

discussion re: string equality in Java

the issue...

How are strings represented? How do tests for equality work?

investigation

Check out the following tester & its results.

tester

```
public static void main(String[] args) {
    String x, y;
    x = "abc";
    y = "abc";
    checkEq( x, y, "compile time strings");

    x = new String("abc");
    y = new String("abc");
    checkEq( x, y, "constructed strings");

    x = (new String("abc")).intern();
    y = (new String("abc")).intern();
    checkEq( x, y, "interned strings  ");

    char[] c = {'a', 'b', 'c'};
    char[] d = {'a', 'b', 'c'};
    checkEq( c, d, "independent char[]s");

    char[] e = c;
    checkEq( c, e, "assigned char[]s #1");

    c[1] = 'X';
    checkEq( c, e, "assigned char[]s #2");
}

private static void checkEq( Object a, Object b, String txt )
{
    System.out.print( txt );
    if (a == b)
        System.out.println(" - ==");
    else
    {
        System.out.print(" - !=");
        if (a.equals(b))
            System.out.println(" but equal");
        else
            System.out.println(" and not equal");
    }
}
```

results

```
compile time strings - ==
constructed strings - != but equal
interned strings     - ==
independent char[]s - != and not equal
assigned char[]s #1 - ==
assigned char[]s #2 - ==
```

questions

- what does the compiler do with literal strings?
- what does the String constructor do?
- is the == operator overloaded to equals with Strings? –if not what does it do?
- some say that it is rarely necessary to use the String constructor.
Why may we agree with this?
Why may we disagree?
- are Strings the same as char[] arrays?
- what does **intern** do?
could intern be useful for other object types? why do you think intern is not an Object method?