Damian Kryzia

CS2400-02

Project 2 - Set ADT

10/07/2022

Project Specification

The set ADT is implemented using a generic LinkedSet class that implements the generic SetInterface interface in Java. LinkedSet uses a list of linked node objects to store the contents of the set. This is achieved using the private Node class. Each node contains the data of a set entry and the reference to the next node. Below are the methods from SetInterface that represent all the functions of a set and when implemented by LinkedSet, manipulate the Nodes as needed:

Modifier and Type	Method	Description
boolean	add(T newEntry)	Adds a new entry to the set.
void	clear()	Clears the set.
boolean	contains(T anEntry)	Checks whether the set contains a specific entry.
boolean	equals(SetInterface<T> rhs)	Check whether a set is equal to this set.
int	getCurrentSize()	Returns the current size of the set.
boolean	isEmpty()	Checks whether the set is empty.
T	remove()	Removes the most recently added entry from the set.
boolean	remove(T anEntry)	Removes a specific entry from the set.
boolean	subset(SetInterface<T> rhs)	Checks whether a set is a subset of this set.
T[]	toArray()	Converts the contents of the set into an array.
String	toString()	Converts the set into a displayable string.
SetInterface<T>	union(SetInterface<T> rhs)	Creates a union of two sets.

Below is the implementation of the private Node class used by LinkedSet:

```
private class Node {

	private T data;
	private Node next;

	private Node(T dataPortion)
	{
		this(dataPortion, null);
	}

	private Node(T dataPortion, Node nextNode)
	{
		data = dataPortion;
		next = nextNode;
	}

}
```

Testing Methodology

The ADT is tested in the SetTest.java file which contains the main() method.

First, the union(), equals(), and subset() methods are tested using different pairs of sets. The cases include sets with multiple entries, some matching in both sets, some not; sets of different sizes, empty and non-empty sets, etc. This way, all scenarios are tested to prove the correctness of these methods.

Next, the other methods of SetInterface are tested:

- getCurrentSize() is tested for two sets of different sizes.
- isEmpty() is first tested on a non-empty set. Then, the set is cleared, subsequently testing the clear() method, and isEmpty() is tested on the now empty set.
- add() is tested in the case of adding a new entry (successful addition) and in the case of adding an already existing entry (failed addition).
- remove() is tested in the case of removing the most recent entry (no parameters) and in the case of removing a specific entry from the set.
- contains() is tested in the case of an entry that is known to exist inside the set and in the case of an entry that is known to not exist inside the set.
- toArray() is tested on a non-empty set, the array is displayed.
- toString() is tested as part of the other tests above as in many cases, the set is displayed to the user.