**Weak references**

A *weak reference* is a reference that isn't strong enough to force an object to remain in memory. Weak references allow you to leverage the garbage collector's ability to determine reachability for you, so you don't have to do it yourself.

You create a weak reference like this:

WeakReference weakWidget = new WeakReference(widget);

and then elsewhere in the code you can use weakWidget.get() to get the actual Widget object. Of course the weak reference isn't strong enough to prevent garbage collection, so you may find (if there are no strong references to the widget) that weakWidget.get() suddenly starts returning null.

**Soft references**

A *soft reference* is exactly like a weak reference, except that it is less eager to throw away the object to which it refers. An object which is only weakly reachable will be discarded at the next garbage collection cycle, but an object which is softly reachable will generally stick around for a while.

SoftReferences aren't *required* to behave any differently than WeakReferences, but in practice softly reachable objects are generally retained as long as memory is in plentiful supply. This makes them an excellent foundation for a cache, such as the image cache described above, since you can let the garbage collector worry about both how reachable the objects are (a strongly reachable object will *never* be removed from the cache) and how badly it needs the memory they are consuming.

And [Peter Kessler](http://weblogs.java.net/blog/peterkessler/) added in a comment:

The Sun JRE does treat SoftReferences differently from WeakReferences. We attempt to hold on to object referenced by a SoftReference if there isn't pressure on the available memory. One detail: the policy for the "-client" and "-server" JRE's are different: the -client JRE tries to keep your footprint small by preferring to clear SoftReferences rather than expand the heap, whereas the -server JRE tries to keep your performance high by preferring to expand the heap (if possible) rather than clear SoftReferences. One size does not fit all.