

22. 10 .19 - Flow Control (フロー制御)

「Operator」

기능별, 우선순위

1. 산술연산자
2. 비교 (비교연산자는 이항으로 사용됨)
3. 논리 (2개 이상의비교 연산자를 결합 시키기 위해)
4. 복합연산자
5. 비트
6. 대입

항의 개수 (Operand)

- 단항 (괄호도 연산자 => 연산의 우선순위, 자료형 변환기능을 제공)
- 이항
- 삼항 (> :)

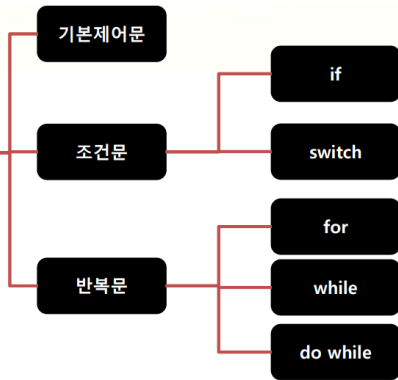
※ 이항연산 ※

1. 문자열 > 실수 > 정수 (더큰 범위를 담을 수있는 자료형이 우선)
2. Size of byte

재사용성, 유저 프렌들리로 인해 프로그래밍언어가 발전해 갔다.

**흐름(The sequence of
executed code) 제어문** →
(하나의 실행문장을 제어 하
는 것)

흐름제어 (컴퓨터언어)



구조적언어 (의 기본) / 선택, 반복의 구현법은 언어마다 다르지만
하기의 내용은 **JAVA**

• 기본제어문 (上から下へ)

- 위에서 아래로

• 선택문(선택) - and or ?

- `if`
- `switch`

• 반복문(반복) - 조건이 참이면 계속 반복

- `for`
- `while`
- `do while`

※ **if 시리즈**중 선택 할 수 있는 최대의 개수는 **1개** 이다.
(**else** 가 있으면 무조건 **else** 에 마지막 조건이 걸리기 때문에 역시 **1개**의 선택사항이 나온다)

else

(이미 언급된 것에 덧붙여) 또[그 밖의] 다른

```

※선택※
// if
//   1개중 0개 또는 1개
//   if (조건식) 실행될 문장 ;

// if - else
//   2개중 1개
//   if (3>2)
//     System.out.println("참");
//   else
//     System.out.println("거짓");

else가 들어오면 선택하는 경우의수는 0이 될 수 없다!!!! 무조건 else 에 걸린다.

// if..... - else if
//   n개 중 1개 또는 0개
//   if ( 2 > 3 )
//     System.out.println(1);
//   else if ( 2 > 4 )
//     System.out.println(2);
//   else if ( 2 > 1 )          // ※ 조건이 부합되면 해당 조건문을 빠져 나간다
//   {
//     System.out.println(3);
//     System.out.println(3.5);
  
```

```

    }

    else if ( 2 > 4)
        System.out.println(4);
    else if ( 2 > 4)
        System.out.println(5);

    System.out.println(6);

    ※ 2개 이상 선택 시
    조건식을 변경 , if문을 따개서 만들면 된다

    // if - else if ..... else
    //     n개 중에 1개
    if (2>3)
        System.out.println(1);
    elif (3>4)
        System.out.println(2);
    elif (3>4)
        System.out.println(3);
    else
        System.out.println(4);

    System.out.println(5);

    // switch
    //     n개 중 0 ~ n개

```

흐름제어문 : 기본 , 선택 , 반복

선택 과 반복 은 판별하는 기준으로 조건식 으로 나타낸다

선택 또는 반복시 실행되는 문장은

- **기본** : 조건식 다음에 **한 개의 문장**이 **실행** 된다. (조건식은 괄호 안에다가 넣어줘야한다)
- 조건식을 판단후 리턴하는 결과 값의 자료형은 반드시 **불린형(true , false)** 이다
- **그룹화** : **{ }** (블레이스) 를 이용하여 선택 or 반복문 실행 시 실행되는 코드들을 **그룹화** 할 수 있다.

```

public class test_1 {

    public static void main(String[] args) {

        if (3 < 2)
            System.out.println("1"); // 기본 : 한개의 문장만 실행 함으로 밑에의 내용과 별개이다

        System.out.println("2");
        System.out.println("3");

        [ console ]
        2
        3

        if (3 < 2)

            System.out.println("1");

```

```

        System.out.println("2");
        System.out.println("3");

    if (3 < 2)
    {
        System.out.println("1"); // 블레이스: 1개 이상의 문장들을(여러개의 코드들을) 하나로 묶기 위해서 사용한다.
        System.out.println("2");
    }
    System.out.println("3");

```

※ { } 블레이스 는 흐름제어문의 선택 과 반복 시 그 조건식이 참 이면 선택될 때 한 개 이상의 문장들을 그룹화 하기 위해 사용한다 0 컴퓨터는 구분자로 코드를 구분한다.

범위 구분

한줄	여러줄
;	{ }

```

Scanner scn = new Scanner(System.in);

// 콘솔로부터 문자열 입력
String inputstring = scn.next();

// 입력 받은 문자열에서 첫번째 글자를 반환
char inputChar = inputstring.charAt(0);

```

입력 알파벳이 f t u k 일 경우 해당 값 출력

```

Scanner scn = new Scanner(System.in);

// 콘솔로부터 문자열 입력
String inputstring = scn.next();

System.out.println(inputChar);

if (inputChar == 'f') {
    System.out.println('f');
}

else if (inputChar == 't') {
    System.out.println('t');
}

else if (inputChar == 'u') {
    System.out.println('u');
}

else if (inputChar == 'k') {
    System.out.println('k');
}else {

```

```

        System.out.println("잘못된 입력 값");
    }

```

조건식이 많으면 배열 사용 한다

```

import java.util.Scanner;

public class test_1 {

    public static void main(String[] args) {

        Scanner scn = new Scanner(System.in);

        // 콘솔로부터 문자열 입력
        String inputstring = scn.next();

        // 입력 받은 문자열에서 모든 글자를 반환

        for ( int i = 0 ; i < inputstring.length() ; i++ )

        {
            char inputChar = inputstring.charAt(i);

            System.out.println(inputChar);

        }

    }

}

```

```

import java.util.Scanner;

public class test_1 {

    public static void main(String[] args) {

        Scanner scn = new Scanner(System.in);

        // 콘솔에서 문자 입력
        String inputstring = scn.next();

        int c_h = 0; // h의 수
        int c_t = 0; // t의 수
        int c_p = 0; // p의 수
        String igai = "" ; // 그ほか의 문자를 넣는 변수

        for ( int i = 0 ; i < inputstring.length() ; i++ )

        {
            char inputChar = inputstring.charAt(i);

            // 'h' , 't' , 'p' -> 문자ごとに発生する頻度を出力すること
            // httppppabc ->

```

```

// その他 3個

// 1. 条件の h t p をわきまえる
// 2. そのあと 当たる文字がある場合は変数に1ずつ増加させる

// h 場合
if (inputChar == 'h')
{
    ++c_h ; // 当たる際に1ずつ増加
}
// t 場合
else if (inputChar == 't')
{
    ++c_t ; // 当たる際に1ずつ増加
}
// p 場合
else if (inputChar == 'p')
{
    ++c_p ; // 当たる際に1ずつ増加
}

// その他の場合
else
{
    igai += inputChar ;// 当たる際に文字を増加
}

}

//出力の内容
System.out.println("h의 개수 : " + c_h);
System.out.println("t의 개수 : " + c_t);
System.out.println("p의 개수 : " + c_p);
System.out.println("그외 : " + igai);

}
}

```

조건식 먼저 만들어 놓고 안에 내용들 하나씩 작성하기

배열까지 하고 열라 짜라

메소드는 oop를 알아야해서 혼자서 나중에 보라

미리 문제 주면 한주전에 예습을 해서 온다 / 반드시 문제를 설명 할 수 있도록 → 성적 드간다.

Quiz 3. 구구단 출력 프로그램

<변경 후 코드>

```
import java.util.Scanner;

public class quize_test2_syuuri {

    public static void main(String[] args) {

        // [問題] 키보드로부터 정수 2개를 입력 받아, M~N 사이의 정수를 i

        // 키보드로 부터 정수 입력 받기 new Scanner 사용
        Scanner scn = new Scanner(System.in);

        // 변수 //
        System.err.println("M값을 입력하세요");
        int M = scn.nextInt(); // 첫 번째 입력 값
        System.err.println("N값을 입력하세요");
        int N = scn.nextInt(); // 두 번째 입력 값

        // * 안내 출력문 *
        if (M > N)
            System.out.println("N : " + N + " ~ " + "M : " + M
            else
                System.out.println("M : " + M + " ~ " + "N : " + N

        // 출력 포맷은 아래와 반드시 동일, [정수 자리 구분"," 출력]

        // 1. M < N or M == N
        if (M < N || M == N) {
            // M ~ N 까지 사이를 순차적으로 출력할 반복문 작성
            for (int index = M; index < N + 1; ++index) {

                // 1. M < N 일 경우 M에서 N까지 정수 출력
                if (M < N) {
                    System.out.print(index);

                    // 끝자리면 , 출력 안함
                    if (index < N) {
                        System.out.print(", ");
                    }
                }

                // 3. M == N 일 경우 입력 값 출력
                else {
                    System.out.println("N, M 정수 값 : " + M);
                }
            }
        }

        // 2. M > N 일 경우 N에서 M까지 정수 출력
        else if (M > N) {
            // for 반복문
            for (int inversion_index = N; inversion_index < M + 1; ++inversion_index) {
                System.out.print(inversion_index);

                // 끝자리면 , 출력 안함
                if (inversion_index < M) {
                    System.out.print(", ");
                }
            }
        }

        // // 3. M == N 일 경우 입력 값 출력
```

Quiz 3

1. 아래 형식의 “구구단 출력” 프로그램을 작성하라 (40점)

1.1 구구단 출력 (15점)

1.2 짝수단 출력 구현 시, “% 연산자를 이용하여 구현 할 것!” (5점)

1.3 출력 포맷 준수, 하드코딩 점수 없음, 반복문 안에 로직을 만들어 작성 (20점)

*** 출력 결과**

2 X 1 = 2	2 X 2 = 4	2 X 3 = 6
2 X 4 = 8	2 X 5 = 10	2 X 6 = 12
2 X 7 = 14	2 X 8 = 16	2 X 9 = 18
4 X 1 = 4	4 X 2 = 8	4 X 3 = 12
4 X 4 = 16	4 X 5 = 20	4 X 6 = 24
4 X 7 = 28	4 X 8 = 32	4 X 9 = 36
6 X 1 = 6	6 X 2 = 12	6 X 3 = 18
6 X 4 = 24	6 X 5 = 30	6 X 6 = 36
6 X 7 = 42	6 X 8 = 48	6 X 9 = 54
8 X 1 = 8	8 X 2 = 16	8 X 3 = 24
8 X 4 = 32	8 X 5 = 40	8 X 6 = 48
8 X 7 = 56	8 X 8 = 64	8 X 9 = 72

2. 키보드로부터 정수 2개를 입력 받아, M~N 사이의 정수를 출력하라 (40점)

2.1 M < N 일 경우 M에서 N까지 정수 출력 (15점)

2.2 M > N 일 경우 N에서 M까지 정수 출력 (15점)

2.3 M == N 일 경우 입력 값 출력 (5점)

2.4 출력 포맷은 아래와 반드시 동일, [정수 자리 구분 “,” 출력] (5점)

*** 출력 결과**

M < N 일 경우	M > N 일 경우	M == N 일 경우
M값을 입력하세요 1 N값을 입력하세요 5 M : 1 ~ N : 5사이 정수 값은 1, 2, 3, 4, 5 ----감사합니다----	M값을 입력하세요 3 N값을 입력하세요 -3 N : -3 ~ M : 3사이 정수 값은 3, 2, 1, 0, -1, -2, -3 ----감사합니다----	M값을 입력하세요 5 N값을 입력하세요 5 N, M 정수 값 : 5 ----감사합니다----

```
//      else if (M == N)
//      {
//          System.out.println("N, M 정수 값 : " + M);
//      }

      System.out.print("\n" + "----- 감사합니다 -----");
      // -----

  }
}
```

<변경 전 코드 >

```
import java.util.Scanner;

public class quize_test_2 {

    public static void main(String[] args) {

        // [問題] 키보드로부터 정수 2개를 입력 받아, M~N 사이의 정수를 출력하라

        // 키보드로 부터 정수 입력 받기      new Scanner 사용
        Scanner scn = new Scanner(System.in);

        // 변수 //
        System.err.println("M값을 입력하세요");
        int M = scn.nextInt() ; // 첫 번째 입력 값
        System.err.println("N값을 입력하세요");
        int N = scn.nextInt() ; // 두 번째 입력 값

        // ※ 안내 출력문 ※
        if (M < N || M > N )
            System.out.println("M : " + M + " ~ " + "N : " + N + " 사이 정수 값은");

        // M ~ N 까지 사이를 순차적으로 출력할 반복문 작성
        for (int index = M ; index < N+1 ; ++index )
        {

            // 출력 포맷은 아래와 반드시 동일, [정수 자리 구분," 출력]

            // 1.  M < N 일 경우 M에서 N까지 정수 출력
            if (M < N)
            {
                System.out.print(index);

                // 끝자리면 , 출력 안함
                if (index < N)
                {
                    System.out.print(", ");
                }
            }

            // 2.  M > N 일 경우 N에서 M까지 정수 출력
            else if (M > N)
            {

                // for 반복문
                for (int inversion_index = N ; inversion_index < M+1 ; ++inversion_index )
                {
                    System.out.print(inversion_index);

                    // 끝자리면 , 출력 안함
                    if (index < M)
```



```

        {
            System.out.print(", ");
        }
    }

}

// 3. M == N 일 경우 입력 값 출력
else if (M == N )
{
    System.out.println("N, M 정수 값 : " + M);
}

}

System.out.print("\n"+"----- 감사합니다 -----");
//-----
}
}

```

```

package exam;

import java.util.Scanner;

/**
 * Object
 * First input value == 'a' and Second input value == 'c'
 */

public class test08 {
    public test08() {
        Scanner sc = new Scanner(System.in);
        boolean flag = false;

        while (true) {
            // USER input just one character
            char inputValue = (sc.next()).charAt(0);

            if (inputValue == 'a') {    // when input value is 'a'
                flag = true;          // first input flag is true
            } else if (inputValue == 'c' && flag) {    // when input value is 'c' and first input flag is true
                System.out.println("참");    // print "참"
                break;
            } else {
                flag = false;
            }
        }
    }
}

```

```

package exam;

import java.util.Scanner;

```

```

public class test09 {
    public test09() {
        int N = 100;
        int target = 100;

        Scanner sc = new Scanner(System.in);

        // 자연수 N 값 입력
        while (true) {
            System.out.println("1~N까지 7의 배수합 구하기.");
            N = sc.nextInt();

            if (N >= 1) break;
        }

        // 1 ~ N
        int total = 0;
        int fnd = 0;
        boolean isfnd = false;

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

            if (i % 7 == 0) {
                total += i;
                if (total > target && !isfnd) {
                    fnd = i;
                    isfnd = true;
                }
            }
        }

        // Print Result
        System.out.println("1~" + N + "까지 7의 배수의 합은: " + total);
        if (isfnd) {
            System.out.println("누적된 합의 값이 처음으로 " + target + "을 넘기는 7의 배수는 " + fnd);
        }
    }
}

```

자바스크립트는 메모리상에서 어떻게 동작하는지 잘 알아야한다.

메모리구조 공부 꼭하기

- > 인강

남한테 알려주기 , 알려주면서 모르는게 뭔지 알게 된단다

CS 에 대해서 공부 좀하기

ソフトバンク

プロジェクトマネージャーとしての能力 (PM)

転職の相当に範囲が縮まってしまう。

メルカリ (プログラムの能力重視、半端ないハードルって)

<https://about.mercari.com/>

必須条件

mercari

- メルカリ・ソウゾウのミッションとバリューへの共感
- 以下のうち1つ以上の経験(3年以上)
 - Backend: Go, PHP, Java, TypeScript などを用いたAPIの開発経験
 - Frontend: TypeScript, React, Next.js などを用いた開発経験
 - Mobile: Swift, Kotlinを用いた開発経験
- パフォーマンスとスケーラビリティを考慮した設計開発能力
- チームでプロダクト開発を行うコミュニケーション能力
- 新しい技術も積極的に学んでいける方

```
import java.util.Scanner;

public class test_1 {

    public static void main(String[] args) {

        Scanner scn = new Scanner(System.in);

        // 콘솔로부터 문자열 입력
        String inputstring = scn.next();

        // 입력 받은 문자열에서 모든 글자를 반환

        int c_h = 0; // hの数
        int c_t = 0; // tの数
        int c_p = 0; // pの数
        String igai = "" ; // そのほかの文字を入れる変数

        for ( int i = 0 ; i < inputstring.length() ; i++ )

        {
            char inputChar = inputstring.charAt(i);

            // 'h' , 't' , 'p' -> 文字ごとに発生する頻度を出力すること
            // hhttppppabc ->
            // その他 3個

            // 1. 条件の h t p をわきまえる
            // 2. そのあと 当たる文字がある場合は変数に1ずつ増加させる

            // h 場合
            if (inputChar == 'h')
            {
                ++c_h ; // 当たる際に1ずつ増加
            }
            // t 場合
            else if (inputChar == 't')
            {
                ++c_t ; // 当たる際に1ずつ増加
            }
            // p 場合
            else if (inputChar == 'p')
            {
                ++c_p ; // 当たる際に1ずつ増加
            }
        }
    }
}
```

```

    }

    // その他の場合
    else
    {
        igai += inputChar ;// 当たる際に文字を増加
    }

}

//出力の内容
System.out.println("h의 개수 : " + c_h);
System.out.println("t의 개수 : " + c_t);
System.out.println("p의 개수 : " + c_p);
System.out.println("그외 : " + igai);

}

}

```

```

import java.util.Scanner;

public class quize_test2_syuuri {

    public static void main(String[] args) {

        // [問題] 키보드로부터 정수 2개를 입력 받아, M~N 사이의 정수를 출력하라

        // 키보드로 부터 정수 입력 받기 new Scanner 사용
        Scanner scn = new Scanner(System.in);

        // 변수 //
        System.err.println("M값을 입력하세요");
        int M = scn.nextInt(); // 첫 번째 입력 값
        System.err.println("N값을 입력하세요");
        int N = scn.nextInt(); // 두 번째 입력 값

        // ※ 안내 출력문 ※
        if (M > N)
            System.out.println("N : " + N + " ~ " + "M : " + M + " 사이 정수 값은");
        else
            System.out.println("M : " + M + " ~ " + "N : " + N + " 사이 정수 값은");

        // 출력 포맷은 아래와 반드시 동일, [정수 자리 구분"," 출력]

        // 1. M < N or M == N
        if (M < N || M == N) {
            // M ~ N 까지 사이를 순차적으로 출력할 반복문 작성
            for (int index = M; index < N + 1; ++index) {

                // 1. M < N 일 경우 M에서 N까지 정수 출력
                if (M < N) {
                    System.out.print(index);

                    // 끝자리면 , 출력 안함
                    if (index < N) {
                        System.out.print(", ");
                    }
                }

                // 3. M == N 일 경우 입력 값 출력
                else {
                    System.out.println("N, M 정수 값 : " + M);
                }
            }
        }
    }
}

```

```

    }

    // 2. M > N 일 경우 N에서 M까지 정수 출력
    else if (M > N) {
        // for 반복문
        for (int inversion_index = N; inversion_index < M + 1; ++inversion_index) {
            System.out.print(inversion_index);

            // 끝자리면 , 출력 안함
            if (inversion_index < M) {
                System.out.print(", ");
            }
        }
    }

    // 3. M == N 일 경우 입력 값 출력
    else if (M == N)
    {
        System.out.println("N, M 정수 값 : " + M);
    }

    System.out.print("\n" + "----- 감사합니다 -----");
    // -----

}

}

```