

SoftReference operation synchronization detail

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2006-05/msg03335.html>

- *From:* "Stanimir Stamenkov" <stanio@xxxxxx>
 - *Date:* 31 May 2006 08:17:30 -0700
-

The following may have more apparent effect when using a `WeakReference`. Suppose I initialize a class member field which is a `SoftReference`, at some point:

```
SoftReference cache = new SoftReference(obj);
```

Then I have the following code accessing the referenced object, in a method body:

```
Object obj = null;
if (cache != null) {
    obj = cache.get();
}
if (obj == null) {
    obj = new Object(); /* re-initialize obj */
    cache = new SoftReference(obj);
}
/* do with obj */
```

Is it guaranteed the 'cache' won't be cleared after the 'if (cache != null)' evaluates to true and prior executing the first statement in the 'if' block?

Is it guaranteed 'Reference.get()' will return a non-null result if the corresponding `SoftReference` or `WeakReference` is not cleared yet?, so I could go without the explicit 'obj = null' initialization in the above example:

```
Object obj;
if (cache != null) {
    obj = cache.get();
} else {
    obj = new Object(); /* re-initialize obj */
    cache = new SoftReference(obj);
}
/* do with obj */
```

—
Stanimir

