Operations on Sets

Out: 3/26
Due: 4/18 by 11:50 PM

Learning Objectives

- Creating Array List
- Manipulating an Array List

Definition 1. A **set** is a finite or infinite collection of objects in which order has no significance, and multiplicity is generally also ignored. Members of a set are often referred to as elements and the notation $x \in A$ is used to denote that x is an element of a set A. A set is usually denotes as a list of elements. For example, $\{2, 3, 4, 5, 6\}$ is a set that contains five elements.

In this project we will define a Set as an object that has an array list. We will define several basic operations on sets.

1 Basic Operations on Sets

Definition 2. The **cardinality** of a set is the number of elements that the set contains. For example, the cardinality of $\{2, 3, 4, 5, 6\}$ is five.

Definition 3. The **intersection** of two sets A and B is the set of elements common to A to B. This is written $A \cap B$, and is pronounced "intersection" or "cap."

Definition 4. The **union** of two sets A and B is the set obtained by combining the members of each without allowing multiplicity. This is written $A \cup B$, and is pronounced "union" or "cup ."

Definition 5. The **symmetric difference** of sets A and B is the set of elements belonging to either set A or set B but not both. The symmetric difference of sets A and B is usually denoted sets $A \,\Delta\, B$.

Definition 6. The **difference** of sets A and B , denoted $A - B$, is the set of elements belonging to set A but not B.

Definition 7. Sets A and B are **equal**, denoted $A = B$, both sets have the same elements. The way in which the elements are ordered does not matter.

Definition 8. A set A is a **subset** of set B, denoted $A \subseteq B$, if every element of set A is also an element of set B.

Definition 9. A set A is a **proper subset** of set B, denoted $A \subset B$, if every element of set A is also an element of set B but both sets A and B are not equal.

Implementing The Set Class

Your first task in this project is to complete the implementation of a class, *Set*. The *Set* class, in the file *Set.java*, on the course Moodle page for details about the methods in the class.

Writing A Client Class

I have also provided a sample client class *UseSet.java* to test your implementation of the Set class. After you have completed the implementation of the *Set* class, test it by running the UseSet.java implementation that i have provided for you. When everything goes smoothly, rewrite the main method so that it does the following:

1. Prompts the user to enter a list of integers representing the elements of the set s_1. It then reads these numbers from the keyboard into an array list and then uses an appropriate constructor from the Set class to create S_1.

2. Repeats this process to create two more sets S_2 and S_3.

3. Prints a message indicating whether $S_1 \subseteq S_2$.

4. Prints a message indicating whether whether $S_2 \subset S_3$.

5. Computes $S_1 \cap (S_2 \cup S_3)$.

6. Computes $(S_1 - S_2) \cap (S_1 \Delta S_2)$.
7. Computes $S_4 = (S_1 - S_2) \cup (S_2 - S_1)$.
8. Computes $S_5 = (S_1 \Delta S_2) \cup (S_2 \Delta S_1)$
9. Finally, determines whether $S_4 = S_5$

A typical program interaction would be:

```
Enter the elements of the first set> 2 3 5 7 1
Enter the elements of the second set> 1 2 3 4 5 6 7 8 9
Enter the elements of third set> 3 6 9 12 15 18 21
```

$\{2,3,5,7,1\} \subseteq \{1,2,3,4,5,6,7,8,9\}$ –> true

$\{1,2,3,4,5,6,7,8,9\} \subset \{6,9,12,15,18,21\}$ –> false

$\{2,3,5,7,1\} \cap (\{1,2,3,4,5,6,7,8,9\} \cup \{6,9,12,15,18,21\}) = \{2,3,5,7,1\}$

$(\{2,3,5,7,1\} - \{1,2,3,4,5,6,7,8,9\}) \cap (\{2,3,5,7,1\}\Delta\{1,2,3,4,5,6,7,8,9\}) = \{\}$

$(\{2,3,5,7,1\} - \{1,2,3,4,5,6,7,8,9\}) \cup (\{1,2,3,4,5,6,7,8,9\} - \{2,3,5,7,1\}) = \{4,6,8,9\}$

$(\{2,3,5,7,1\}\Delta\{1,2,3,4,5,6,7,8,9\}) \cup (\{1,2,3,4,5,6,7,8,9\}\Delta\{2,3,5,7,1\}) = \{4,6,8,9\}$

$\{4,6,8,9\} = \{4,6,8,9\}$ -> true

Submitting Your Work

Compile your two Java source files (Set.java and UseSet.java) until they are free of any syntax and/or semantic errors. You may submit your work as many times as you wish once the due time has not elapsed. After the due time, any further submission will mean that you get no points for the lab. Be sure to follow the coding guidelines. Your program will be graded for both proper style and syntax.

1. All source code files you submit must have a header with the following:

```
/**
  * @file filename
  * @author your name
  * @date 9999-99-99
  * Description: describe what this file does
  * Course: CSC 1350 Section X
  * pawsID: xxxxxxxx
  * Project #: 5
  * Instructor: Duncan
  */
```

2. Document every method in every class except for the main method in UseSet.java.

3. Once you are convinced that your program works correctly, locate your source files (Set.java and UseSet.java).

4. Create a zip file myPawsID_5.zip and add the files to it, where myPawsID is your LSU paws ID - the part of your LSU email address before the @ sign.

5. Electronically submit your work for grading via Moodle using the digital drop box.