CSC230 FA2014

Project #4

Set Ops

The purpose of this program is to practice

- Interface design
- Class design
- Implementing an interface
- Testing your work

Write a program called SetOps.java. In this file you will write/develop an interface, a class that implements the interface, and a main method that tests your interface.

Recall that a set is a special bag that does not allow duplicates.

Your assignment:

1. Develop a generic SetInterface. The SetInterface should include, at a minimum, the following basic set operations:
    a. Union
    b. Intersection
    c. Difference
    d. Symmetric Difference
    e. Add an item to the set
    f. Determine if an item is in the set
    g. Return the number of items in the set
    h. Remove an item (specified) from the set
    i. Remove an item (unspecified) from the set
    j. Return an array that contains all of the elements in the set

2. Write a generic ArraySet class. This class should implement the SetInterface. The only fields in the ArraySet class should be an array and an int that represents the number of items in the set. You may also add a static field for the initial capacity of the array. You must provide a way for the array to grow.
   In addition to the methods in the interface, this class should have a constructor. You may add any private, utility methods that you wish. You may also override the toString method.

3. Write a main method that
    a. Declares three SetInterface reference variables, initialized to three ArraySet objects.
    b. Populates each with the contents of the files "test1.in", "test2.in", and "test3.in", respectively.
    c. Outputs the contents of each set
    d. Outputs the union of the three sets
    e. Outputs the intersection of the three sets
    f. Outputs the elements that are in exactly one of the three sets (be careful!)

Caveats

- You should not add any methods to your SetInterface that do not define behaviors common to sets

- The fields in the ArraySet class should be private

Here is a sample run:

```
run:
S1 = [George, John, Paul, Ringo]
S2 = [James, Thomas, John, George]
S3 = [John, Luke, James, Peter]
S1 union S2 union S3 = [James, Thomas, Luke, Peter, Ringo, Paul, John, George]
S1 intersect S2 intersect S3 = [John]
Elements that are in exactly 1 set = [Thomas, Luke, Peter, Ringo, Paul]
BUILD SUCCESSFUL (total time: 0 seconds)
```