Code:

```
/*
 * Name: Shahad Eid Albalawi
 * ID: 438002072
 * Section: 4C3
 */
package shahad_eid_albalawi_438002072;

import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;
import java.util.Stack;

/**
 *
 * @author Dell
 */
public class patientInfo {

    private String PatientName;
    private int PatientID;

    public String getPatientName() {
        return PatientName;
    }

    public void setPatientName(String PatientName) {
        this.PatientName = PatientName;
    }

    public int getPatientID() {
        return PatientID;
    }

    public void setPatientID(int PatientID) {
        this.PatientID = PatientID;
    }

    @Override
    public String toString() {
        return "patientInfo{" + "PatientName=" + PatientName + ",
PatientID=" + PatientID + '}';
    }
```

```
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int size1, size2;
        String ADT1 = "", ADT2 = "";

        System.out.println("*****************************************
*****************************************");
        System.out.println("* 1:Merging two Single Linked List structure
type                                        *");
        System.out.println("* 2:Merging two Stacks data structure
type                                            *");
        System.out.println("* 3:Merging two Queses data structure
type                                            *");
        System.out.println("* 4:Merging Single Linked List with Stack to
Linked List                                 *");
        System.out.println("* 5:Merging Single Linked List with Queue to
Linked List                                 *");
        System.out.println("* 6:Merging Singly Linked List with Stack only
Patient's name that start with S to Queue *");
        System.out.println("*****************************************
*****************************************");
        System.out.println("Input Choise >>> ");
        int choise = input.nextInt();

        switch (choise) {
            case 1:
                ADT1 = "Single Linked List";
                ADT2 = "Single Linked List";
                size1 = readSize(ADT1, "First");
                size2 = readSize(ADT2, "Second");
                LinkedList<patientInfo> list1,
                 list2;
                LinkedList<patientInfo> mergedList = new
LinkedList<patientInfo>();
                list1 = readLinkedList(size1);
                mergedList.addAll(list1);
                list2 = readLinkedList(size2);
                mergedList.addAll(list2);

                System.out.println(" ----------- Before Merge -----------
");
                System.out.println("---- list1 ");
                System.out.println(list1);
                System.out.println("---- list2 ");
                System.out.println(list2);
```

```
                System.out.println(" ----------- After Merge -----------
");
                System.out.println("---- mergedList ");
                System.out.println(mergedList);
                break;

            case 2:
                ADT1 = "Stack";
                ADT2 = "Stack";
                size1 = readSize(ADT1, "First");
                size2 = readSize(ADT2, "Second");
                Stack<patientInfo> stack1,
                 stack2;
                Stack<patientInfo> mergedStack = new Stack<patientInfo>();
                stack1 = readStack(size1);
                mergedStack.addAll(stack1);
                stack2 = readStack(size2);
                mergedStack.addAll(stack2);

                System.out.println(" ----------- Before Merge -----------
");
                System.out.println("---- stack1 ");
                System.out.println(stack1);
                System.out.println("---- stack2 ");
                System.out.println(stack2);
                System.out.println(" ----------- After Merge -----------
");
                System.out.println("---- mergedStack ");
                System.out.println(mergedStack);

                break;
            case 3:
                ADT1 = "Queue";
                ADT2 = "Queue";
                size1 = readSize(ADT1, "First");
                size2 = readSize(ADT2, "Second");
                Queue<patientInfo> queue1,
                 queue2;
                Queue<patientInfo> mergedQueue = new
LinkedList<patientInfo>();
                queue1 = readQueue(size1);
                mergedQueue.addAll(queue1);
                queue2 = readQueue(size2);
                mergedQueue.addAll(queue2);
```

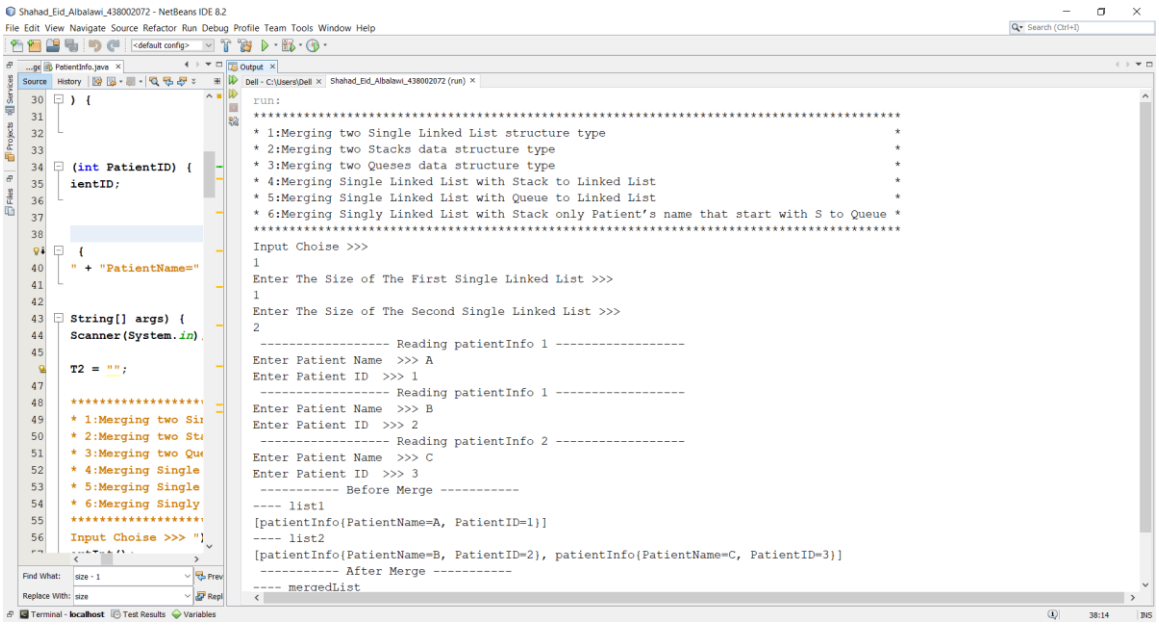
```
                System.out.println(" ----------- Before Merge -----------
");
                System.out.println("---- queue1 ");
                System.out.println(queue1);
                System.out.println("---- queue2 ");
                System.out.println(queue2);
                System.out.println(" ----------- After Merge -----------
");
                System.out.println("---- mergedQueue ");
                System.out.println(mergedQueue);

                break;
            case 4:
                ADT1 = "Single Linked List";
                ADT2 = "Stack";
                size1 = readSize(ADT1, "First");
                size2 = readSize(ADT2, "Second");
                LinkedList<patientInfo> list1_1;
                Stack<patientInfo> stack2_1;
                LinkedList<patientInfo> mergedlist1_1 = new
LinkedList<patientInfo>();
                list1_1 = readLinkedList(size1);
                mergedlist1_1.addAll(list1_1);
                stack2_1 = readStack(size2);
                mergedlist1_1.addAll(stack2_1);

                System.out.println(" ----------- Before Merge -----------
");
                System.out.println("---- list1 ");
                System.out.println(list1_1);
                System.out.println("---- stack2 ");
                System.out.println(stack2_1);
                System.out.println(" ----------- After Merge -----------
");
                System.out.println("---- mergedList ");
                System.out.println(mergedlist1_1);

                break;
            case 5:
                ADT1 = "Single Linked List";
                ADT2 = "Queue";
                size1 = readSize(ADT1, "First");
                size2 = readSize(ADT2, "Second");
                LinkedList<patientInfo> list1_2;
                Queue<patientInfo> queue2_1;
```

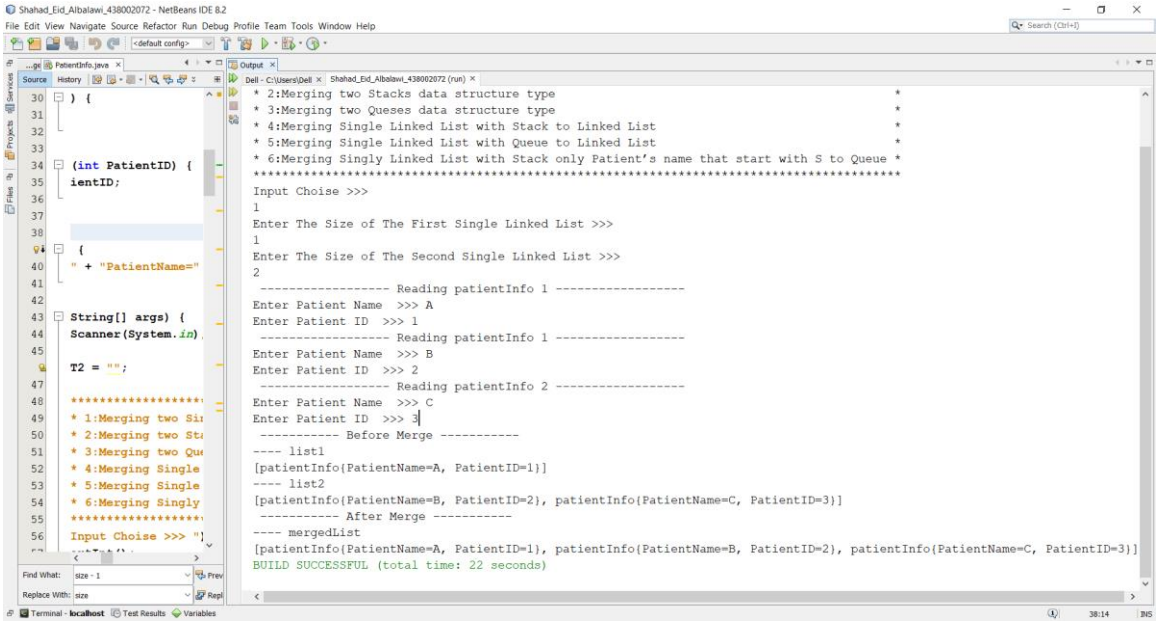
```
                LinkedList<patientInfo> mergedlist1_2 = new
LinkedList<patientInfo>();
                list1_2 = readLinkedList(size1);
                mergedlist1_2.addAll(list1_2);
                queue2_1 = readQueue(size2);
                mergedlist1_2.addAll(queue2_1);

                System.out.println(" ----------- Before Merge -----------
");
                System.out.println("---- list1 ");
                System.out.println(list1_2);
                System.out.println("---- queue2 ");
                System.out.println(queue2_1);
                System.out.println(" ----------- After Merge -----------
");
                System.out.println("---- mergedList ");
                System.out.println(mergedlist1_2);

                break;
            case 6:
                ADT1 = "Single Linked List";
                ADT2 = "Stack";
                size1 = readSize(ADT1, "First");
                size2 = readSize(ADT2, "Second");
                LinkedList<patientInfo> list1_3;
                Stack<patientInfo> stack2_2;
                Queue<patientInfo> mergedlist1_3 = new
LinkedList<patientInfo>();
                list1_3 = readLinkedList(size1);
                mergedlist1_3.addAll(list1_3);
                stack2_2 = readStack(size2);
                for (patientInfo info : stack2_2) {
                    if (info.PatientName.startsWith("S")) {
                        mergedlist1_3.add(info);
                    }
                }

                System.out.println(" ----------- Before Merge -----------
");
                System.out.println("---- list1 ");
                System.out.println(list1_3);
                System.out.println("---- stack2 ");
                System.out.println(stack2_2);
                System.out.println(" ----------- After Merge -----------
");
```

```
                System.out.println("---- mergedList ");
                System.out.println(mergedlist1_3);

                break;
            default:
                System.err.println("Invalid Input");
                System.exit(0);
        }

    }

    public static int readSize(String ADT, String Order) {
        System.out.println("Enter The Size of The " + Order + " " + ADT +
" >>> ");
        return new Scanner(System.in).nextInt();
    }

    public static patientInfo readPatientInfo(int Order) {
        patientInfo info = new patientInfo();
        System.out.println(" ------------------ Reading patientInfo " +
Order + " ------------------ ");
        System.out.print("Enter Patient Name  >>> ");
        info.setPatientName(new Scanner(System.in).next());
        System.out.print("Enter Patient ID  >>> ");
        info.setPatientID(new Scanner(System.in).nextInt());
        return info;
    }

    public static LinkedList<patientInfo> readLinkedList(int size) {
        LinkedList<patientInfo> infos = new LinkedList<>();
        for (int i = 0; i < size; i++) {
            infos.add(readPatientInfo(i + 1));
        }

        return infos;
    }

    public static Stack<patientInfo> readStack(int size) {
        Stack<patientInfo> infos = new Stack<>();
        for (int i = 0; i < size; i++) {
            infos.push(readPatientInfo(i + 1));
        }

        return infos;
    }
```

```
    public static Queue<patientInfo> readQueue(int size) {
        Queue<patientInfo> infos = new LinkedList<>();
        for (int i = 0; i < size; i++) {
            infos.add(readPatientInfo(i + 1));
        }

        return infos;
    }

}
```

Output:




Shahad_Eid_Albalawi_438002072 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
PatientInfo.java
Source History
) {
(int PatientID) {
ientID;
{
" + "PatientName="
String[] args) {
Scanner(System.in)
T2 = "";
* 1:Merging two Si
* 2:Merging two St
* 3:Merging two Qu
* 4:Merging Single
* 5:Merging Single
* 6:Merging Singly
Input Choise >>> "
Find What: size - 1
Replace With: size
Output
Dell - C:\Users\Dell
Shahad_Eid_Albalawi_438002072 (run)
* 2:Merging two Stacks data structure type *
* 3:Merging two Queses data structure type *
* 4:Merging Single Linked List with Stack to Linked List *
* 5:Merging Single Linked List with Queue to Linked List *
* 6:Merging Singly Linked List with Stack only Patient's name that start with S to Queue *
Input Choise >>>
1
Enter The Size of The First Single Linked List >>>
1
Enter The Size of The Second Single Linked List >>>
2
---------- Reading patientInfo 1 ----------
Enter Patient Name >>> A
Enter Patient ID >>> 1
---------- Reading patientInfo 1 ----------
Enter Patient Name >>> B
Enter Patient ID >>> 2
---------- Reading patientInfo 2 ----------
Enter Patient Name >>> C
Enter Patient ID >>> 3
---------- Before Merge ----------
---- list1
[patientInfo{PatientName=A, PatientID=1}]
---- list2
[patientInfo{PatientName=B, PatientID=2}, patientInfo{PatientName=C, PatientID=3}]
---------- After Merge ----------
---- mergedList
[patientInfo{PatientName=A, PatientID=1}, patientInfo{PatientName=B, PatientID=2}, patientInfo{PatientName=C, PatientID=3}]
BUILD SUCCESSFUL (total time: 22 seconds)
Terminal - localhost
Test Results
Variables
38:14
INS