CSC 1351-3, Spring 2020, Lab 6 Name this project Set

Create a project named Set that illustrates the use of generic classes and generic methods.

Implementing a Generic Set Class

DEFINITION 1. **Generics** is a way of defining classes, interfaces and methods by using type parameters to represent types. A more technical term for generics is **parametric polymorphism**. A generic method, interface or class is fully specified at run time. In Java, a generic type can only be specialized using an object type. Every primitive type in Java has an associated object type (a wrapper class).

DEFINITION 2. 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 denoted as a list of elements. For example, $\{2, 3, 4, 5, 6\}$ is a set that contains five elements.

In today's lab, you will implement a generic class, the **Set<T>** class. The class will be generic but its methods will not be generic. Some methods will use the formal type parameter of the class. In this lab, we introduce a basic software engineering concept called *composition* - one object is composed of another. A set is composed of a list. We will use the Java API ArrayList<T>, also a generic class, to implement the Set<T> class.

Basic Set Operations

DEFINITION 3. The **cardinality** of a set is the number of elements that the set contains. For example, the cardinality of $A = \{2, 3, 4, 5, 6\}$, denoted $|A|$ is 5.

DEFINITION 4. The **intersection** of two sets A and B is the set of elements common to A and B. This is written $A \cap B$, and is pronounced "intersection" or "cap."

DEFINITION 5. 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 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 $A = B$, when 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 $A \subset B$ or $A \subsetneq B$, if every element of set A is also an element of set B but both sets A and B are not equal.

Some Useful Java ArrayList API Methods

1. ArrayList<ObjectType> listName = new ArrayList();:: creates an empty array list, *listName*, that can store objects of the specified type.
2. listName.add(item);:: appends the specified *item* to the (back of) the list.
3. listName.get(index);:: returns the item in the array list with the specified index. Array lists use zero-based indexing.
4. listName.size();:: returns the current length, number of items, in the specified array list.
5. listName.isEmpty();:: returns true if the length of *listName* is 0 and, false, otherwise.
6. listName.contains(item);:: returns true if the array list contains the specified *item* and, false, otherwise.

This project will have the following two classes:

public class Set<T>

Complete the implementation of the Set<T> class provided in the starter code by implementing the methods union, diff, equals, subset, and properSubset.

public class SetDemo

Complete the implementation of the SetDemo class provided in the starter code by implementing the generic methods setOperationsDemo and getMax. Implement the method setOperationsDemo so that it produces exactly the output shown below. In the output, u means union, n means intersection, and the minus sign means difference. The method getMax finds and returns the maximum element in the set.

Output

```
s1 = {4, 2, 5, 1, 3}
s2 = {7, 3, 2, 9, 5}
s3 = {5, 8, 3, 2, 4}
(s1 - s2) u (s2 - s1) = {4, 1, 7, 9}
(s1 u s2) - (s1 n s2) = {4, 1, 7, 9}
Is (s1 - s2) n (s2 - s1) empty? true
Are s1 n (s2 u s3) and (s1 n s2) u (s1 n s3) equal? true
Are (s1 n s2) u (s2 n s3) and (s1 u s3) equal? false
Is (s1 - s2) a subset of s2? false
Is (s1 - s2) u (s2 n s3) a subset of s1? true
Is (s1 - s2) u (s2 n s3) a proper subset of s1? false
Max element in s1 = 5
Max element in s2 = 9
Max element in s3 = 8

s1 = {D, B, A, E, C}
s2 = {G, I, C, B, E}
s3 = {F, C, H, B, D}
(s1 - s2) u (s2 - s1) = {D, A, G, I}
(s1 u s2) - (s1 n s2) = {D, A, G, I}
Is (s1 - s2) n (s2 - s1) empty? true
Are s1 n (s2 u s3) and (s1 n s2) u (s1 n s3) equal? true
Are (s1 n s2) u (s2 n s3) and (s1 u s3) equal? false
Is (s1 - s2) a subset of s2? false
Is (s1 - s2) u (s2 n s3) a subset of s1? true
Is (s1 - s2) u (s2 n s3) a proper subset of s1? true
Max element in s1 = E
Max element in s2 = I
Max element in s3 = H

s1 = {Gina, Bob, Jim, Anna, Mary}
s2 = {Tom, Hal, Mary, Bob, Anna}
s3 = {Hal, Mary, Bob, Sara, Gina}
(s1 - s2) u (s2 - s1) = {Gina, Jim, Tom, Hal}
(s1 u s2) - (s1 n s2) = {Gina, Jim, Tom, Hal}
Is (s1 - s2) n (s2 - s1) empty? true
Are s1 n (s2 u s3) and (s1 n s2) u (s1 n s3) equal? true
Are (s1 n s2) u (s2 n s3) and (s1 u s3) equal? false
Is (s1 - s2) a subset of s2? false
Is (s1 - s2) u (s2 n s3) a subset of s1? false
Is (s1 - s2) u (s2 n s3) a proper subset of s1? false
Max element in s1 = Mary
Max element in s2 = Tom
Max element in s3 = Sara
```