

# Day15; 20220720

🕒 작성일시	@2022년 8월 3일 오전 10:24
▼ 강의 번호	
📅 유형	
🔗 자료	
☑ 복습	<input type="checkbox"/>
☰ 속성	

- 두 데이터의 대소를 판단하는 알고리즘
  - 2개의 데이터를 비교하려면 선택 구조를 사용한다.
  - 조건식에서는 관계 연산자를 사용한다.

데이터를 비교하는 처리를 고려하자 a와 b를 비교하여 a가 크면 a를 출력하고 그렇지 않으면 b를 출력한다. 특별한 계산이 필요하지 않은 단순한 알고리즘이므로 이것만으로 대소 판정을 마칠 수 있다.

- pseudo code

a, b 입력

a > b

a 출력

\_\_\_\_\_

b 출력

```
if(a > b){  
    System.out.println(a);  
}  
else{  
    System.out.println(b);  
}
```

}

---

a와 b를 비교한다.

a가 큰가?

a를 출력한다.

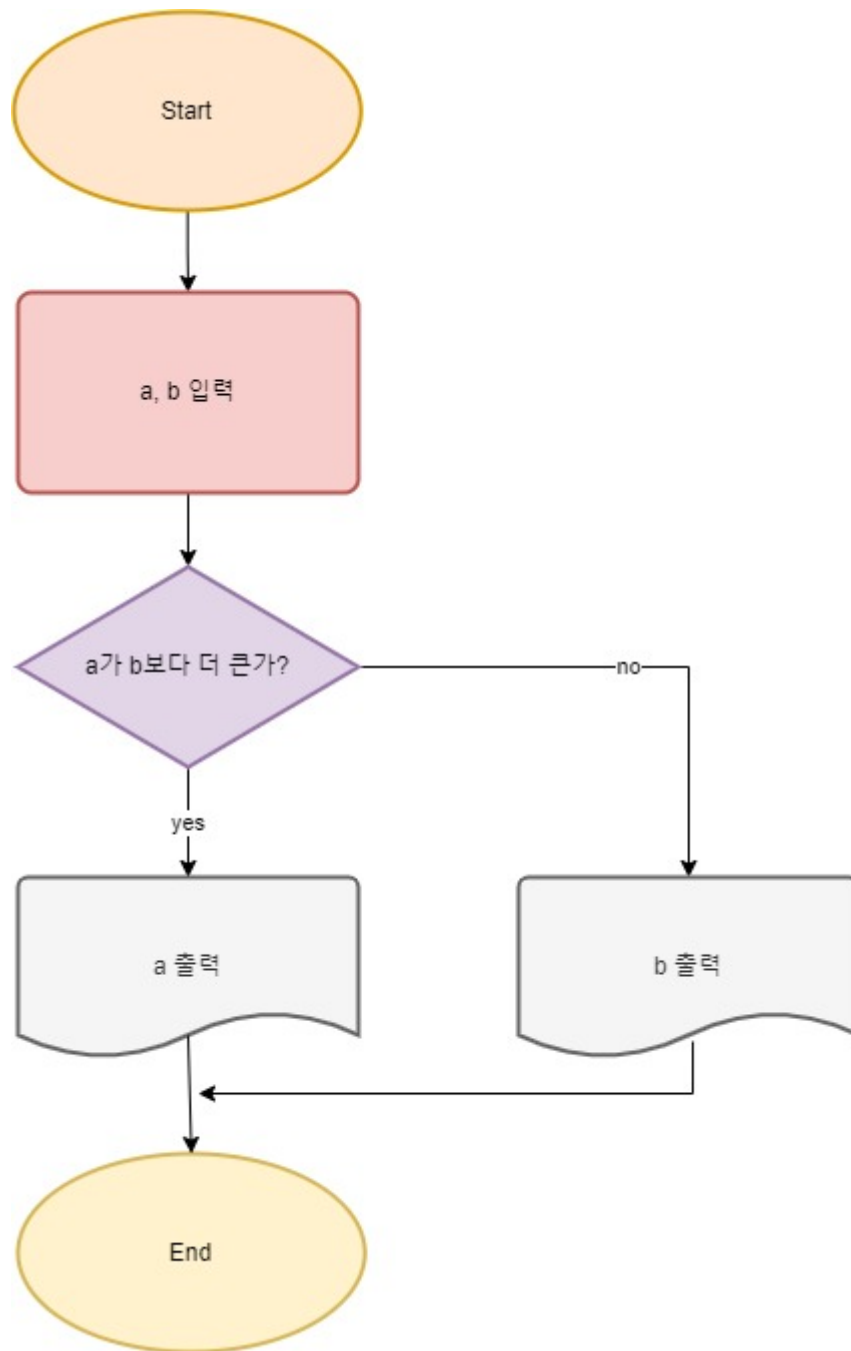
-----

b를 출력한다.

---

+ :: • pseudo code

```
if(a > b){  
    System.out.println(a);  
}else{  
    System.out.println(b);  
}
```



---

### 비교 연산자 정리

“>” 크다 초과

“<” 작다

“≥” 크거나 같다 이상

“≤” 작거나 같다 이하

“≠” 같지 않다.

“==” 같다

---

a와 b를 입력

a가 큰가?

a를 출력한다.

-----

b가 큰가?

b를 출력한다.

-----

a, b가 같습니다.

---

- 두 변수의 데이터를 교환하는 알고리즘
  - 두 변수의 데이터를 직접 교환할 수는 없다. (파이썬은 가능)
  - 교환용 임시 변수를 추가하여 사용하게 된다.

우선 정수형 변수 2개를 준비하고 각각 a, b로 정한다. 변수 a의 값과 b의 값을 직접 교환할 수는 없으므로 임시변수 t를 사용하여 먼저 a의 값을 t로 대입하고 b의 값을 a에 대입하고 마지막으로 다시 t의 값 (즉 원래 a였던 값)을 b 대입하여 교환을 마치게 된다. 교환을 마친 최종 a와 b값을 출력한다.

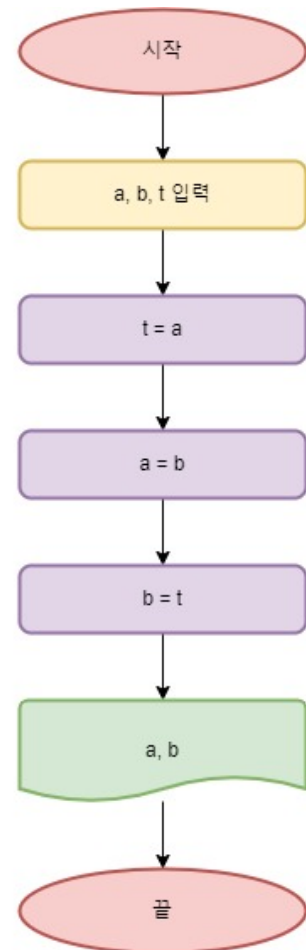
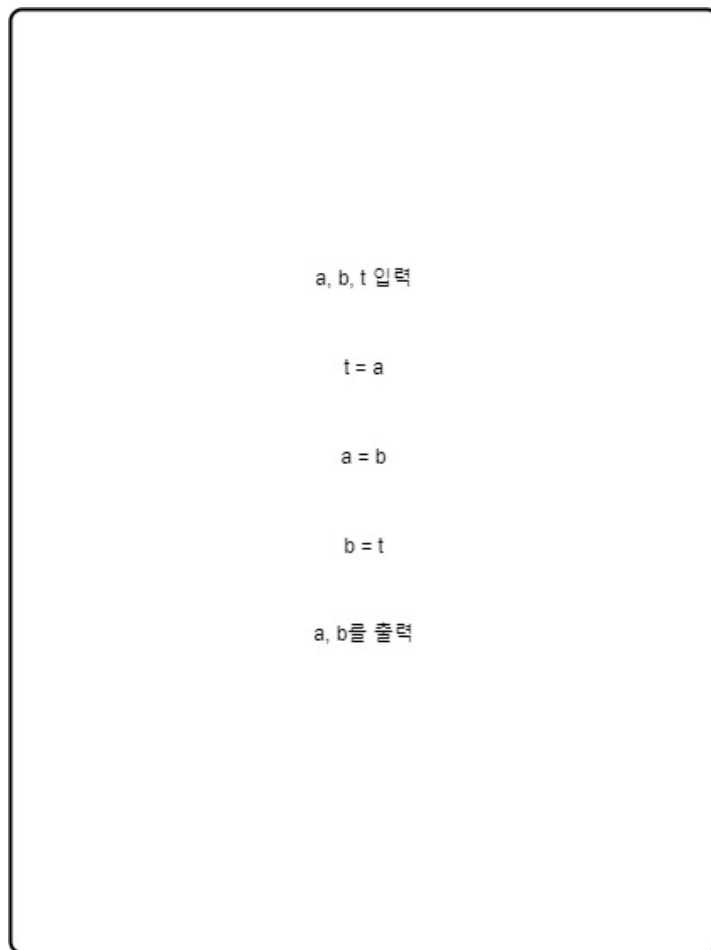
a, b, t 입력

t = a

a = b

b = t

a, b를 출력



```
package day15_1;
```

```
public class Change {
```

```

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int a, b, t;
        a = 3;
        b = 2;
        t = 0;

        t = a;
        a = b;
        b = t;

        System.out.println(a);
        System.out.println(b);
    }

```

```
}
```

- 합계값을 계산하는 알고리즘

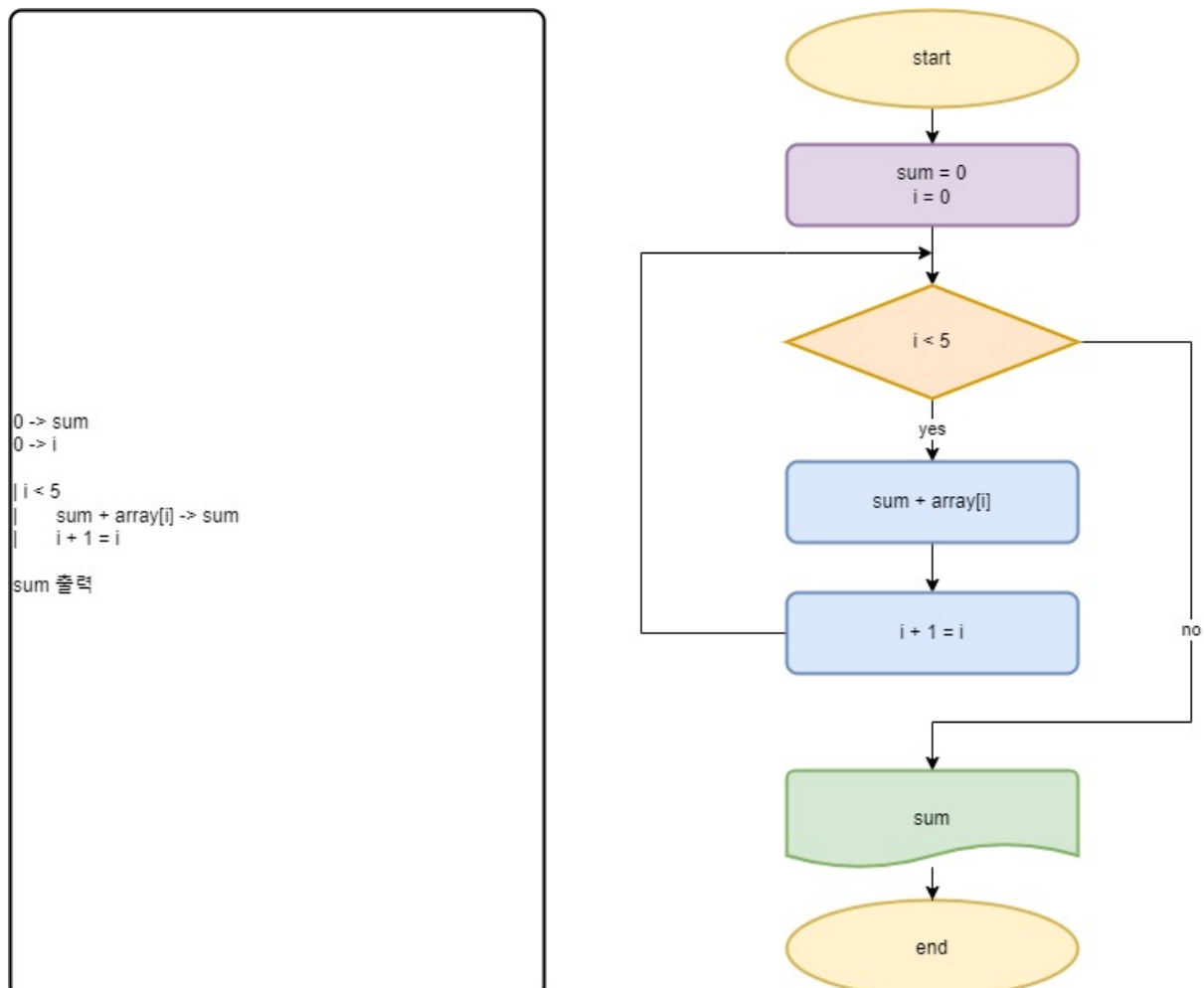
- 요소들의 합계를 계산하려면 반복 구조를 사용해야 한다.
- 합계값을 대입할 변수 sum을 준비한다.
- 변화하는 첨자(몇 번째, index) 변수를 주로 i를 사용한다.
- 반드시 반복 구조를 사용할 때는 반드시 무한 루프에 빠지지 않게 주의하자.

12 13 11 14 10    sum?

0   1   2   3   4    [ 첨자 ]

12 13 11 14 10    [ 배열 ]

데이터의 합계를 구하는 가장 단순한 알고리즘은 전부를 더하는 처리를 실행하는 것이다.



```
package day15_1;

public class Repeat {
```

```
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int sum = 0;
        int[] array = {12, 13, 11, 14, 10};

        for(int i = 0; i< array.length; i++) {
            sum += array[i];
        }
        System.out.println(sum);
    }
}
```

```
}
```

- 최대값을 찾는 알고리즘
  - 임시 최대값 변수 x를 사용한다.
  - 변화하는 첨자는 변수 i를 사용한다.
  - 반복처리는 무한루프에 빠지지 않도록 처리한다.

가장 큰값을 찾는 일을 어떻게 분해할까?

앞에서부터 뒤로 순서대로 2개의 데이터를 비교해 나가는 방법을 사용하자. 두 데이터의 대소를 판정하는 앞에서 사용한 방법을 이용한다.

0	1	2	3	4	[첨자]
12	13	11	14	10	[배열]