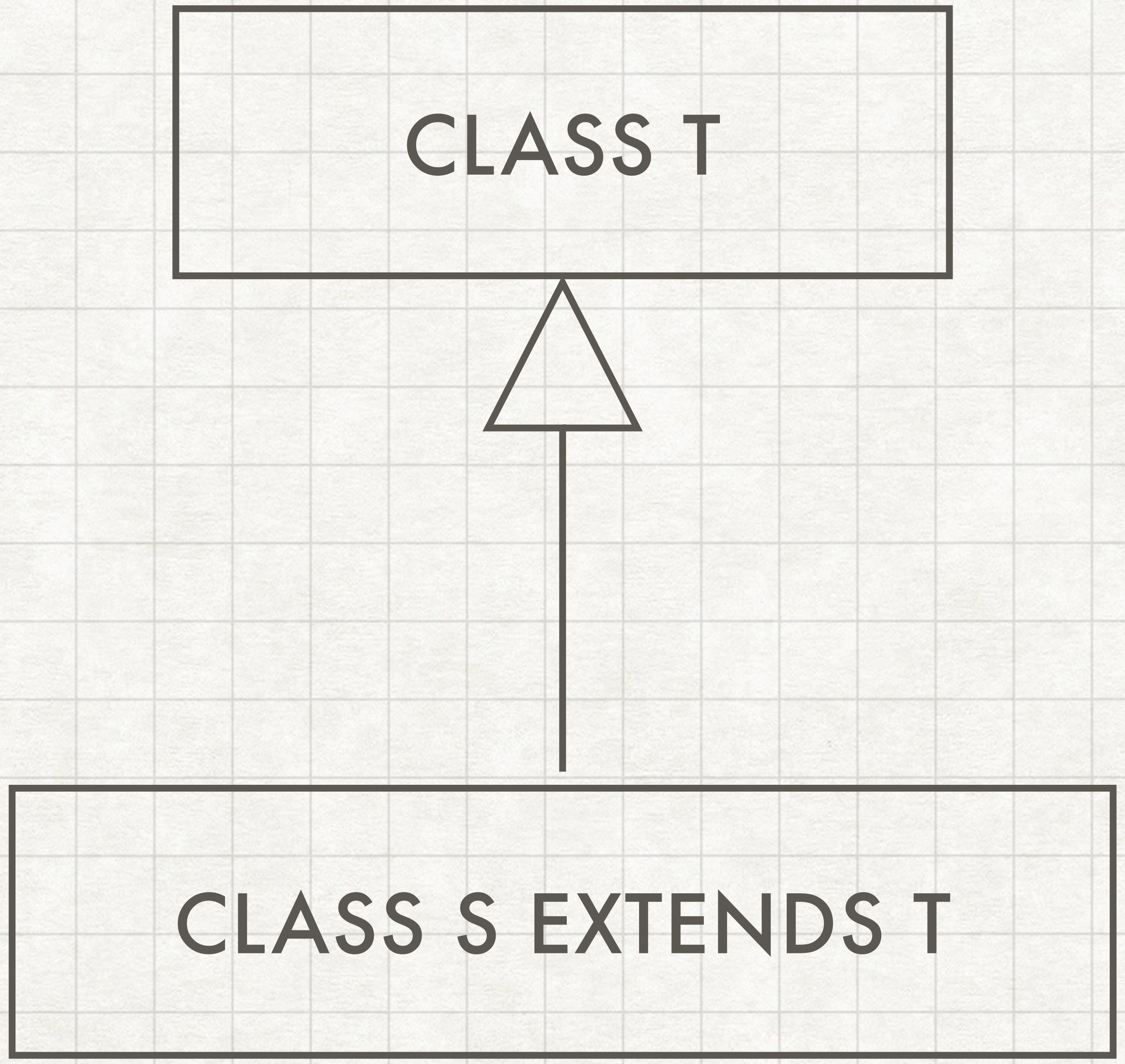
LSP

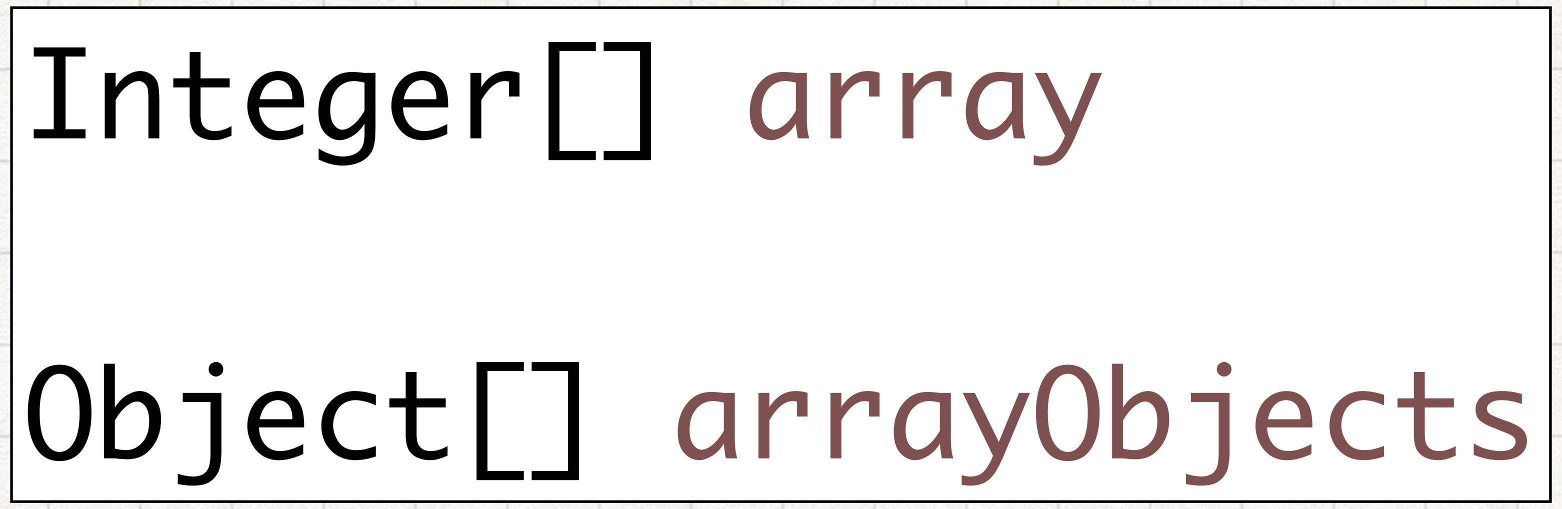
LISKOV SUBSTITUTION PRINCIPLE



```
T obj = new T();
```

```
T obj = new S();
```

> *Subtype Requirement*: Let $\varphi(x)$ be a property provable about objects x of type T. Then $\varphi(y)$ should be true for objects y of type S where S is a subtype of T.
>
> — *Barbara Liskov*



```
Integer[] array

Object[] arrayObjects
```

Is there any subtype relationship between them?

```
public static void sort(Object[] a)
```

```
Integer[] array = { 1,2,3 };

Object[] arrayObjects = array;
```

Integer[] is a subtype of Object[]

Is this alright?

```
Integer[] array = { 1,2,3 };

Object[] arrayObjects = array;

arrayObjects[0] = “Hello";
```

```
Exception in thread "main" java.lang.ArrayStoreException:
java.lang.String
 at proves.Main.main(Main.java:12)
```

initially there was java.util.Date

... and then java.sql.Timestamp arrived

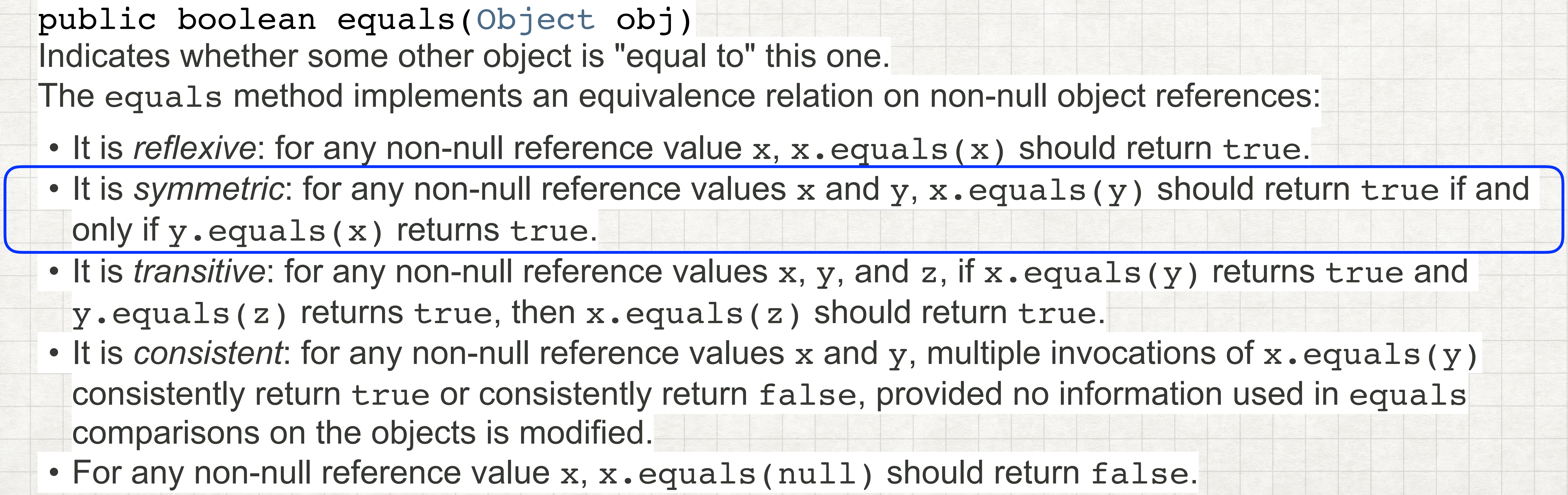
sort of Date + nanoseconds

```java
public class Timestamp extends java.util.Date
```

Any problems?

```java
Date date = new Date();
Timestamp ts = new Timestamp(date.getTime());

System.out.println(date.equals(ts));
System.out.println(ts.equals(date));
```

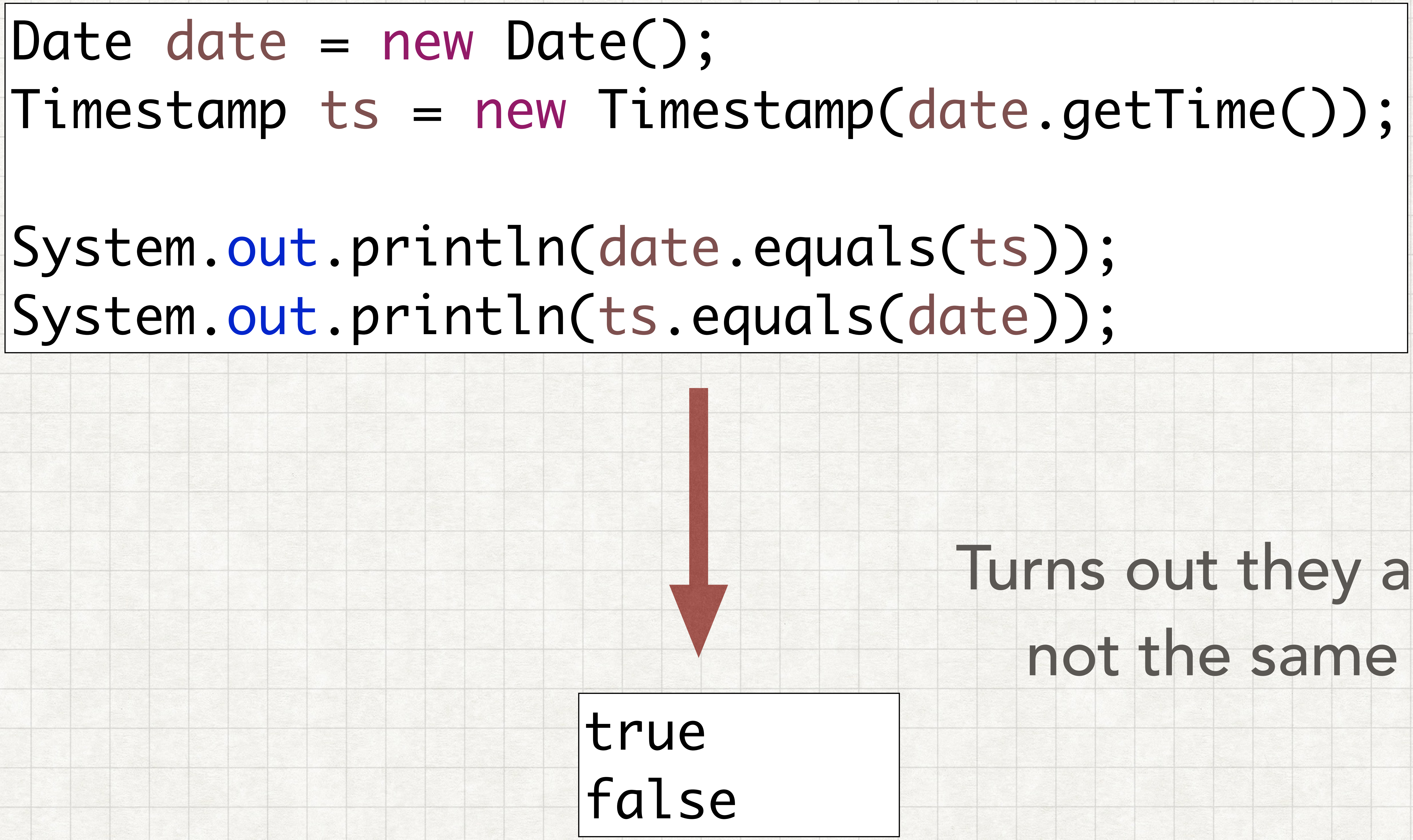


`public boolean equals(Object obj)`

Indicates whether some other object is "equal to" this one.

The `equals` method implements an equivalence relation on non-null object references:

- It is *reflexive*: for any non-null reference value `x`, `x.equals(x)` should return `true`.
- It is *symmetric*: for any non-null reference values `x` and `y`, `x.equals(y)` should return `true` if and only if `y.equals(x)` returns `true`.
- It is *transitive*: for any non-null reference values `x`, `y`, and `z`, if `x.equals(y)` returns `true` and `y.equals(z)` returns `true`, then `x.equals(z)` should return `true`.
- It is *consistent*: for any non-null reference values `x` and `y`, multiple invocations of `x.equals(y)` consistently return `true` or consistently return `false`, provided no information used in `equals` comparisons on the objects is modified.
- For any non-null reference value `x`, `x.equals(null)` should return `false`.



```
Date date = new Date();
Timestamp ts = new Timestamp(date.getTime());

System.out.println(date.equals(ts));
System.out.println(ts.equals(date));
```

```
true
false
```

Turns out they are
not the same