

## 1. ECLIPSE (JAVA)

### 1) for문 : 반복문

- for문의 기본 구조 -

```
for(초기값 ; 조건식 ; 증식 ) {  
    // 조건이 참일 때, 실행되는 실행문  
}
```

- 단일 for문 예제 -

```
for(int i=0; i<4; i++) {  
    System.out.print("!");  
}
```

< 예제 로직 >

1. int i = 0
2. i < 4 체크
3. System.out.println("i")
4. i++
5. i = 1
6. i < 4 체크
7. System.out.println("i")
8. i++
9. i = 2
10. i < 4 체크
11. System.out.println("i")
12. i++
13. i = 3
14. i < 4 체크
15. System.out.println("i")
16. i++
17. i = 4
18. i < 4 체크
19. i는 4보다 작지 않으므로, for문 탈출

출력결과

!!!!

- 단일 for문을 이용한 list 값 출력 예제-

```
ArrayList<Integer> arr = new ArrayList<Integer>();  
arr.add(9);  
arr.add(7);  
arr.add(5);  
  
for(int val : arr) {  
    System.out.println(val);  
}
```

< 예제 로직 >

1. int val = arr.get(0)
2. System.out.println(9)
3. val = arr.get(1)
4. System.out.println(7)
5. val = arr.get(2)
6. System.out.println(5)

출력결과

9  
7  
5

## 2) 파일입출력

### 3) 메모리구조 (static, stack, heap)

#### (1) static 영역

- 필드 부분에서 선언된 변수(전역변수)와 정적멤버변수(static이 붙은 자료형) Static 영역에 데이터 저장
- Static영역의 프로그램의 시작부터 종료가 될 때까지 메모리에 남아있게된다.

#### (2) stack 영역

- 메소드 내에서 정의하는 기본 자료형(int, double, byte, long, boolean 등) 에 해당되는 지역변수의 데이터의 값이 저장되는 공간
- 메소드가 호출 될 때, 메모리가 할당되고 종료되면 메모리 해제

#### (3) heap 영역

- 참조형(Reference Type)의 데이터 타입을 갖는 객체(인스턴스), 배열등 저장
- 이때 변수(객체, 객체변수, 참조변수)는 **Stack 영역의 공간에서 실제 데이터가 저장된 Heap 영역의 참조값** (reference value, 해시코드 / 메모리에 저장된 주소를 연결해주는 값)을 **new 연산자를 통해 리턴 받는다.** 다시 말하면 **실제 데이터를 갖고 있는 Heap 영역의 참조 값을 Stack 영역의 객체가 갖고 있다.** 이렇게 리턴 받은 참조 값을 갖고 있는 객체를 통해서만 해당 인스턴스를 핸들 할 수 있다.

SQL Developer는 소스 참조