Object Oriented programming lab

Submitted by:

Jisha Chacko

S2RMCA:A batch

Roll no:44

1).**Java program to create generic stack and do the push and pop operation**

A stack class is provided by the Java collection framework and it implements the Stack data structure. The stack implements LIFO i.e. Last In First Out. This means that the elements pushed last are the ones that are popped first.

1. push() Method adds element x to the stack.
2. pop() Method removes *the* last element of *the* stack.
3. top() Method returns the last element of the stack.
4. empty() Method returns whether *the* stack is empty or not.

```
import java.io.*;

import java.util.*;

public class Example {

  public static void main (String[] args) {

    Stack<Integer> s = new Stack<Integer>();

    s.push(5);

    s.push(1);

    s.push(9);

    s.push(4);

    s.push(8);

    System.out.print("The stack is: " + s);

    System.out.print("\nThe element popped is: ");

    Integer num1 = (Integer) s.pop();

    System.out.print(num1);

    System.out.print("\nThe stack after pop is: " + s);

    Integer pos = (Integer) s.search(9);

    if(pos == -1)

      System.out.print("\nThe element 9 not found in stack");

    else

      System.out.print("\nThe element 9 is found at position " + pos + " in stack");

  }

}
```

output

```
The stack is: [5, 1, 9, 4, 8]
The element popped is: 8
The stack after pop is: [5, 1, 9, 4]
The element 9 is found at position 2 in stack
```

2.Generic method implement bubble sort

Bubble sort is a simple sorting algorithm. This sorting algorithm is a comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order. This algorithm is not suitable for large datasets as its average and worst case complexity is of O(n2) where n is the number of items.

```
public class BubbleSort {
static void bubbleSort(int[] arr) {
int n = arr.length;
int temp = 0;

for(int i = 0; i < n; i++) {
for(int j=1; j < (n-i); j++) {
if(arr[j-1] > arr[j]) {
temp = arr[j-1];
arr[j-1] = arr[j];
arr[j] = temp;
}
}
}
}
public static void main(String[] args) {
int arr[] = { 2, 5, -2, 6, -3, 8, 0, -7, -9, 4 };
System.out.println("Array Before Bubble Sort");

for(int i = 0; i < arr.length; i++) {
System.out.print(arr[i] + " ");
}
System.out.println();
bubbleSort(arr);
System.out.println("Array After Bubble Sort");

for(int i = 0; i < arr.length; i++) {
System.out.print(arr[i] + " ");
}
}
}
```

Output

```
 Array Before Bubble Sort
 2 5 -2 6 -3 8 0 -7 -9 4
 Array After Bubble Sort
 -9 -7 -3 -2 0 2 4 5 6 8
```

3.Maintain a list of string using arraylist from a collection of framework, perform built-in operation

The ArrayList class extends AbstractList and implements the List interface. ArrayList supports dynamic arrays that can grow as needed.

Standard Java arrays are of a fixed length. After arrays are created, they cannot grow or shrink, which means that you must know in advance how many elements an array will hold.

Array lists are created with an initial size. When this size is exceeded, the collection is automatically enlarged. When objects are removed, the array may be shrunk.

```
import java.util.*;
public class ArrayListDemo {

  public static void main(String args[]) {
    // create an array list
    ArrayList al = new ArrayList();
    System.out.println("Initial size of al: " + al.size());
```

```
      // add elements to the array list
      al.add("C");
      al.add("A");
      al.add("E");
      al.add("B");
      al.add("D");
      al.add("F");
      al.add(1, "A2");
      System.out.println("Size of al after additions: " + al.size());

      // display the array list
      System.out.println("Contents of al: " + al);

      // Remove elements from the array list
      al.remove("F");
      al.remove(2);
      System.out.println("Size of al after deletions: " + al.size());
      System.out.println("Contents of al: " + al);
   }
}
```

output

```
Initial size of al: 0
Size of al after additions: 7
Contents of al: [C, A2, A, E, B, D, F]
Size of al after deletions: 5
Contents of al: [C, A2, E, B, D]
```