**TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES**

**QUEZON CITY**

**COLLEGE OF INFORMATION TECHNOLOGY EDUCATION (CITE)**

**CS 201 - Data Structures and Algorithms**

|  |  |
| --- | --- |
| **Name: Aristotle Buenaventura** | **Date: September 20, 2021** |
| **Program/Section: IT21S1** | **Instructor: Ms. Rosmina Joy M. Cabauatan** |
| **Assessment Task: Quiz 4** | |

**Code**

**RunnerSLL3 Class**

package quiz4;

// Name: Aristotle Buenaventura

// Section: IT12S1

import javax.swing.JOptionPane;

public class RunnerSLL3 extends Inventory {

public static void main(String[] args) {

byte size = 10;

System.out.println("Please input patient's name");

System.out.println("Patient's condition" );

System.out.println("[P]Pregnant");

System.out.println("[S]Senior Citizen");

System.out.println("[D]Disabled");

System.out.println("[N]Normal");

System.out.println();

Inventory list = new Inventory();

int seniorCount=0;

int pregnantCount=0;

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

String name = JOptionPane.showInputDialog("Name: ");

System.out.println("Name: " + name);

String condition = JOptionPane.showInputDialog("Condition: ").toUpperCase();

System.out.print("Condition: ");

if (condition.equals("P")) {

System.out.println("Pregnant");

if (seniorCount ==0) {

list.prepend(name);

}

else{

list.add(seniorCount + 1,name);

}

++pregnantCount;

list.print();

}else if(condition.equals("S")) {

System.out.println("Senior Citizen");

list.prepend(name);

++seniorCount;

list.print();

}else if(condition.equals("D")) {

System.out.println("Disabled");

if (seniorCount ==0) {

list.prepend(name);

} else if (pregnantCount == 0) {

list.add(seniorCount+1,name);

}else{

list.add(seniorCount + pregnantCount + 1,name);

}

list.print();

}else if(condition.equals("N")) {

System.out.println("Normal");

list.append(name);

list.print();

}else {

System.out.println("Wrong Input!");

}

if(i<size-1) {

System.out.println("(Next patient)");

System.out.println();

}

}

list.printAll();

}

}

**Inventory Class**

package quiz4;

public class Inventory {

// reference to head / first node of the Singly Linked List

public Node head = null;

// class Node that hold data and a reference/link

// to the next Node in the list

class Node {

private String data;

private Node next;

public Node(String name) {

this.data = name;

this.next = null;

}

}

/\*

\* Method to add a node at the beginning of the list

\*/

public void prepend(String name) {

// Create a new node

Node newNode = new Node(name);

// Check if the list is empty

if (this.head == null) {

// Make the new node as head

this.head = newNode;

} else {

// Point the new node's next to head

newNode.next = this.head;

// Make the new node as head

this.head = newNode;

}

}

/\*

\* Method to add a node at the end of the list

\*/

public void append(String data) {

// Create a new node

Node newNode = new Node(data);

// Check if the list is empty

if (this.head == null) {

// Make the new node as head

this.head = newNode;

} else {

Node cur = this.head;

// traverse to the end of the list

while (cur.next != null) {

cur = cur.next;

}

cur.next = newNode;

}

}

/\*

\* Method to add a node at the specified position in the list

\*/

public void add(int position, String data) {

// Create a new node

Node newNode = new Node(data);

// Init the cur and prev nodes to the head

Node cur = this.head, prev = this.head;

if (position == 0) {

// Point the new node's next to head

newNode.next = head;

// Make the new node as head

this.head = newNode;

return;

}

// traverse to the end of the list and check positions moved

while (--position > 0) {

// update the prev and cur references

prev = cur;

cur = cur.next;

}

// update prev to point to new node

prev.next = newNode;

// & new node to point to current node

newNode.next = cur;

}

public void print() {

if (this.head == null) {

System.out.println("The List is empty.");

} else {

System.out.print("Patient's list: ");

Node cur = this.head;

while (cur != null) {

System.out.print(cur.data + " -> ");

cur = cur.next;

}

System.out.println("NULL\n");

}

}

public void printAll() {

System.out.print("Total Patient's list: ");

Node cur = this.head;

while (cur != null) {

System.out.print(cur.data + ", ");

cur = cur.next;

}

System.out.println("<- Last Patient");

}

}// End of the program

**Output**

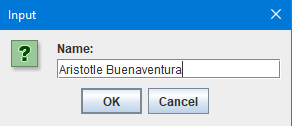
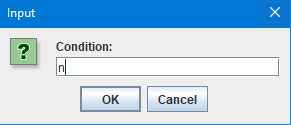
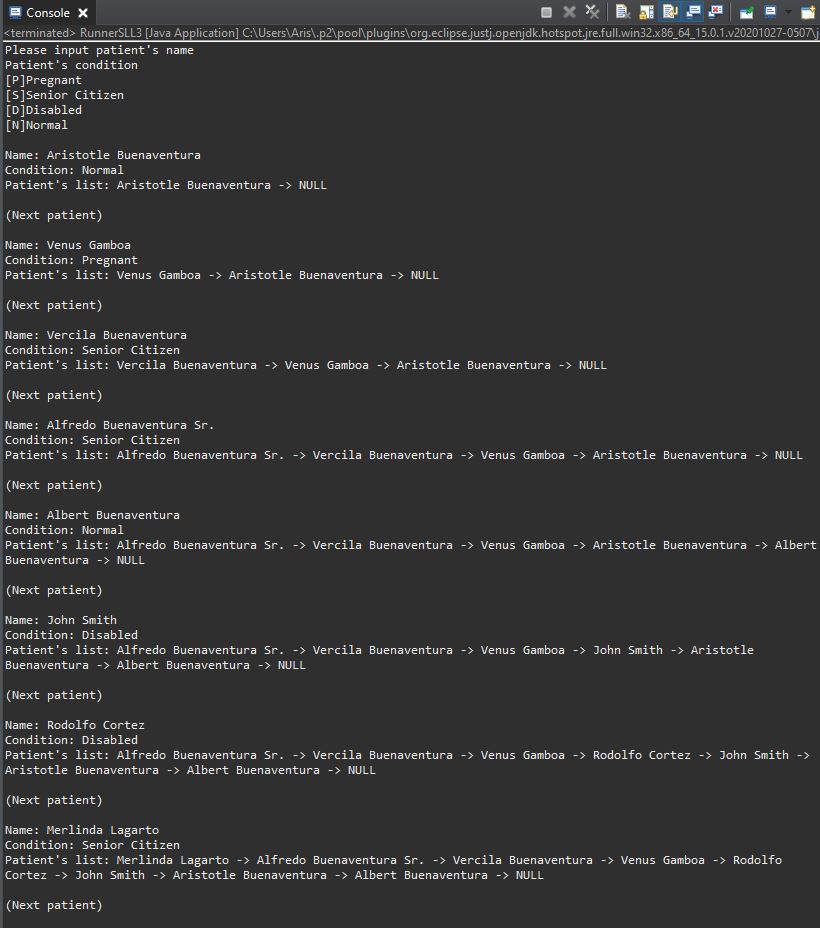


Figure 1. Enter the Patient’s Name

Figure 2. Enter the Patient’s Condition

****

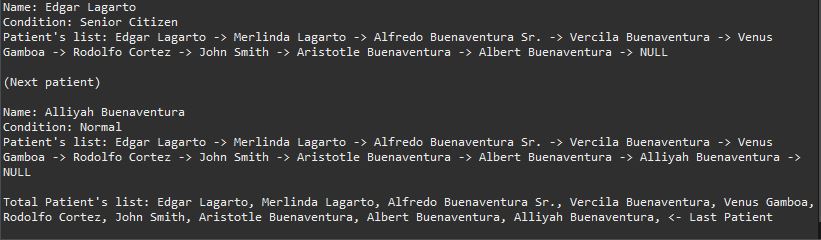
****

Figure 3. The Final Output