

2021-12-23 수업

<second HW 복습>

- class 작성 시 -> 대문자로 시작
- 변수명 or 매서드 -> 소문자로 시작. 다중 단어의 경우 시작만 소문자고 나머지는 이니셜마다 대문자 ex) int max, int maxNum
(**변수명을 a로하는 것은 지양하자. ex) int a = 0;
의도를 파악하기 쉬운 변수명으로 지정하기)

- range값을 지정할 때 상수 값을 넣기보단 변수를 따로 지정해주는 것이 좋음
ex) 64~90 사이의 랜덤값 출력시
int rand = (int)(Math.random() * 27 + 64); ---> 이렇게 상수값으로 지정하는 것 보다

```
int min = 64;  
int max = 90;  
int range = (max - min) + 1;  
int rand = (int)(Math.random() * range + min); ---> 이렇게 표현하는 것이 좋다
```

- Math.random()은 0.0~1.0미만의 double값 리턴

- final int --> 상수값을 지정
(**상수값을 지정할 때는 변수명은 전부 대문자로 ex. final int **MAX**)
상수값을 넣을 때 습관적으로 final을 써주도록 하자

- ! 는 NOT으로 참을 거짓으로, 거짓을 참으로 만들

<FOR, WHILE문 복습>

1. WHILE문

1) 구조

초기화;

while(조건문){

조건문을 충족했을 시 작업할 내용

}

2) while문에서 증감식의 위치는 두 가지로 가능(나중에는 1)을 많이 쓰게 될 것)

(1) while(**i++ < 10**){

 }

(2) while(i<10){

i++;

```
}
```

3) while을 이용한 무한루프

```
while(true){  
}
```

2. FOR문

1) 구조

```
for(초기화 ; 조건문 ; 증감식){  
}
```

**초기화 부분은 for문 진입 시 최초 한 번만 실행됨

**조건문은 매 반복마다 검사

**증감은 { } 내부의 내용이 끝나면 진행

2) 시나리오

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

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

```
}
```

의 경우

(1) i = 1이 됨 (초기화를 1로 했으니까)

(2) i <=10이 맞는지 조건을 확인 후 조건이 참이니까 { } 내부 진행

(3) i값인 1을 출력

(4) { } 내용 끝났으니까 i=2로 증감

(5) 조건문과 또 비교하고 참이니까 { } 내부 진행

(6) i값인 2를 출력

(7) { } 내부 끝났으니까 i=3으로 증감

(8) (4)~(7) 반복

3) for을 이용한 무한루프

```
for ( ; ; ) {  
}
```

--> 초기화, 조건문, 증감식 모두 없애기

4) sum이 for 외부에 필요한 이유

```
int sum = 0;
```

```
for(int i=1 ; i<101 ; i++){  
    sum += i;  
}
```

(1) sum은 지역변수로 main 내부 어디서든 살아있음

만약 sum을 for문 내부에 적는다면 for문이 끝나는 순간 for문과 함께 소멸한다 (정보손실)

(2) sum=0은 실제로 메모리 상에 중구난방으로 데이터가 들어있음

int sum이라고 해놓으면 실제 sum에는 무작위의 값들이 들어있기 때문에

int sum = 0; 이라고 0을 설정해서 덧셈(+)을 통한 누산을 명확하게 해줄 필요가 있음

3. 공통 적용

- +=, -=, *=, /=, %=

num += x ---> num = num + x 와 동의어

- 전위연산자 ++i ---> 해당줄이 실행되기 전에 덧셈 완료

- 후위연산자 i++ ----> 해당줄이 실행된 이후 덧셈 완료 (해당줄의 아랫줄에서 증감 발생 하는 듯)