

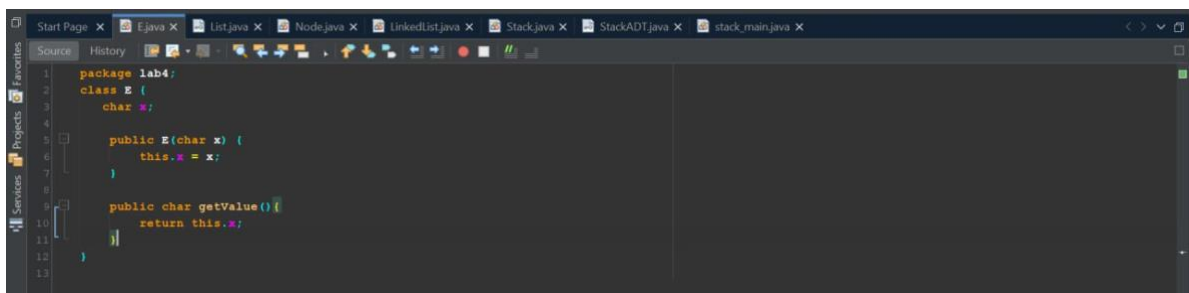
Name : Abdulmajeed Abdullah Almazmomi.

ID : 2240297.

Section : T66.

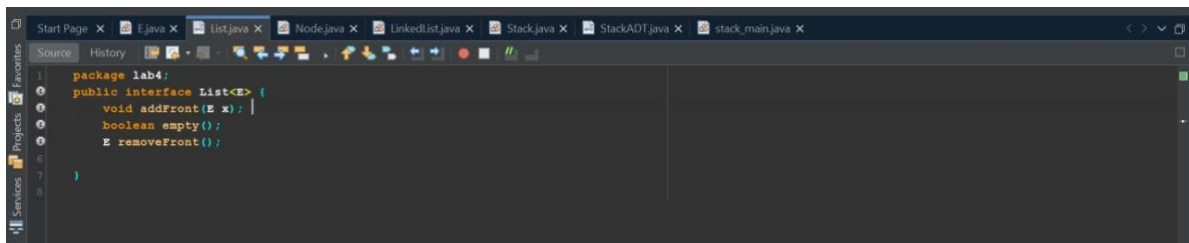
LAB : 5.

E class:



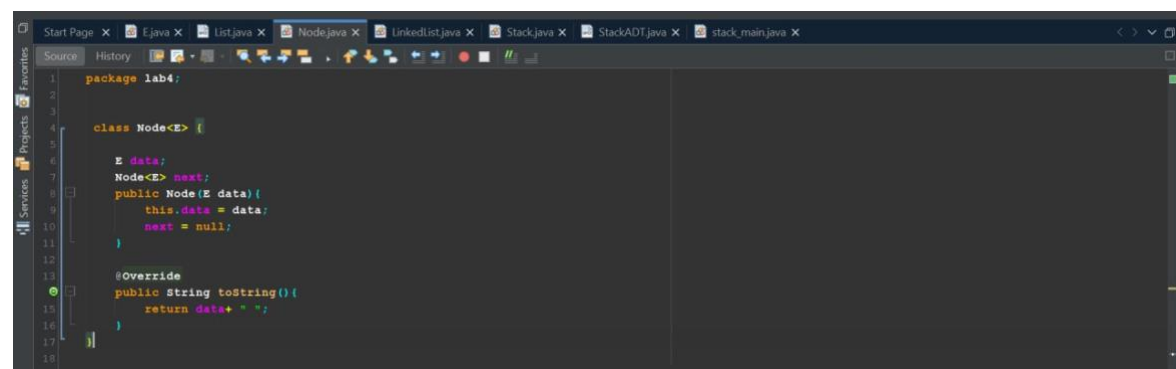
```
1 package lab4;
2 class E {
3     char x;
4
5     public E(char x) {
6         this.x = x;
7     }
8
9     public char getValue() {
10        return this.x;
11    }
12 }
13
```

List class:



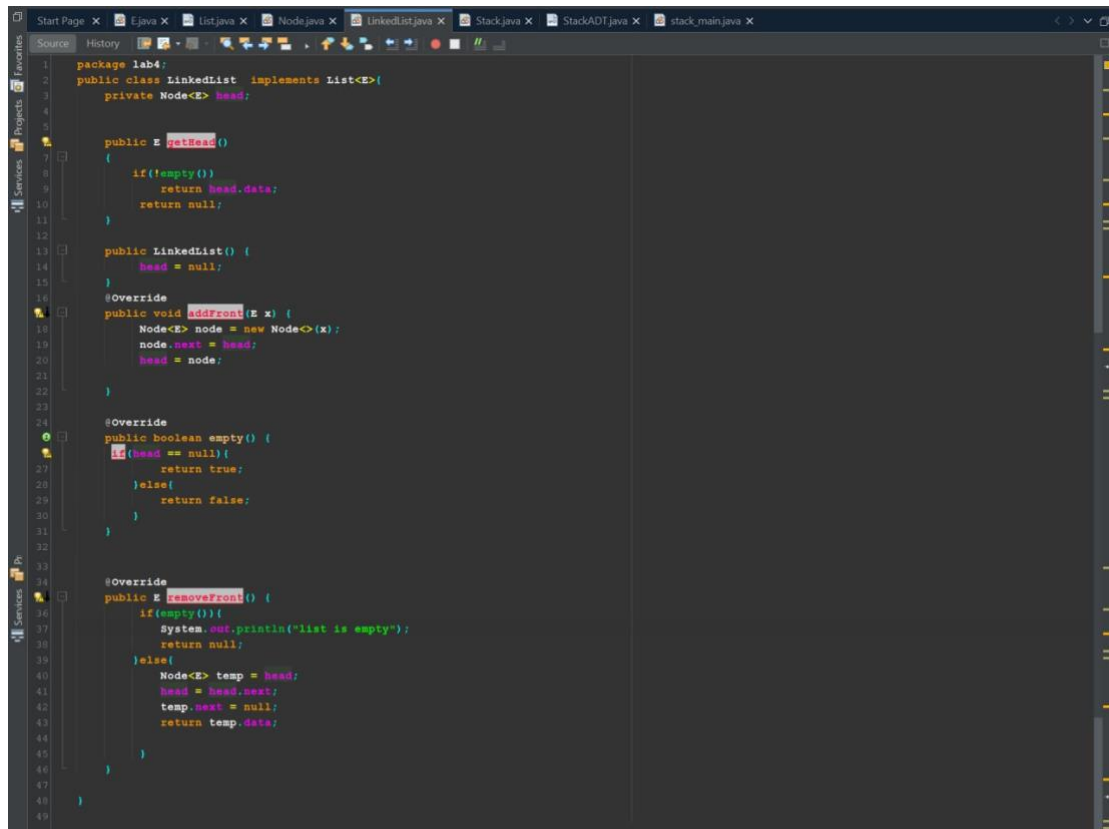
```
1 package lab4;
2 public interface List<E> {
3     void addFront(E x);
4     boolean empty();
5     E removeFront();
6 }
7
```

Node class:



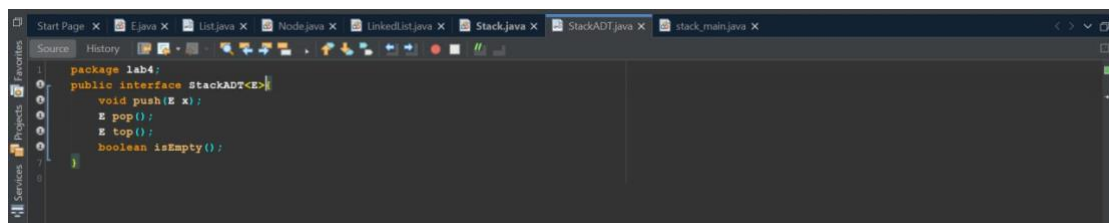
```
1 package lab4;
2
3 class Node<E> {
4     E data;
5     Node<E> next;
6     Node(E data) {
7         this.data = data;
8         next = null;
9     }
10
11     @Override
12     public String toString() {
13         return data + " ";
14     }
15 }
16
```

Linkedlist class:



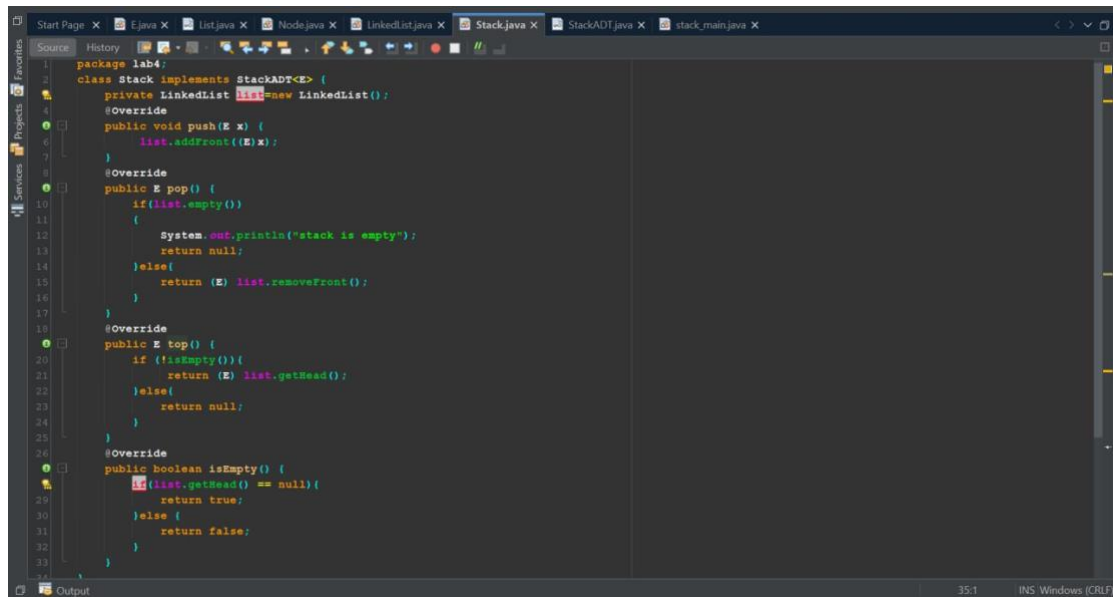
```
1 package lab4;
2 public class LinkedList implements List<E>{
3     private Node<E> head;
4
5
6     public E getHead()
7     {
8         if(!empty())
9             return head.data;
10        return null;
11    }
12
13    public LinkedList() {
14        head = null;
15    }
16    @Override
17    public void addFront(E x) {
18        Node<E> node = new Node<>(x);
19        node.next = head;
20        head = node;
21    }
22
23    @Override
24    public boolean empty() {
25        if(head == null){
26            return true;
27        }else{
28            return false;
29        }
30    }
31
32    @Override
33    public E removeFront() {
34        if(empty()){
35            System.out.println("list is empty");
36            return null;
37        }else{
38            Node<E> temp = head;
39            head = head.next;
40            temp.next = null;
41            return temp.data;
42        }
43    }
44 }
45
46 }
```

StackADT:



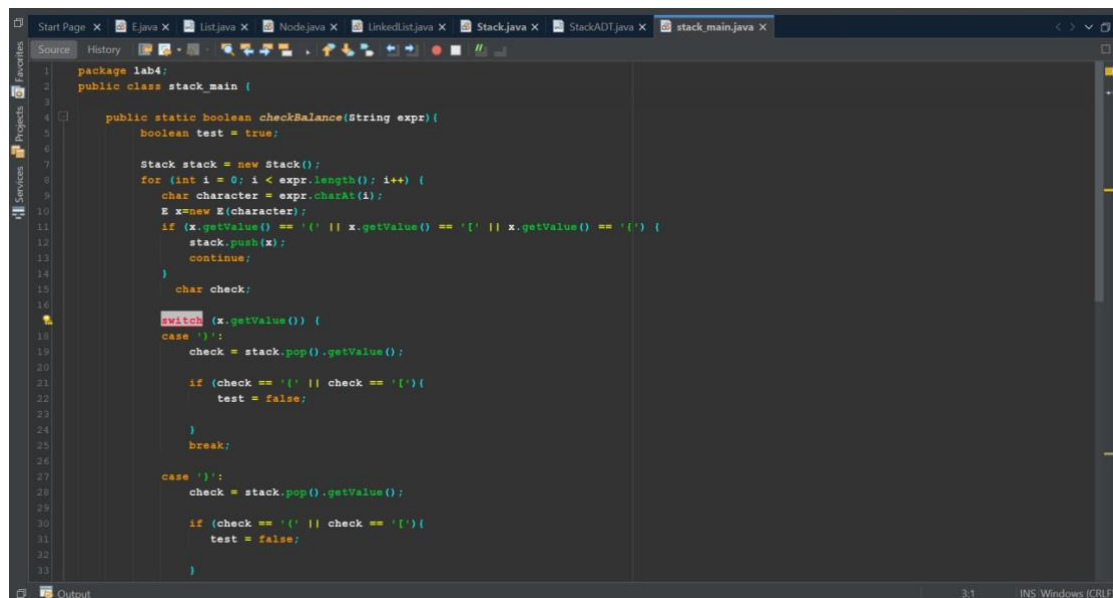
```
1 package lab4;
2 public interface StackADT<E>{
3     void push(E x);
4     E pop();
5     E top();
6     boolean isEmpty();
7 }
```

Stack class:

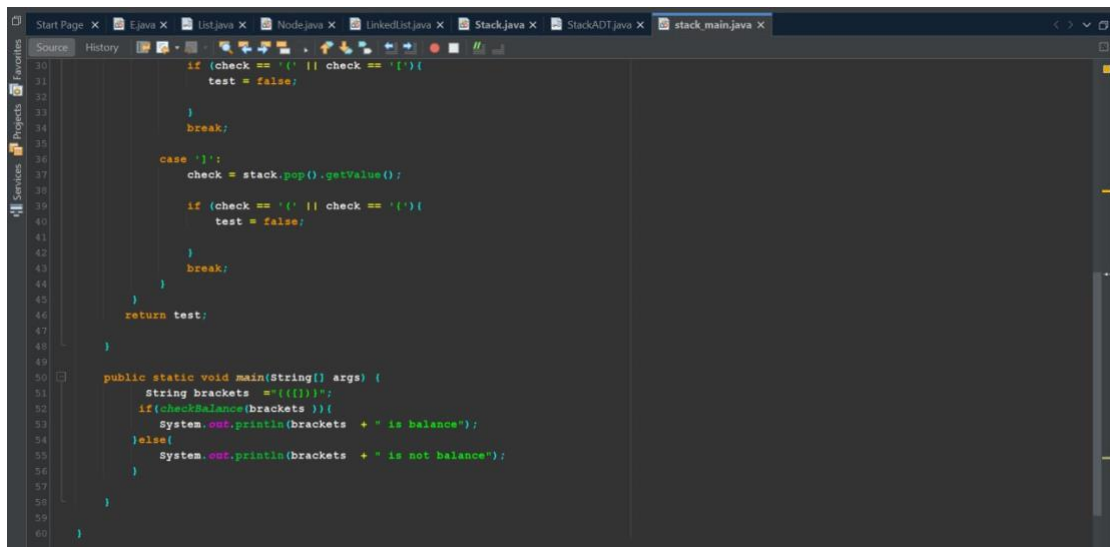


```
1 package lab4;
2 class Stack implements StackADT<E> {
3     private LinkedList<E> list = new LinkedList();
4     @Override
5     public void push(E x) {
6         list.addFront(x);
7     }
8     @Override
9     public E pop() {
10        if (list.isEmpty())
11        {
12            System.out.println("stack is empty");
13            return null;
14        }
15        return (E) list.removeFront();
16    }
17    @Override
18    public E top() {
19        if (!isEmpty()) {
20            return (E) list.getHead();
21        }
22        return null;
23    }
24    @Override
25    public boolean isEmpty() {
26        return (list.getHead() == null);
27    }
28    }
29    }
30    }
31    }
32    }
33    }
```

Stack_main class:

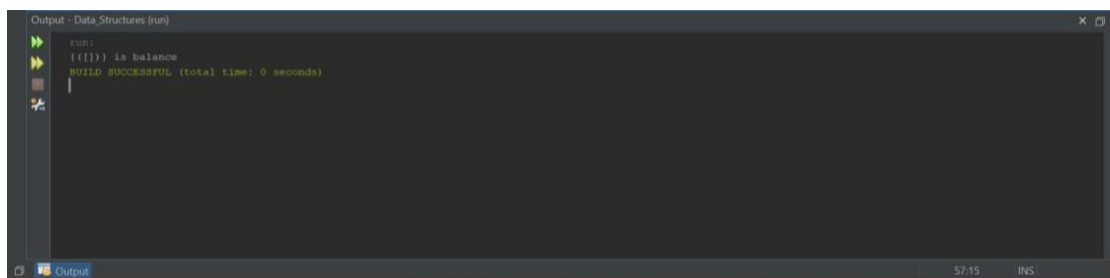


```
1 package lab4;
2 public class stack_main {
3
4     public static boolean checkBalance(String expr){
5         boolean test = true;
6
7         Stack stack = new Stack();
8         for (int i = 0; i < expr.length(); i++) {
9             char character = expr.charAt(i);
10            E x = new E(character);
11            if (x.getValue() == '(' || x.getValue() == '[' || x.getValue() == '{') {
12                stack.push(x);
13                continue;
14            }
15            char check;
16            switch (x.getValue()) {
17                case ')':
18                    check = stack.pop().getValue();
19                    if (check == '(' || check == '[' || check == '{') {
20                        test = false;
21                    }
22                    break;
23                case '}':
24                    check = stack.pop().getValue();
25                    if (check == '(' || check == '[' || check == '{') {
26                        test = false;
27                    }
28            }
29        }
30        return test;
31    }
32 }
```

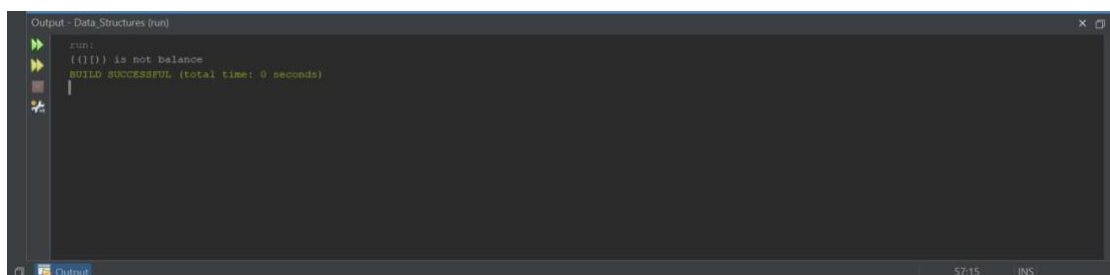


```
30         if (check == '(' || check == '['){
31             test = false;
32         }
33     }
34     break;
35
36     case ')':
37         check = stack.pop().getValue();
38
39         if (check == '(' || check == '['){
40             test = false;
41         }
42     }
43     break;
44 }
45
46 return test;
47
48 }
49
50 public static void main(String[] args) {
51     String brackets = "({[]})";
52     if (checkBalance(brackets)) {
53         System.out.println(brackets + " is balance");
54     } else {
55         System.out.println(brackets + " is not balance");
56     }
57 }
58
59 }
60 }
```

The output:



```
run:
({[]}) is balance
BUILD SUCCESSFUL (total time: 0 seconds)
```



```
run:
({[]}) is not balance
BUILD SUCCESSFUL (total time: 0 seconds)
```