Vanna Moore

CMPS 390

Program 4

/\* This program creates a linked list that loads numbers 0-9 and prints it to the console. It then prompts the user to enter the amount of integers they want in their list. Once they enter an amount, the program prompts them to enter a value for each list item. Once they've entered each value, the program prints the list. I used the Scanner class for the user input, and created classes for node and list and used the methods from those classes to create the lists and show them on the screen. \*/

**Main Method**

import java.util.Scanner;

public class babyList{

public static void main(String[] args){

node nodeA = new node();

list listA = new list();

listA.init();

for(int i = 0; i < 10; i++){

listA.addLast(i);

}

System.out.println("Contents of linked list A:");

listA.showList();

System.out.println();

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the number of integers in your list. ");

int size = scanner.nextInt();

list listB = new list();

listB.init();

if(size != 0){

System.out.print("What is the first number?");

int num0 = scanner.nextInt();

listB.addFirst(num0);

for(int j = 1; j < size; j++){

System.out.print("What is the next number?");

int k = scanner.nextInt();

listB.addLast(k);

}

}

System.out.println("The contents of your list is:");

listB.showList();

}

}

**Main Method Screenshots**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Node Class**

public class node{

int data;

node next;

}

**Node Class Screenshot**

**A screenshot of a computer

Description automatically generated**

**List Class Code**

public class list{

int data;

node next;

node curr;

node front;

node tail;

node newNode;

public void init(){

front = null;

}

public node makeNode(int data){

newNode = new node();

newNode.data = data;

newNode.next = null;

return newNode;

}

public node addFirst(int data){

front = makeNode(data);

tail = front;

return front;

}

public node addLast(int data){

if(front == null){

front = makeNode(data);

}

else{

tail = findTail();

tail.next = makeNode(data);

tail = tail.next;

tail.next = null;

}

return tail;

}

public node findTail(){

node curr;

curr = front;

while(curr.next != null){

curr = curr.next;

}

return curr;

}

public void showList(){

node curr;

curr = front;

while(curr != null){

System.out.println(curr.data);

curr = curr.next;

}

}

public void buildList(int length){

int j;

init();

for(j = 0; j < length; j++){

if(j == 0){

node front = makeNode(j);

}

else{

tail = findTail();

tail.next = makeNode(j);

}

}

}

}

**List Class Screenshots**

**A screenshot of a computer

Description automatically generated**

**A computer screen shot of a program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Code Running**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**