//new class 객체를 생성하면 이름을 주는 것이 아닌 주소가 담겨 있는 것

**package** inheritance2;

**class** Test **extends** Object {

@Override

**public** String toString() {

// Overriding 해주게 되면 우선권이 자식클래스에 떨어지게 된다

// Object, public String toString() (부모)

// Test, public String toString() (자식)

**return** getClass() + "@개바부"; // 주소값을 @개바부로 바꿔주겠다

// 클래스 이름을 갖고 오고 싶으면 getClass() 입력. getClass는 Object꺼

}

}

//-----------

**class** Sample **extends** Object {

}

//-----------

**public** **class** ObjectMain {

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

Test t = **new** Test();

System.***out***.println("객체 t = " + t); // 객체 t = class inheritance2.Test@개바부

System.***out***.println("객체 t = " + t.toString()); // 클@16진수,,,toString은 갖고 있는 주소를 @레퍼런스값 나오게 해주는 놈

// toString은 object가 갖고 있는데 Sample은 Object한테 상속을 받았기 때문에 자동으로 toString을 불러내고 있어

// 위와 아래의 결과가 같다

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

Sample s = **new** Sample();

System.***out***.println("객체 s = " + s.toString()); // 클@16진수, 객체 s = inheritance2.Sample@1175e2db

System.***out***.println("객체 s = " + s.hashCode()); // 10진수, 객체 s = 292938459

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

String str = "apple";

System.***out***.println("객체 str = " + str.toString()); // 문자열, 객체 str = apple

// 일반적인 객체를 찍으면 toString을 불러와 주소값을 가져오지만

// String 클래스만 유별나게 Overriding하면 부모 것을 무시하고 자식 것을 처리하므로 문자열이 나온다

System.***out***.println("객체 str = " + str.hashCode()); // 10진수-문자열을 다 표현해주지 못한다

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

String aa = **new** String("apple");

String bb = **new** String("apple");

System.***out***.println("aa==bb : " + (aa == bb)); // 주소 - F

System.***out***.println("aa.equals(bb) : " + (aa.equals(bb))); // 문자열 비교 - T

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

Object cc = **new** Object();

Object dd = **new** Object();

System.***out***.println("cc==dd : " + (cc == dd)); // 주소 - F

System.***out***.println("cc.equals(dd) : " + (cc.equals(dd))); // 주소 - F

// equals는 문자열 물어보는 것이 아니라 Object일 때는 주소값을 비교하는 것

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

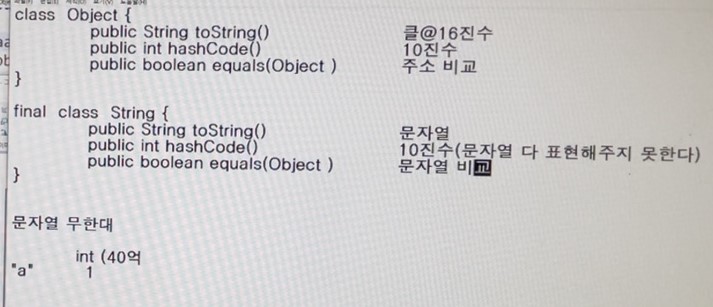
Object ee = **new** String("apple");

Object ff = **new** String("apple");

System.***out***.println("ee==ff : " + (ee == ff));// 주소 - F

System.***out***.println("ee.equals(ff) : " + (ee.equals(ff))); // 문자열 - T

}



}

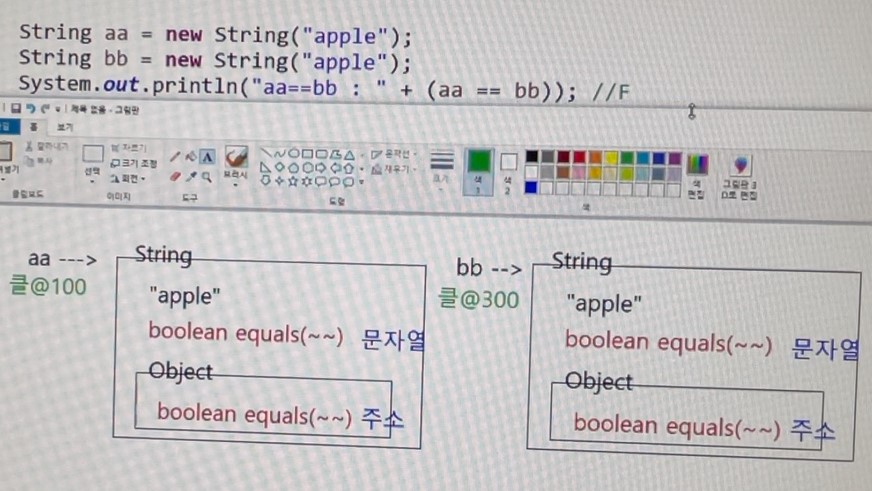
[결과]

객체 t = class inheritance2.Test@개바부

객체 t = class inheritance2.Test@개바부

객체 s = inheritance2.Sample@1175e2db

객체 s = 292938459



객체 str = apple

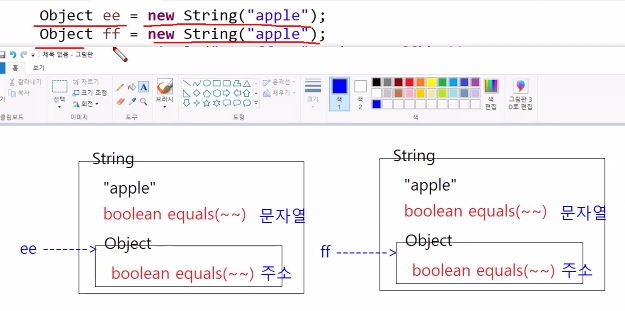
객체 str = 93029210

aa==bb : false

aa.equals(bb) : true

cc==dd : false

cc.equals(dd) : false



ee==ff : false

ee.equals(ff) : true