Java Object Comparisons

Mason Vail, Boise State University Computer Science

What Does “Equal” Mean?

Same Object?

or

Equivalent Objects?

Comparing references vs. Comparing objects

obj1 == obj2

Compares *references* to see if they are aliases: the same object in memory.

obj1.equals(obj2)

Compares two *objects* for equivalence, if `equals()` has been overridden.
If not overridden, `equals()` has the same result as `==`.

Comparable: Defining Natural Order

Natural Order - the most common or obvious ordering.

Integers:

1 3 3 4 6 9 11 12 12 12 15

`Comparable` interface defines natural order via `int` **`compareTo()`**

```
int result = obj1.compareTo(obj2)
```

-value: obj1 comes before obj2

0: obj1 and obj2 are equivalent - .equals() should be true

+value: obj2 comes before obj1

Comparable Contact Class

```
//constructor, accessors, mutators not shown
public class Contact implements Comparable<Contact> {
  private String lastName, firstName;
  private long phoneNumber;

  public int compareTo(Contact other) {
    int result = this.lastName.compareTo(other.lastName);
    if (result == 0) {
      result = this.firstName.compareTo(other.firstName);
    }
      return result;
  }

  public boolean equals(Contact other) {
    return (this.lastName.equals(other.lastName)
            && this.firstName.equals(other.firstName));
  }
}
```

Comparator: Alternate Ordering

For ordering other than by natural order (or for classes that did not implement `Comparable`), the **`Comparator`** interface defines a similar **`compare()`** method.

```
Comparator reverseComparator = new ReverseComparator();
  int result = reverseComparator.compare(obj1, obj2);
```

-value: obj1 comes before obj2

0: obj1 and obj2 are equivalent - .equals() should be true

+value: obj2 comes before obj1

Comparator for Contact Class

```
/** orders Contacts in reverse order by last name, first name */
public class ReverseComparator implements Comparator<Contact> {

  public int compare(Contact c1, Contact c2) {
    int result = -(c1.getLastName().compareTo(c2.getLastName()));
    if (result == 0) {
      result = -(c1.getFirstName().compareTo(c2.getFirstName()));
    }
      return result;
  }

}
```

Java Object Comparisons

Mason Vail, Boise State University Computer Science