**package** thread\_;

**public** **class** SingleTon {

**private** **int** num = 3;

**private** **static** SingleTon *instance* = **null**; // 초기화 딱 한번

**public** **static** SingleTon getInstance() { // 자료형 : SingleTon

**if** (*instance* == **null**) { // instance가 null인 경우는 딱 한번

*instance* = **new** SingleTon(); // 처음 딱 한번만 수행하고 그 다음부터 안한다

}

**return** *instance*;

}

**public** **static** **void** main(String[] args) {

SingleTon aa = **new** SingleTon();

aa.num++;

System.***out***.println("aa = " + aa); // 클@16진수. 주소값 다르게 잡힘

System.***out***.println("num = " + aa.num); // 4

System.***out***.println();

SingleTon bb = **new** SingleTon();

bb.num++;

System.***out***.println("bb = " + bb); // 클@16진수

System.***out***.println("num = " + bb.num);

System.***out***.println();

System.***out***.println("--- 싱글톤 ---"); // 메모리에 딱 한번만 잡겠다. 주소값 같게 잡힘

SingleTon cc = SingleTon.*getInstance*();

cc.num++;

System.***out***.println("cc = " + cc); // 클@16진수

System.***out***.println("num = " + cc.num); // 4

System.***out***.println();

SingleTon dd = SingleTon.*getInstance*();

dd.num++;

System.***out***.println("dd = " + dd); // 클@16진수

System.***out***.println("num = " + dd.num); // 5

System.***out***.println();

}

}

// SingleTon cc = new SingleTon();.....(x)

// SingleTon cc = SingleTon.getInstance() - 메소드 이용. static이니까 클래스명으로 다이렉트하게 들어감

// new를 시켜주면 heap메모리를 차지하고, 무조건 메모리에 new한만큼 만든다