Elijah Tay, Alex Stalica Lab 10

LinkedSet.java

package linkedlist;

/\*\*

\* A quick demo of how to implement a set using a singly-linked list

\* @author trao

\*

\*/

public class LinkedSet<T> {

//---------------------------------------------------------------

//The body of the list is held in a singly-linked list

SLList<T> body;

//----------------------------------------------------------------

// Constructor creates a new singly-linked list for the body

public LinkedSet()

{

body = new SLList<T>();

}

//----------------------------------------------------------------

// member method simply delegates the membership testing to SLList

public boolean member(T elm)

{

return body.member(elm);

}

//----------------------------------------------------------------

// Inserts a new element in to the set, if it is not already there

public void insert(T str)

{

if(body.isEmpty()) {

body.insertAt(str,0);

}else {

boolean present = false;

SLNode<T> thisNode = this.body.getHead();

for(int i = 0; i < this.body.size(); i++) {

if(thisNode.getInfo() == str) {

present = true;

}

thisNode = thisNode.getNext();

}

if(!present) {

body.insertAt(str,0);

}

}

}

//----------------------------------------------------------------

// Simple printing method, delegates the job to SLList

public void printSet()

{

System.out.println (body.toString());

}

//----------------------------------------------------------------

// builds a set by inserting all elements of an array

public void buildSet(T[] elements)

{

for(T ele : elements) {

this.insert(ele);

}

}

//--------------------------------------------------------------

// Returns the union of this set and the other set without

// modifying this set or the other set

public LinkedSet<T> union(LinkedSet<T> otherSet)

{

LinkedSet<T> result = new LinkedSet<T>();

SLList<T> resultBody = result.body;

SLList<T> otherBody = otherSet.body;

//insert all elements from this list

SLNode thisNode = body.getHead();

for(int i = 0; i < body.size(); i++) {

result.insert((T)thisNode.getInfo());

thisNode = thisNode.getNext();

}

//use same loop from above with other list

//since insert method already does not insert

//duplicate elements, we can just call insert

SLNode otherNode = otherBody.getHead();

for(int i = 0; i < otherBody.size(); i++) {

result.insert((T)otherNode.getInfo());

otherNode = otherNode.getNext();

}

return result;

}

//--------------------------------------------------------------

// Returns the intersection of this set and the other set

// without modifying this set or the other set

public LinkedSet<T> intersection(LinkedSet<T> otherSet)

{

LinkedSet<T> result = new LinkedSet<T>();

SLList<T> resultBody = result.body;

SLList<T> otherBody = otherSet.body;

//Establish first nodes

SLNode<T> thisNode = body.getHead();

SLNode<T> otherNode = otherBody.getHead();

//loop over each node in this list

for(int i = 0; i < body.size(); i++) {

//for each node in this list, compare to each node in other list

for(int j = 0; j < otherBody.size(); j++) {

//if two nodes have the same info, insert to result list

if(thisNode.getInfo() == otherNode.getInfo()) {

result.insert((T)thisNode.getInfo());

}

otherNode = otherNode.getNext();

}

thisNode = thisNode.getNext();

otherNode = otherBody.getHead(); //reset otherNode to first node

}

return result;

}

}

LinkedSetTester.java

package linkedlist;

public class LinkedSetTester {

public static void main(String[] args) {

String[] s1 = {"abc", "defg", "abc", "ijkl"};

// As an array, repeated elements are OK

LinkedSet<String> set1 = new LinkedSet<String>();

set1.buildSet(s1); // but no repeated elements in Set

System.out.println("Set1");

System.out.println("---------------");

set1.printSet();

System.out.println("---------------");

String[] s2 = {"xyz", "defg", "abc", "pqr", "xyz"};

LinkedSet<String> set2 = new LinkedSet<String>();

set2.buildSet(s2);

System.out.println("Set2");

System.out.println("---------------");

set2.printSet();

System.out.println("---------------");

System.out.println("Set1 union Set2");

System.out.println("---------------");

set1.union(set2).printSet();

System.out.println("---------------");

System.out.println("Set1 intersection Set2");

System.out.println("---------------");

set1.intersection(set2).printSet();

System.out.println("---------------");

}

}

LinkedSetTester.java Output:

Set1

---------------

[ ijkl,defg,abc,]

---------------

Set2

---------------

[ pqr,abc,defg,xyz,]

---------------

Set1 union Set2

---------------

[ xyz,pqr,abc,defg,ijkl,]

---------------

Set1 intersection Set2

---------------

[ abc,defg,]

---------------