By Anando Zaman

public class Set

Data type	Method name	Parameters	Description
Set Object	Set	**String[] arr1**	Constructor for creating a Set. Takes in a String arr1 of **max size 20** and creates a set object which contains the array(`this.Arr1)` along with the index of the last non-null value(`this.Arr1_index`). This is all done with the **assumption** that there are **no null values** passed into the **String[] arr1** in between the start and end of the array. For **example;** the **array {"1","2","3","4"} is valid but {"1",null, "2","3"} is NOT.** If an array of size > 20 is passed into the constructor, an error msg is printed and ends the constructor.
int	get_Arr1_index		Getter method used to return the `this.Arr1_index`
String[]	get_Arr1		Getter method used to return the `this.Arr1`
void	append	String s	Appends a (String s) to `this.Arr1`. This is with the **assumption** a **NON-NULL value** is appended. If size of `this.Arr1` is greater than 20, prints to the console "ARRAY IS FULL"
void	remove	String s	Finds the String s in the Set objects `this.Arr1` array. If not found, prints to the console "NOTHING TO REMOVE". If successfully finds the String in the Set, then removes the element and shifts over the elements to the right of the element by 1 space to the left. Last value becomes null and the Arr1_index decrements by 1. Example; Set set1 consists of `this.Arr1 = {"1", "2", "3", "4"} & this.Arr1_index = 3` `set1.remove("3") results in this.Arr1 = {"1", "2", "4"} & this.Arr1_index = 2`
Set Object	Union	Set set2	Return the union of this set with another set (i.e. set1 ∪ set1).
Set Object	Intersection	Set set2	Return the intersection of this set with another set (i.e. set1 ∩ set1).
Set Object	Difference	Set set2	Return the difference of this set with another set (i.e. R − S, the set of all elements in R that do not belong to S) (note: *this* set is R).

Set Object	Product	Set set2	Returns the Cartesian product (cross product) of this set by another set (i.e. this_set X
			Set2, the set of all possible pairs of concatenated elements.
			Example below:
			`String[] array2 = {"1","2","3"};`
			`Set p2 = new Set(array2);`
			
			`String[] array3 = {"1","2","3"};`
			`Set p3 = new Set(array3);`
			
			`Set prod = p2.Product(p3);`
			
			So prod would have {11, 12, 13, 21, 22, 23, 31, 32, 33}

boolean	isEqual	Set set2	Returns a Boolean value to check if two set objects are equivalent. Two sets are equivalent if they have the same length and are subsets of each other. Ie; {1,2,3,4} is equivalent to {1,2,3,4}. However, {1,2,3} is NOT equivalent to {1,2,3,4}. It is rather a subset of {1,2,3,4}
boolean	isSubset	Set set2	Returns a Boolean value to check if this_set is a subset of set2 object. Example; {1,2,3} is a subset of {1,2,3,4}. So returns true
int	getCount	No parameters	Returns an int value representing the number of elements in the Set object. Does not include null values of the array.
String	toString	No parameters	Return the elements of the set in the following format: {*elem*$_1$, *elem*$_2$, …, *elem*$_n$}