

2XB3 - A1 - Implementing ADTs

Bassel Rezkalla - 400137450 - rezkallb

Set API

Methods	Description	Parameters	Return Type
getStrArr	A simple getter, returns this Set's state variable, "StringArray".	N/A	ArrayList<String>
add	A function for adding an element to this Set. Returns a new Set, consisting of this Set's elements as well as the added String s. No duplicates in the Set can be created by this function.	String s	Set
remove	A function for removing an element from a Set. Returns a new Set identical to this Set, but excluding the removed String s.	String s	Set
Union	A function for returning the union Set of 2 Sets. Returns a new Set which is the union of this Set and Set S.	Set S	Set
Intersection	A function for returning the intersection Set of 2 Sets. Returns a new Set which is the intersection of this Set and Set S.	Set S	Set
Difference	A function for returning the Set difference of 2 Sets. Returns a new Set which is the Set difference of this Set and Set S.	Set S	Set
Product	A function for returning the Cartesian product (cross product) of this set by Set S. Returns a new set of all possible pairs of concatenated elements of the form rs where r is in this Set and s is in Set S.	Set S	Set

2XB3 - A1 - Implementing ADTs

Bassel Rezkalla - 400137450 - rezkallb

isEqual	Determines whether or not this Set is equal to Set S. Returns a boolean value; true if both sets are equal, false otherwise.	Set S	Boolean
isSubset	Determines whether or not this Set is a subset of Set S. Returns a boolean value; true if this set is a subset of Set S, false otherwise.	Set S	Boolean
getCount	A simple getter for retrieving the number of elements in this Set. Returns the size of this Set as an integer.	N/A	Integer
toString	A function for representing this Set as a string in the following format: "{elem1,elem2,...,elemn}".	N/A	String

Design decisions / assumptions

I made a few key design decisions when implementing my API for Sets.

I decided to use ArrayLists for my state variable to store the Set's elements.

I decided to make my class private and final, to conserve encapsulation and immutability respectively. I did the same for my ArrayList state variable.

Since Sets (by definition) do not contain duplicate elements, I made sure to prevent duplicates in both the constructor, and an add method that I created (for adding String elements to a Set). This way, my Sets have no way of acquiring duplicate elements.

Since Sets were defined to be immutable, all the Set methods return new Sets rather than modifying the currently existing Sets.

2XB3 - A1 - Implementing ADTs

Bassel Rezkalla - 400137450 - rezkallb

I also chose to use overloading to have 2 constructors for the Set ADT. One constructor for creating a Set with an array of Strings as an argument, and the other constructor for creating an empty set, taking no arguments.

For my toString method, I assumed that a result String in the form: "{a,b,c}" (excluding the outermost quotes" was acceptable. All the elements are enclosed in curly braces, and each element is separated by a comma.