Lab05 - Search and Sort

Due: Fri May 2, 2014

Searching and sorting… searching and sorting...

- ❏ Chapter 16 Sort and Search

I have some files for you. Copy all the Java files from my k: drive Lab05 to yours.

Monday - Search

Focus topic: `Comparable` interface, binary search, the Arrays class
Let's modify the the book's `RecursiveBinarySearcher` class (page 986) to work for objects, not ints. The `Comparable` interface is the key!

Steps:

- ❏ Modify `RecursiveBinarySearcher` to work with an array of `Comparable` objects.

- ❏ Create your `Lab05` class and put `main()` there. I have a stub in `Lab05Helper`. We'll search for baby boy names. Source: www.babycenter.com/top-baby-names-2013.

Here's a sample run:

```
** Welcome to Lab 05 **
     Chap 16 - Fun with search and sort

Boys names are:
[Aiden, Alexander, Benjamin, Caden, Caleb, Connor, Daniel, Elijah, ...

Enter a name [or enter to exit]: Mason
Mason was found at element 18

Enter a name [or enter to exit]: Tyler
Tyler was not found.

Enter a name [or enter to exit]:
```

Details:

- String objects are already `Comparable`. You'll notice that it's case-sensitive.
- Peek at page 490 to declare and initialize your array of strings.
- `Arrays.sort( Comparable[] array)` can sort an array of objects.
- `Arrays.toString( Object[] array)` can print the objects in array.

Wed - Sort

Let’s modify Quicksort to work with Objects and with a `Comparator`.

- Modify `IntQuickSorter` class (page 974-977) to work for objects with a `Comparator`. Change the class name to `QuickSorter`. Two steps: 1) make it work with objects, rather than ints, and 2) add a Comparator parameter.

- Create a Lab05b class for your main(). I have help in `Lab05Helper`. In your `main()`:
  - Create a petting zoo, an array of Critters
  - Print the array in its original order
  - Sort the array by Critter weight (using a `Comparator`) and print
  - Sort the array by Critter name (using a difference `Comparator`) and print

Details:

- Comparators always remind me of Listeners… so powerful, so flexible. Like a ninja.