

Java Collections

See the documentation for the Java Collections Framework.

This gives implementations for some standard data structures. We will do our own versions of several of these as we study them.

In Lab 2 we will re-implement the ArrayList class.

Example

I want to write a program that will read a sequence of names, sort it, and print it back in alphabetical order. Typical names might be

bob

Fred Flintstone

John Frederick Oberlin

Charles Philip Arthur George Windsor

First, what class structures will we use for this?

I like this:

First, we'll have a Name class to hold the names. This has two String class variables: first and last (which might more properly be called givenNames, familyName. We will make a constructor that takes a string holding the full name and splits it up into those fields.

Then we need to store a whole sequence of these. This seems like a good use of ArrayLists, so we'll make an `ArrayList<Name>` object to hold the list.

There are two ways to handle the sorting. One way is to have the Name class implement the Comparable<Name> interface, which just means that we give it a compareTo() method. This takes a Name n as an argument and returns -1, 0, or 1 if this < n, this ==n, or this > n.

The other way is to make a function object that implements the Comparator interface. This class needs a method `Compare(n1, n2)` that again returns -1, 0 or 1 if $n1 < n2$, $n1 == n2$ or $n1 > n2$.

We'll do both. We will make the Comparator class a *nested class*, meaning that it is a private static class inside the class that contains the ArrayList that we will use it to sort.

Another Example

I want to write an inventory program. This will read in lines from the user that are formatted as

`<object> <count>`

For example,

`hammer 23`

This indicates that we have found 23 hammers. If we see another line

`hammer 5`

then we need to update our count of hammers to 28.

What we are doing here is associating Strings with ints. We could do this by keeping a list of strings and a corresponding list of ints but that would be a pain to maintain. Look through the Collection classes for a structure that might do this.

The Java Utilities package gives us Collections, which are mostly sequential things like lists, stacks and queues, and Maps, which are associative structures. Look at the Java Map interface