성능을 향상시켜 주는 Java 코딩 방법

# 반복문 안의 조건에 함수를 사용하지 않는다.

for(int i = 0; i < list.size(); i++){

System.out.println(list.get(i));

}

for문이 실행될 때마다 list.size() 함수를 호출하여 성능이 떨어진다.

int size = list.size();

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

System.out.println(list.get(i));

}

와 같이 변수를 선언하여 사용하면 함수 호출을 한번만 할 수 있다.

# 문자결합 기능의 함수에는 append를 사용하자.

문자결합 3가지 방법

1. concat 사용

String a = "";

a = a.concat("abcde");

2. += 연산자 사용

String a = "";

a += "abcde";

3. append 사용

StringBuffer a = StringBuffer();

a = a.append("abcde");

성능의 빠르기는 append > concat > += 순이다.

이유는 concat과 연산자는 문자열을 붙일 때마다 메모리가 할당되기 때문에 기존에 사용한 메모리가 남아있고, 메모리 사용이 종료되어도 가비지 컬렉션이 바로 사용이 종료된 메모리를 삭제해주지 않기 때문이다.