

```

1 package Project4;
2
3
4 import java.io.Serializable;
5
6 /*****
7  * This program creates a Node that is used in the second Linked List.
8  * It holds 3 pieces of data. The auto, the Next NodeUndo, and a variable
9  * that represents the action done within the primary Linked List.
10 *
11 * @author Justin Von Kulajta Winn and Nick Layman
12 * @version 1.9
13 *****/
14
15 public class NodeUndo implements Serializable {
16
17     /** this is the auto being held by each NodeUndo */
18     private Auto dataU;
19
20     /** This is the next NodeUndo in the Linked List */
21     private NodeUndo nextU;
22
23     /** This represents the action done within the primary Linked List */
24     private int RemoveOrAdd;
25
26     /*****
27     * This is a constructor that sets the class' variables equal to the
28     * passed variables.
29     * @param data is the auto that is to be set to the NodeUndo's data
30     * @param next is the NodeUndo that the current NodeUndo points to
31     * @param ROA is the integer that represents the action done in the
32     * primary code
33     *****/
34     public NodeUndo(Auto data, NodeUndo next, int ROA) {
35         this.dataU = data;
36         this.nextU = next;
37         this.RemoveOrAdd = ROA;
38     }
39
40     /*****
41     * This function returns the auto being held in the current NodeUndo
42     * @return the auto within the current NodeUndo
43     *****/
44     public Auto getDataU() {
45         return dataU;
46     }
47
48     /*****
49     * This function sets the auto being held in the current NodeUndo
50     * @param dataU is the auto that is to be held in the current NodeUndo
51     *****/
52     public void setDataU(Auto dataU) {
53         this.dataU = dataU;
54     }
55
56     /*****

```

```

57     * This function returns the NodeUndo that the current NodeUndo is
58     * pointing to.
59     * @return the NodeUndo that is pointed to
60     *****/
61     public NodeUndo getNextU() {
62         return nextU;
63     }
64
65     /*****
66     * This function sets the NodeUndo that current NodeUndo is pointing to.
67     * @param nextU is the NodeUndo that current NodeUndo is to be pointed to
68     *****/
69     public void setNextU(NodeUndo nextU) {
70         this.nextU = nextU;
71     }
72
73     /*****
74     * This function returns the value of the RemoveOrAdd of the current
75     * NodeUndo
76     * @return the value of RemoveOrAdd
77     *****/
78     public int getRemoveOrAdd() {
79         return RemoveOrAdd;
80     }
81
82     /*****
83     * This function sets the value of the RemoveOrAdd
84     * @param removeOrAdd is the value that is to be set to the RemoveOrAdd
85     *****/
86     public void setRemoveOrAdd(int removeOrAdd) {
87         RemoveOrAdd = removeOrAdd;
88     }
89 }
90

```