LAB 4: Linked List 2

Statement Purpose:

Purpose of this Lab is to familiarize the students with writing simple Java programs using Linked List. The students are given small tasks related to Linked List which they complete during the lab session under the supervision of the lab instructor. This helps them understand the concepts well which they learnt in their lectures.

Activity Outcomes:

The students will learn how to write methods related to some actions performed on Linked List. They will understand how to write simple methods to

- swap the data of first and last nodes in the linked list
- count and display number of even nodes and also number of odd nodes present in the linked list
- add a new node after some specific node of the linked list
- delete a node after some specific node of the linked list

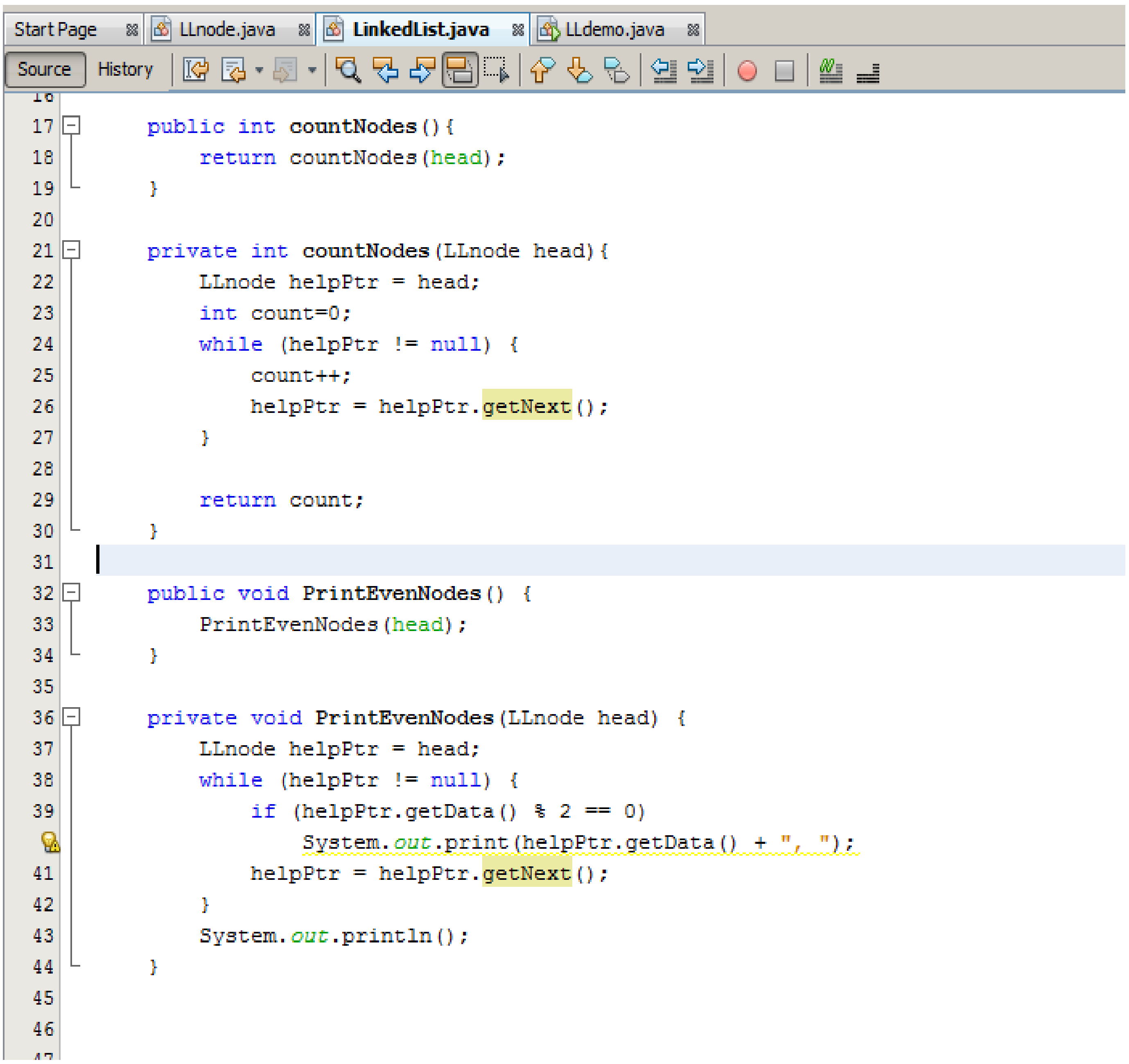
Practice Activity with Lab Instructor: (20 Minutes)

Make a new project in NetBeans called LinkedListProject, and then copy the LLnode, LinkedList, and the main LLdemo class into this project. Your job is to write two new methods. The first should count the number of nodes, and the second should print all even nodes in the linked list.

Solution:

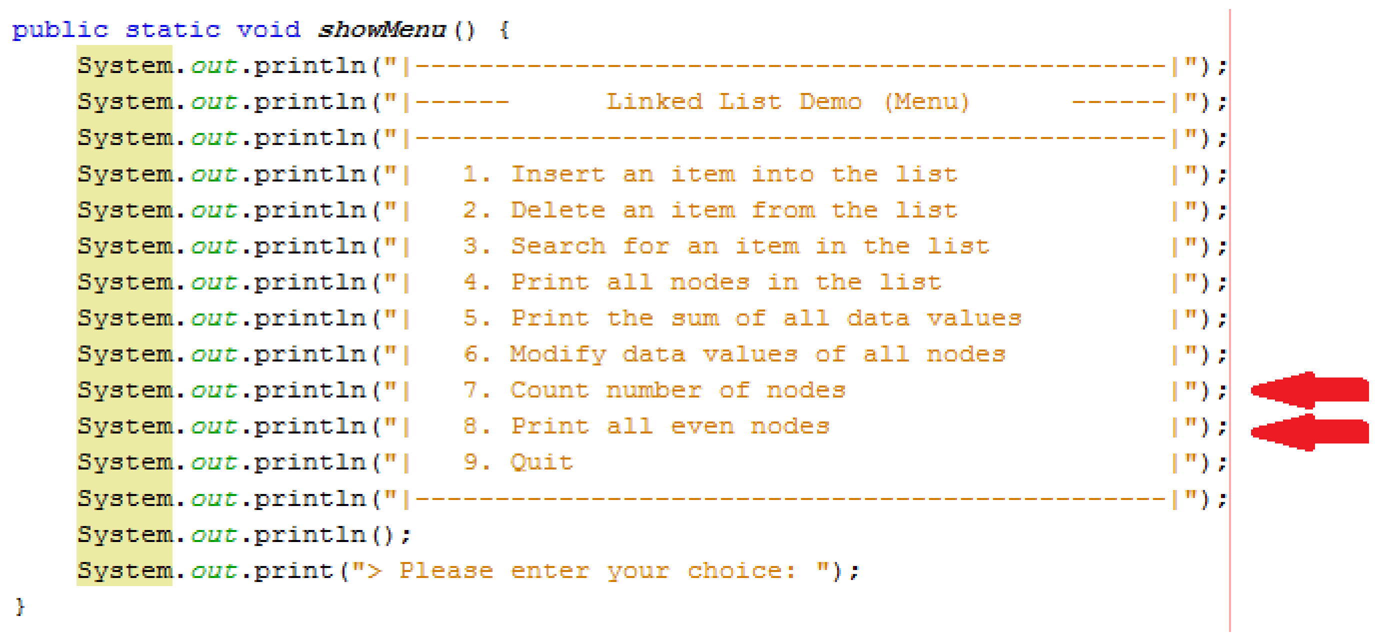
1. Create a project with the name LinkListProject
2. Create a package with the name LinkListUpdated
3. Copy LLnode class in this package.
4. Copy LinkedList class in this package and add the following methods in this class.

LAB 4: Linked List 2



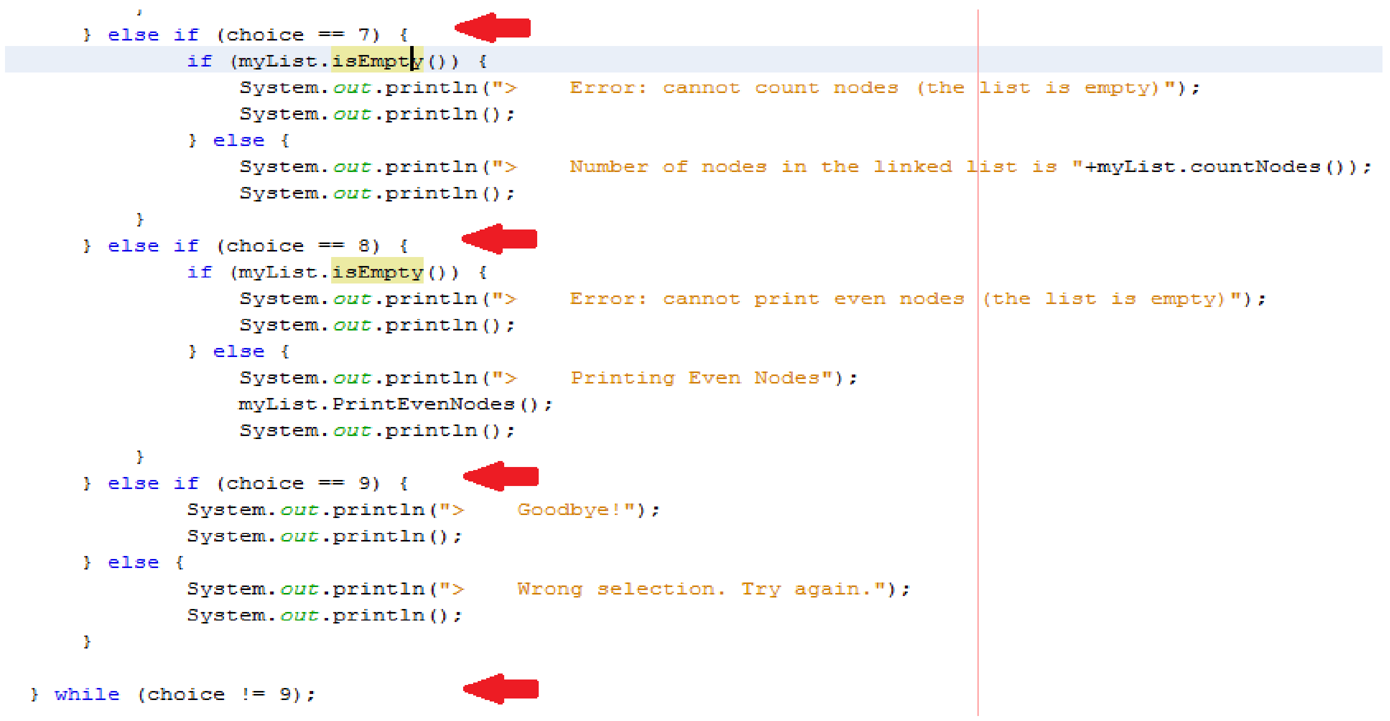
```
    public int countNodes(){
        return countNodes(head);
    }

    private int countNodes(LLnode head){
        LLnode helpPtr = head;
        int count=0;
        while (helpPtr != null) {
            count++;
            helpPtr = helpPtr.getNext();
        }

        return count;
    }

    public void PrintEvenNodes() {
        PrintEvenNodes(head);
    }

    private void PrintEvenNodes(LLnode head) {
        LLnode helpPtr = head;
        while (helpPtr != null) {
            if (helpPtr.getData() % 2 == 0)
                System.out.print(helpPtr.getData() + ", ");
            helpPtr = helpPtr.getNext();
        }
        System.out.println();
    }
```

5. Copy LLdemo main class in this package.
6. Add the following lines of code (shown with red arrow) in the showMenu() method of this class.



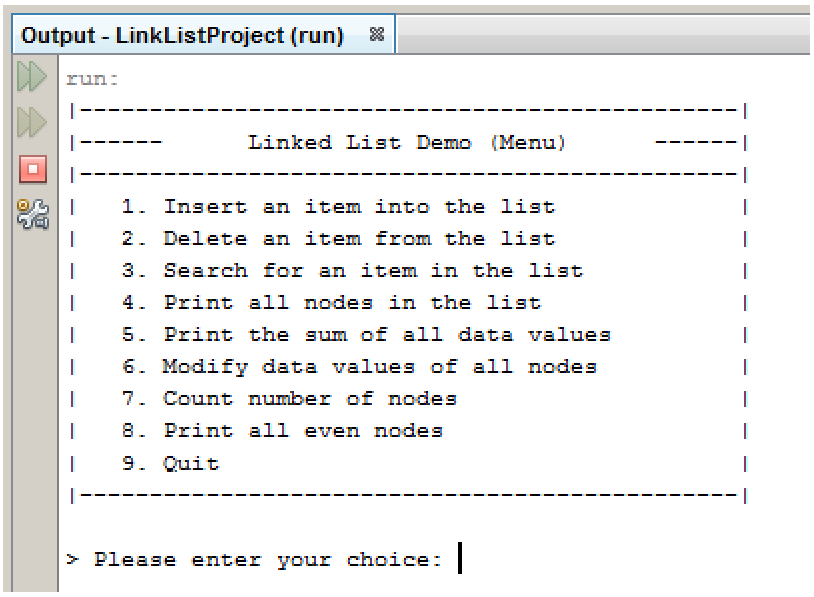
```
public static void showMenu() {
    System.out.println("|------------------------------------------------|");
    System.out.println("|------      Linked List Demo (Menu)       ------|");
    System.out.println("|------------------------------------------------|");
    System.out.println("|    1. Insert an item into the list             |");
    System.out.println("|    2. Delete an item from the list             |");
    System.out.println("|    3. Search for an item in the list           |");
    System.out.println("|    4. Print all nodes in the list              |");
    System.out.println("|    5. Print the sum of all data values         |");
    System.out.println("|    6. Modify data values of all nodes          |");
    System.out.println("|    7. Count number of nodes                    |");
    System.out.println("|    8. Print all even nodes                     |");
    System.out.println("|    9. Quit                                     |");
    System.out.println("|------------------------------------------------|");
    System.out.println();
    System.out.print("> Please enter your choice: ");
}
```

7. Add the following lines of code in the main() method of this class



```
    } else if (choice == 7) {
            if (myList.isEmpty()) {
                System.out.println(">    Error: cannot count nodes (the list is empty)");
                System.out.println();
            } else {
                System.out.println(">    Number of nodes in the linked list is "+myList.countNodes());
                System.out.println();
        }
    } else if (choice == 8) {
            if (myList.isEmpty()) {
                System.out.println(">    Error: cannot print even nodes (the list is empty)");
                System.out.println();
            } else {
                System.out.println(">    Printing Even Nodes");
                myList.PrintEvenNodes();
                System.out.println();
        }
    } else if (choice == 9) {
            System.out.println(">    Goodbye!");
            System.out.println();
    } else {
            System.out.println(">    Wrong selection. Try again.");
            System.out.println();
    }

} while (choice != 9);
```

8. **When you run this program, it will display the following menu on the screen:**


Output - LinkListProject (run)

```
run:
|-----------------------------------------------|
|------      Linked List Demo (Menu)      ------|
|-----------------------------------------------|
|   1. Insert an item into the list             |
|   2. Delete an item from the list             |
|   3. Search for an item in the list           |
|   4. Print all nodes in the list              |
|   5. Print the sum of all data values         |
|   6. Modify data values of all nodes          |
|   7. Count number of nodes                    |
|   8. Print all even nodes                     |
|   9. Quit                                     |
|-----------------------------------------------|

> Please enter your choice: 
```

9. **After inserting some nodes in the linked list, when you press 7, it will display the total number of nodes present in the linked list. If you press 8, it will display only even nodes in the linked list.**

LAB EXERCISES: (60 Minutes)

1. Add a method named swapFirstLast() in the above program which swaps the data of first and last nodes in the linked list.

   Hint: Use following method header:

   ```
   public void swapFirstLast(){
           swapFirstLast(head);
   }

   private void swapFirstLast(LLnode head) ){

                   . . . . . . . . . .

   }
   ```

2. Add a method named countEvenOddNodes() in the above program which counts and displays number of even nodes as well as number of and odd nodes present in the linked list.

   Hint: Use following method header:

   ```
   public void countEvenOddNodes(){
           countEvenOddNodes(head);
   }

   private void countEvenOddNodes(LLnode head) ){
           int EvenNodes=0;
           int OddNodes=0;

            . . . . . . . . . .

           System.out.println("Number of even nodes in the list is "+EvenNodes);
           System.out.println("Number of odd nodes in the list is "+OddNodes);

   }
   ```

3. Add a method named addNodeAfter() in the above program which adds a new node after some specific node of the linked list. The data of new node should be equal to double the data of specified node. For example, if we want to add a node after the node 30, the new node should have data equal to 60 and is added after the node 30. If specified node is not present in the list, display some error message.

Hint: Use following method header:

```
public void addNodeAfter (int data){
        addNodeAfter (head, data);
}

private void addNodeAfter (LLnode head, int data){

                    . . . . . . . . . . .

}
```

4. Add a method named deleteNodeAfter() in the above program which deletes a node after some specific node of the linked list. If specified node is not present in the list, display some error message. If this is last node in the linked list, display some appropriate message like "No node can be deleted after this node as it is the last node in the list".

Hint: Use following method header:

```
public void deleteNodeAfter(int data){

        deleteNodeAfter(head, data);

}
private void deleteNodeAfter(LLnode head, int data){

                . . . . . . . . . . .

}
```