Name: Vangipuram Srinivasa Sarath Chandra
Regd No: 21BCE9853
Title: Digital Assignment Java
Date: 11/June/2022

1. Develop a java program to implement List interface by using stack class and linkedlist classes. (It should include all the methods of List interface). [5M]

List of methods used in stack & Linked List :

addAll	retainAll	removeAll	containsAll
remove	clear	add	addFirst
addLast	get	set	indexOf
lastIndexOf	size	toArray	iterator

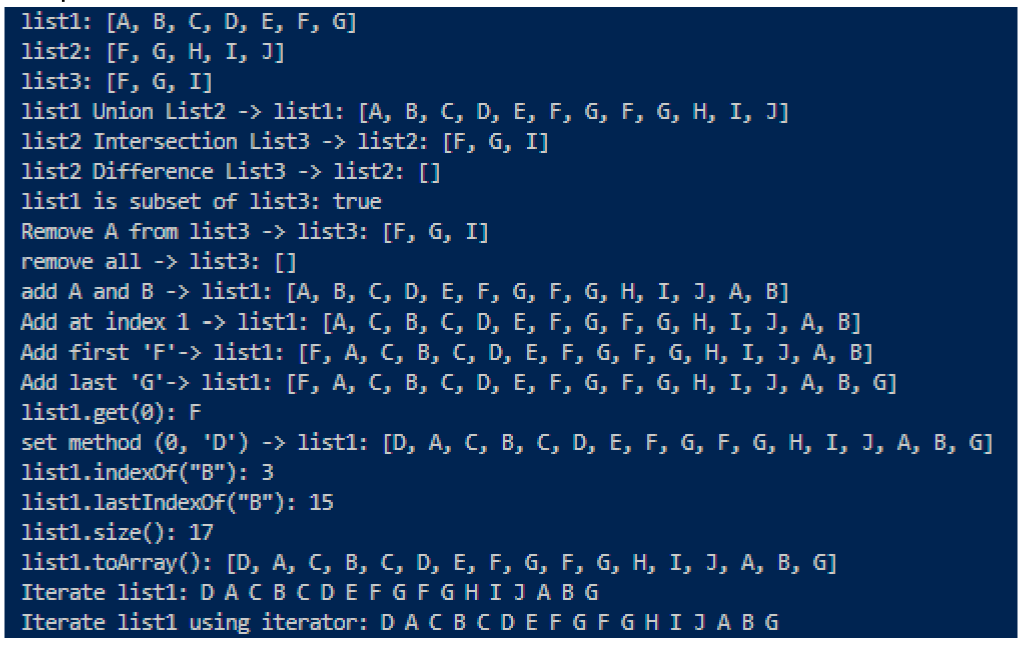
TLDR;

1. CRUD operations
2. Set operations
3. Iterator, size, toArray

Linked List

```java
import java.util.*;

public class LinkedListAssign {

    // creating a linked list using generics
    static <T> LinkedList<T> createLinkedList(T... ele) {
        LinkedList<T> newList = new LinkedList<T>();
        for (T el : ele) {
            newList.add(el);
        }
        return newList;
    }

    public static void main(String[] args) {

        LinkedList<String> list1 = createLinkedList("A", "B", "C", "D", "E", "F", "G");
        LinkedList<String> list2 = createLinkedList("F", "G", "H", "I", "J");
        LinkedList<String> list3 = createLinkedList("F", "G", "I");

        System.out.println("list1: " + list1);
        System.out.println("list2: " + list2);
        System.out.println("list3: " + list3);

        // union
        list1.addAll(list2);
        System.out.println("list1 Union List2 -> list1: " + list1);

        // intersection
        list2.retainAll(list3);
        System.out.println("list2 Intersection List3 -> list2: " + list2);

        // difference
        list2.removeAll(list3);
        System.out.println("list2 Difference List3 -> list2: " + list2);

        // subset
        System.out.println("list1 is subset of list3: " + list1.containsAll(list3));
        // remove
        list3.remove("A");
        System.out.println("Remove A from list3 -> list3: " + list3);

        // remove all
        // list3.removeAll();
        list3.clear();
        System.out.println("remove all -> list3: " + list3);

        // add
        list1.add("A");
        list1.add("B");
        System.out.println("add A and B -> list1: " + list1);

        // add at index
        list1.add(1, "C");
        System.out.println("Add at index 1 -> list1: " + list1);

        // add first and last
        list1.addFirst("F");
        System.out.println("Add first 'F'-> list1: " + list1);
        list1.addLast("G");
        System.out.println("Add last 'G'-> list1: " + list1);

        // get
        System.out.println("list1.get(0): " + list1.get(0));

        // set
        list1.set(0, "D");
        System.out.println("set method (0, 'D') -> list1: " + list1);

        // index of
        System.out.println("list1.indexOf(\"B\"): " + list1.indexOf("B"));

        // last index of
        System.out.println("list1.lastIndexOf(\"B\"): " + list1.lastIndexOf("B"));

        // size
        System.out.println("list1.size(): " + list1.size());

        // toArray
        Object[] array = list1.toArray();
        System.out.println("list1.toArray(): " + Arrays.toString(array));

        // iterate
        System.out.print("Iterate list1: ");
        for (String s : list1) {
            System.out.print(s + ' ');
        }

        // iterate using iterator
        System.out.print("\nIterate list1 using iterator: ");
        Iterator<String> itr = list1.iterator();
        while (itr.hasNext()) {
            System.out.print(itr.next() + ' ');
        }
    }
}
```

Output:



```
list1: [A, B, C, D, E, F, G]
list2: [F, G, H, I, J]
list3: [F, G, I]
list1 Union List2 -> list1: [A, B, C, D, E, F, G, F, G, H, I, J]
list2 Intersection List3 -> list2: [F, G, I]
list2 Difference List3 -> list2: []
list1 is subset of list3: true
Remove A from list3 -> list3: [F, G, I]
remove all -> list3: []
add A and B -> list1: [A, B, C, D, E, F, G, F, G, H, I, J, A, B]
Add at index 1 -> list1: [A, C, B, C, D, E, F, G, F, G, H, I, J, A, B]
Add first 'F'-> list1: [F, A, C, B, C, D, E, F, G, F, G, H, I, J, A, B]
Add last 'G'-> list1: [F, A, C, B, C, D, E, F, G, F, G, H, I, J, A, B, G]
list1.get(0): F
set method (0, 'D') -> list1: [D, A, C, B, C, D, E, F, G, F, G, H, I, J, A, B, G]
list1.indexOf("B"): 3
list1.lastIndexOf("B"): 15
list1.size(): 17
list1.toArray(): [D, A, C, B, C, D, E, F, G, F, G, H, I, J, A, B, G]
Iterate list1: D A C B C D E F G F G H I J A B G
Iterate list1 using iterator: D A C B C D E F G F G H I J A B G
```

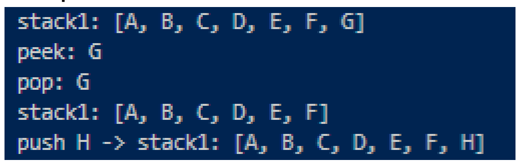
Stack

All the methods metioned in the above LinkedList hold true here too and are similar if not same in every aspect. Altough they acheive the same things they just go about it in a different way here are the methods which aren't implemented in LinkedList and are exclusively in stack:

- pop()
- push()
- peek()

```java
import java.util.*;

public class StackAssign {
    // creating a stack using generics
    static <T> Stack<T> createStack(T... ele) {
        Stack<T> newStack = new Stack<T>();
        for (T el : ele) {
            newStack.push(el);
        }
        return newStack;
    }

    public static void main(String[] args) {
        Stack<String> stack1 = createStack("A", "B", "C", "D", "E", "F", "G");

        System.out.println("stack1: " + stack1);

        // peek
        System.out.println("peek: " + stack1.peek());

        // pop
        System.out.println("pop: " + stack1.pop());
        System.out.println("stack1: " + stack1);

        // push
        stack1.push("H");
        System.out.println("push H -> stack1: " + stack1);

    }
}
```

Output:



```
stack1: [A, B, C, D, E, F, G]
peek: G
pop: G
stack1: [A, B, C, D, E, F]
push H -> stack1: [A, B, C, D, E, F, H]
```