
for(int i = 1; i <= 3; i++){ //y축
```

```
for(int j = 1; j < 5; j++){ //x축
```

```
System.out.print(" " + j);
```

```
}//안
```

쪽 for문의 끝

```
System.out.println();//줄바꿈
```

```
}// 바깥쪽 for문의 끝
```

-----예제 1

//예) 결과를

// 01 02 03 04

// 05 06 07 08

// 09 10 11 12

```
int count = 0;
```

```
for(int i = 1; i <= 3; i++){ //y축
```

```
for(int j = 1; j < 5; j++){ //x축
```

```
System.out.printf("%02d ", ++count); // 정수3자리에 j값 출력!
```

```
} //안
```

```
    } // for문의 끝
```

```
System.out.println(); //줄바꿈
```

```
} // 바깥쪽 for문의 끝
```

-----예제 ex

```
//예) 결과를
```

```
// A B C D
```

```
// E F G H
```

```
// I J K L
```

```
char ch = 'A';
```

```
for(int i = 1; i <= 3; i++){
```

```

for(int j = 1; j <= 4 ; j++){

    System.out.print(" " + ch++);

    //System.out.printf("%2c", ch++);
}

```

```

System.out.println();//줄바꿈
}

```

----- 예제 2

```

//예) 결과를
//*
//* *
//* * *
//* * * *
//* * * * *

```

```

for(int i = 1; i <= 5; i++){

```

```

    for(int j = 0; j < i; j++){

```

```

        System.out.print("* ");

    }

```

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

----- 예제 3

```
//결과를
//      *
//      * * *
//      * * * * *
//      * * * * * * *
//      * * * * * * * *
for(int i = 0; i < 5; i++){
```

```
    for(int j = 0; j < 5 + i; j++){
```

```
        j > 3)
```

```
        if(i +
```

```
            System.out.print("* ");
```

```
            else
```

```
            System.out.print(" ");
```

```
        }
```

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

----- 예제 4

다중for문을 이용하여 아래와 같은 결과를 도출하는 로직을 구현하자.

결과)

```
1 2 3 4 5 6 7 8 9 10
2 3 4 5 6 7 8 9 10 1
3 4 5 6 7 8 9 10 1 2
4 5 6 7 8 9 10 1 2 3
5 6 7 8 9 10 1 2 3 4
6 7 8 9 10 1 2 3 4 5
7 8 9 10 1 2 3 4 5 6
8 9 10 1 2 3 4 5 6 7
9 10 1 2 3 4 5 6 7 8
10 1 2 3 4 5 6 7 8 9
```

```
폴이))  
public class Work {  
    public static void main(String[] args) {
```

```
        for(int i = 1; i <= 10; i++){
```

```
            for(int j = 0; j < 10; j++){
```

```
                num = i + j;
```

int

```
                if(num > 10)
```

```
                    num -= 10;
```

```
                System.out.print(num + " ");
```

```
            }//안쪽 for문
```



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

```
}//바깥쪽 for문
```

```
}//main
```

```
}
```

```
-----for끝
```

while문 : 간편한 구성을 가진 반복문(선 비교 후 처리)

```
int num = 1;
```

```
while(num <= 10){
```

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

```
num++;
```

```
}//while문의 끝
```

예) 결과를

```
//1 = 1 2 3 4 5
```

```
//2 = 6 7 8 9 10
```

```
//3 = 11 12 13 14 15
```

```
//4 = 16 17 18 19 20
```

```
//5 = 21 22 23 24 25
```

```
//while안에 for문을 사용한 예제
```

```
int num = 1;
```

```
int num2 = 1;
```

```
while(num <= 5){
```

```
System.out.print(num++ + " = ");
```

```
for(int i = 1; i <= 5; i++){
```

```
System.out.print(num2++ + " ");
```

```
}
```

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

```
//while문의 끝
```

-----예제 1

do - while문

```
// do~while문 : 선처리, 후비교
```

```
//제어문 중 유일하게 세미콜론(;)을 가진다.
```

```
int i = 11;
```

```
do{
```

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

```
}while(i <= 10); //결과 : 11
```

문제

```
/*
```

```
* Java, Jsp, Android의 시험을 봤다.
```

```
*
```

```
* 각 점수는 랜덤하게 0 ~ 130 점이 대입
```

```
*
```

```
* 각 점수를 검사하는 로직은 do ~ while로 작성하는데 각 과목점수가
```

```
* 0 ~ 100이 아닐 경우에는 반드시 0 ~ 100 사이가 나오도록 재실행을 시킬 것.
```

```
*
```

```
* 이렇게 세과목의 점수가 모두 0 ~ 100 사이가 되면 do-while문을 빠져나와
```

```
* 점수가 화면에 출력되도록 하는 코드를 작성해보자.
```

```
*/
```

```
int java = 0, jsp = 0, android = 0;
```

```
do {
```

```
java= new Random().nextInt(131); //0 ~ 130의 난수 발생
```

```
jsp =
```

```
new Random().nextInt(131);
```

```
android = new Random().nextInt(131);
```

```
} while (java > 100 || jsp > 100 || android > 100);
```

```
System.out.println("자바 : " + java);
```

```
System.out.println("JSP : " + jsp);
```

```
System.out.println("안드로이드 : " + android);
```

Break문 : 반복문 내에서 강제적으로 가장 가까운 반복문을 빠져나갈 때 사용

```
for(int i = 1; i <= 3; i++){
```

```
for(int j = 1; j <= 10; j++){
```

```
if (j
```

```
% 3 == 0)//j 를 3으로 나눈 나머지가 0일때
```

```
break;// 가장 가까운 반복문 탈출!
```

```
System.out.print(" " + j);
```

```
}//안
```

```
쪽 for문의 끝
```

```
System.out.println();//줄 바꿈!
```

```
}//바깥쪽 for문의 끝
```

-----예제 1

```
int n = 1;
```

```
while (true) { //무한반복. !true나 false는 불허
```

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

```
n++;
```

```
// n
```

```
이 10보다 클 때 break로 while문을 빠져나오는 구문을 만들어보자
```

```
if (n
```

```
> 10)
```

```
break;
```

```
}
```

-----예제2

continue문 : 반복문 내에서 특정 문장이나 여러 문장들을 건너뛰고자 할 때.

```
for(int i = 1; i <= 3; i++){
```

```
for(int j = 1; j <= 10; j++){
```

```
    % 3 == 0)//3의 배수일때 가장 가까운 반복문의 처음으로 돌아간다.    if(j
```

```
    continue;
```

```
    System.out.print(" " + j);  
} //안쪽 for문의 끝
```

```
System.out.println();  
} //바깥쪽 for문의 끝
```

-----예제 1

```
int n = 0;
```

```
while(n < 10){  
    n++;
```

```
    % 3 == 0)    if(n
```

```
    continue;
```

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

```
}
```

-----예제 2

break label, continue label문 :

가장 가까운 반복문을 빠져 나오거나 건너뛰는 break나 continue와 달리 특정 반복문에 label을 붙여 한번에 두 개 이상의 반복문을 빠져나올 수 있게 하는 제어문

```
outer : for(int i = 1; i <= 3; i++){
```

```
for(int j = 1; j <= 10; j++){
```

```
% 3 == 0)
```

```
if ( j
```

```
break outer;//outer라는 레이블이 붙은 반복문을 탈출한다.
```

```
System.out.print(" " + j);
```

```
}
```

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

```
}
```

-----예제 1

```
int n = 0;
```

```
outer : while(true){
```

```
if( n
```

```
>= 10) //이 부분은 무한반복 되는거 확인후 가장 마지막에 추가.
```

```
break;
```

```
while(true){
```

```
    n++;
```

```
    if ( n
```

```
    % 3 == 0){
```

```
        System.out.println("continue를 만남");
```

```
        continue outer;
```

```
    }
```

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

```
}
```

```
}
```

-----예제2

서비스 - if문을 이용한 숫자마술

```
public class NumMagic {
```

```
public static void main(String[] args) {
```

```
String select; //yes 혹은 no를 선택하기 위한 변수
```

```
int
```

```
result = 0; //사용자가 생각한 숫자를 출력하기 위한 변수
```

```
System.out.println("1 ~ 30 사이의 숫자를 머리속으로 생각하세요");
```

```
System.out.println("04 05 06 07 12");
```

```
System.out.println("13 14 15 20 21");
```

```
System.out.println("22 23 28 29 30");
```

```
System.out.print("여기에 당신이 생각한 숫자가 있습니까? - y | n : ");
```

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

```
select = scan.next();
```

```
if(select.equals("y"))
```



```
result += 4;
```

```
System.out.println("-----");
```

```
System.out.println("16 17 18 19 20");
```

```
System.out.println("21 22 23 24 25");
```

```
System.out.println("26 27 28 29 30");
```

```
System.out.print("여기에 당신이 생각한 숫자가 있습니까? - y | n : ");
```

```
select = scan.next();
```

```
if(select.equals("y"))
```

```
result += 16;
```

```
System.out.println("-----");
```

```
System.out.println("01 03 05 07 09");
```

```
System.out.println("11 13 15 17 19");
```

```
System.out.println("21 23 25 27 29");
```

```
System.out.print("여기에 당신이 생각한 숫자가 있습니까? - y | n : ");
```

```
select = scan.next();
```

```
if(select.equals("y"))
```

```
result += 1;
```

```
System.out.println("-----");
```

```
System.out.println("08 09 10 11 12");
```

```
System.out.println("13 14 15 24 25");
```

```
System.out.println("26 27 28 29 30");
```

```
System.out.print("여기에 당신이 생각한 숫자가 있습니까? - y | n : ");
```

```
select = scan.next();
```

```
if(select.equals("y"))
```

```
result += 8;
```

```
System.out.println("-----");
```

```
System.out.println("02 03 06 07 30");
```

```
System.out.println("10 11 14 15 18");
```

```
System.out.println("19 22 23 26 27");
```

```
System.out.print("여기에 당신이 생각한 숫자가 있습니까? - y | n : ");
```

```
select = scan.next();
```

```
if(select.equals("y"))
```

```
result += 2;
```

```
System.out.println("-----");
```

```
System.out.println("당신이 생각한 숫자는 " + result + "입니다.");
```

```
}
```

```
}
```

신기하죠?? 어떤구조인지 생각해보세요~

자바 강의 1주차(2) 문제 출제