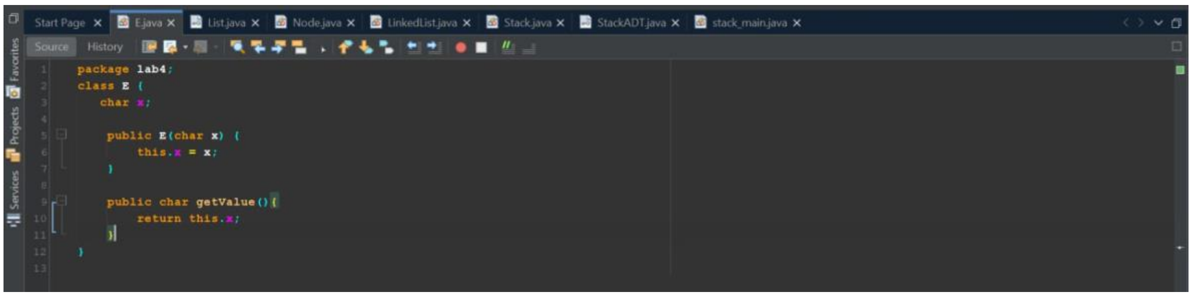
Name : Abdulmajeed Abdullah Almazmomi.

ID : 2240297.

Section : T66.

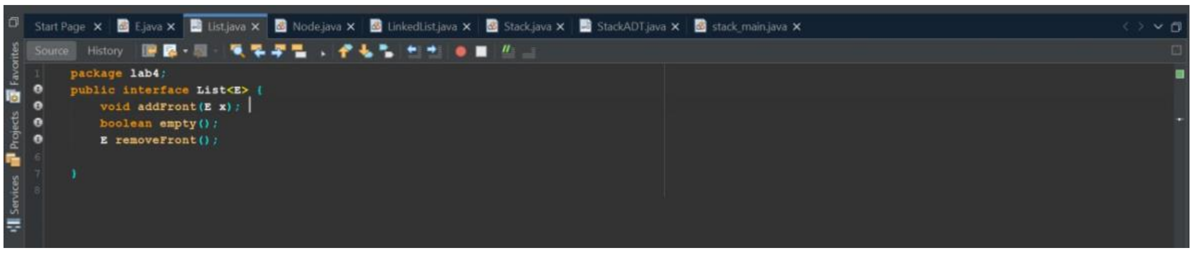
LAB : 5.

E class:



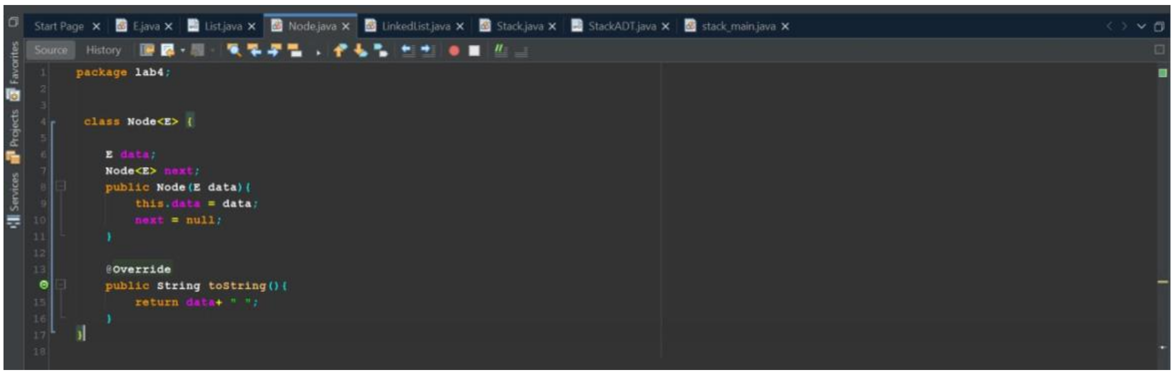
```java
package lab4;
class E {
   char x;

    public E(char x) {
        this.x = x;
    }

    public char getValue(){
        return this.x;
    }
}
```

List class:



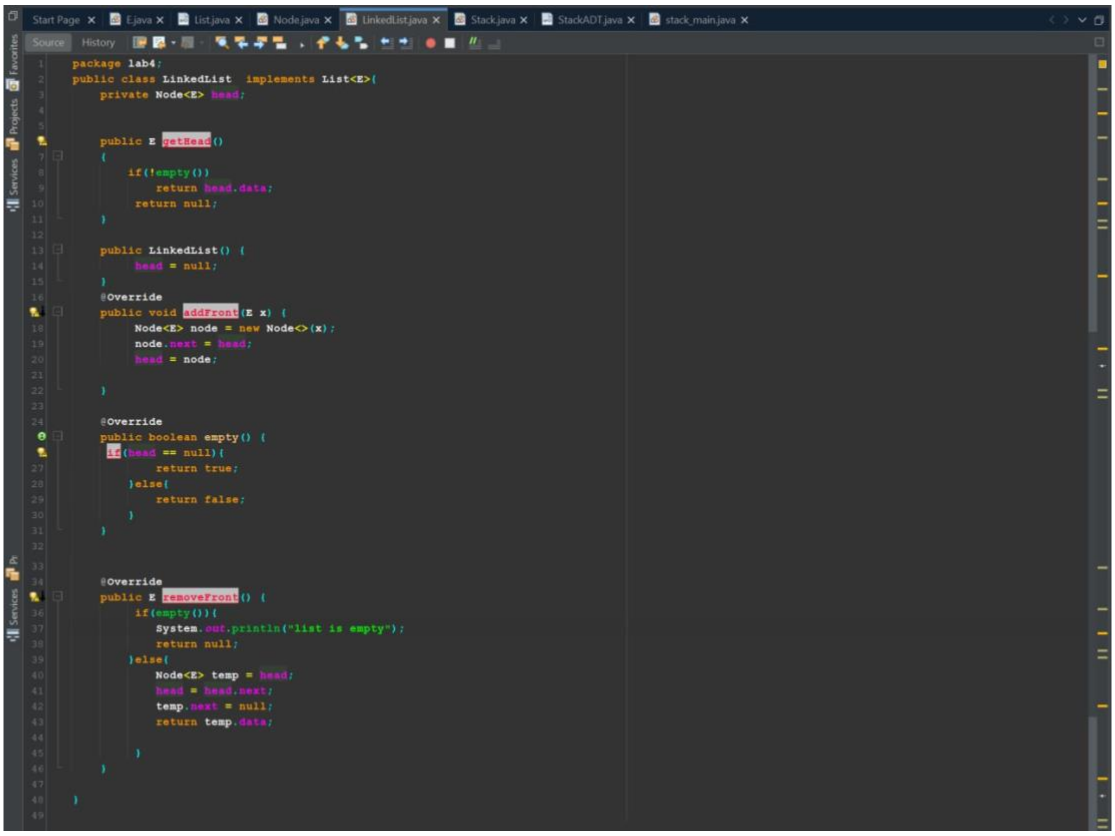
```java
package lab4;
public interface List<E> {
    void addFront(E x);
    boolean empty();
    E removeFront();

}
```

Node class:



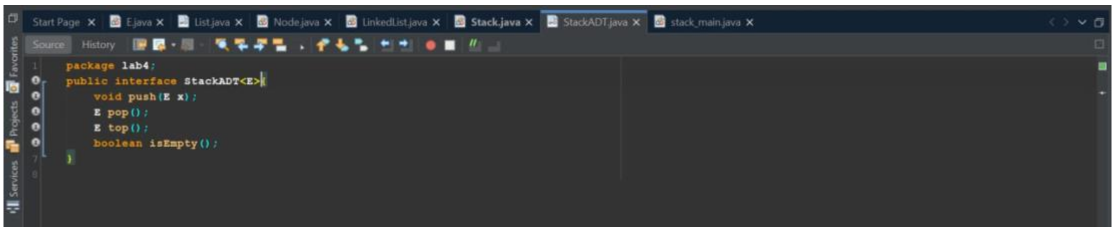
```java
package lab4;


 class Node<E> {

    E data;
    Node<E> next;
    public Node(E data){
        this.data = data;
        next = null;
    }

    @Override
    public String toString(){
        return data+ " ";
    }
}
```

Linkedlist class:



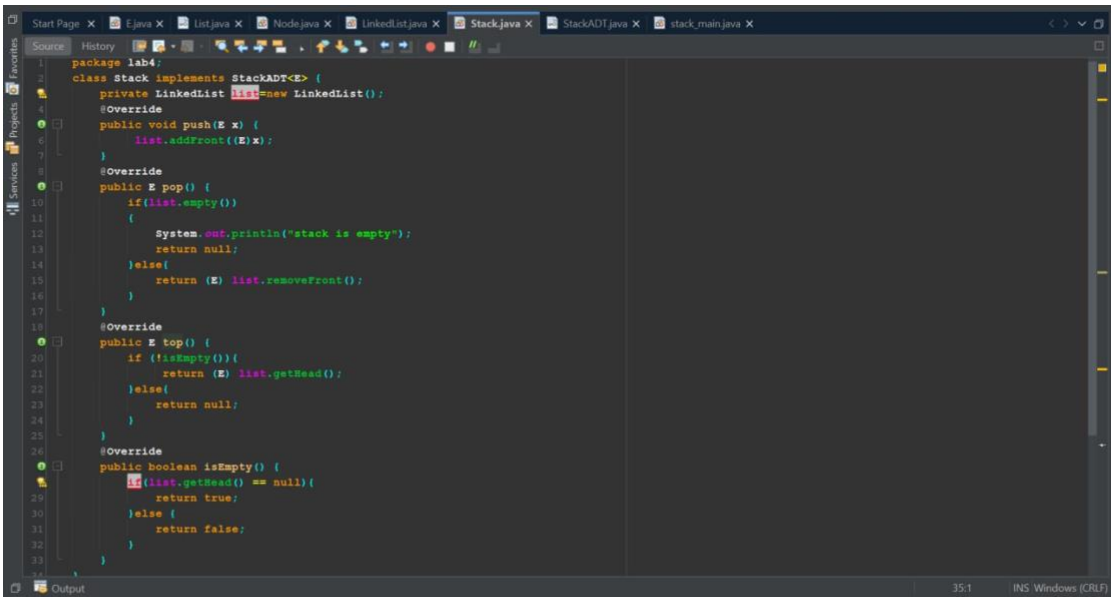
```
package lab4;
public class LinkedList  implements List<E>{
    private Node<E> head;


    public E getHead()
    {
        if(!empty())
            return head.data;
         return null;
    }

    public LinkedList() {
         head = null;
    }
    @Override
    public void addFront(E x) {
         Node<E> node = new Node<>(x);
         node.next = head;
         head = node;

    }

    @Override
    public boolean empty() {
     if(head == null){
            return true;
        }else{
            return false;
        }
    }


    @Override
    public E removeFront() {
         if(empty()){
            System.out.println("list is empty");
            return null;
        }else{
            Node<E> temp = head;
            head = head.next;
            temp.next = null;
            return temp.data;

         }
    }

}
```

StackADT:



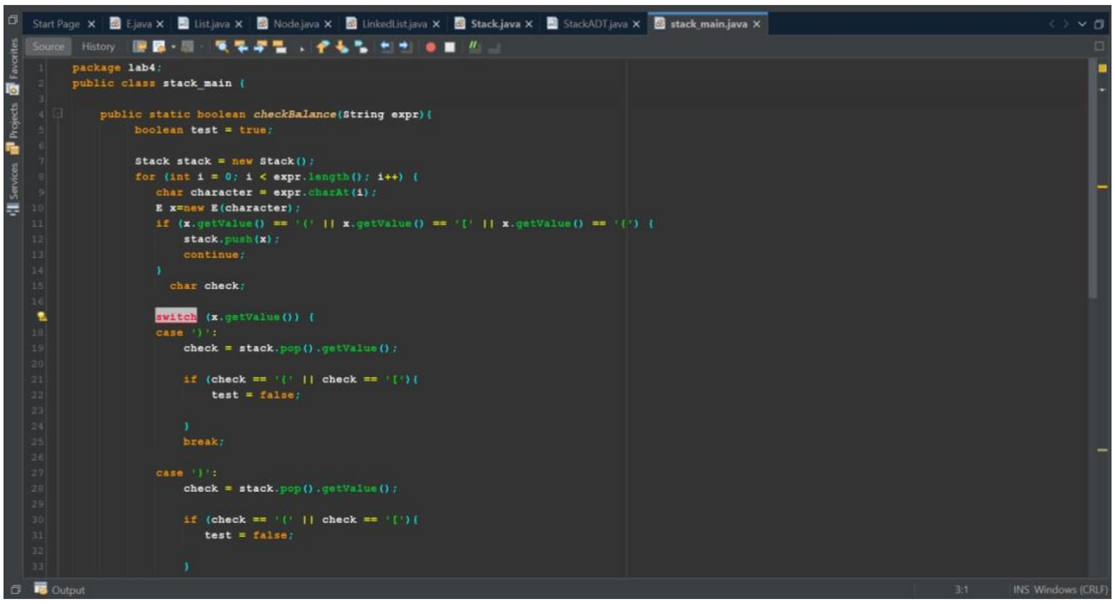
```
package lab4;
public interface StackADT<E>{
    void push(E x);
    E pop();
    E top();
    boolean isEmpty();
}
```

Stack class:

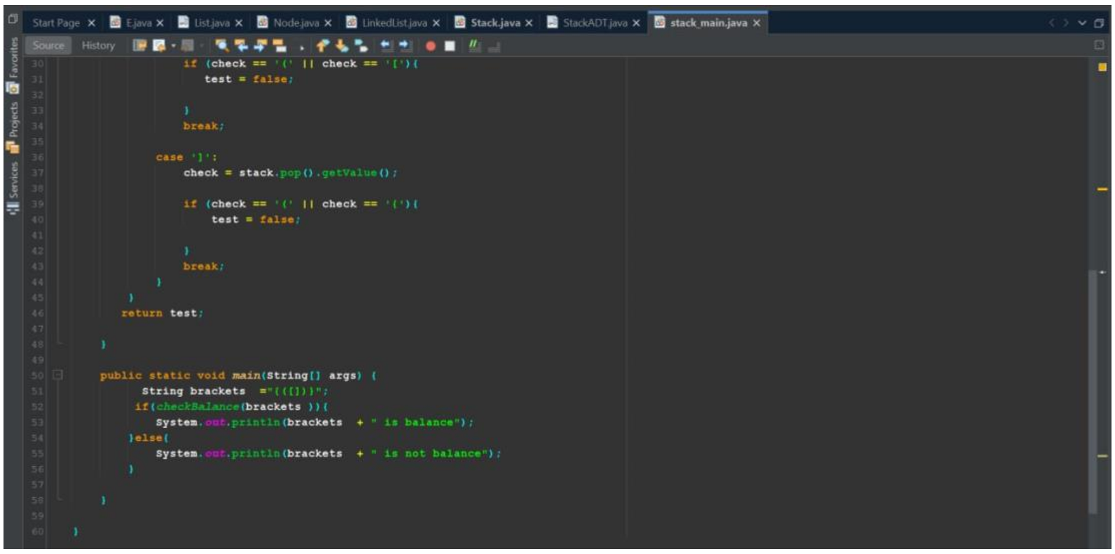


```java
package lab4;
class Stack implements StackADT<E> {
    private LinkedList list=new LinkedList();
    @Override
    public void push(E x) {
         list.addFront((E)x);
    }
    @Override
    public E pop() {
        if(list.empty())
        {
            System.out.println("stack is empty");
            return null;
        }else{
            return (E) list.removeFront();
        }
    }
    @Override
    public E top() {
        if (!isEmpty()){
             return (E) list.getHead();
        }else{
            return null;
        }
    }
    @Override
    public boolean isEmpty() {
        if(list.getHead() == null){
            return true;
        }else {
            return false;
        }
    }
```

Stack_main class:

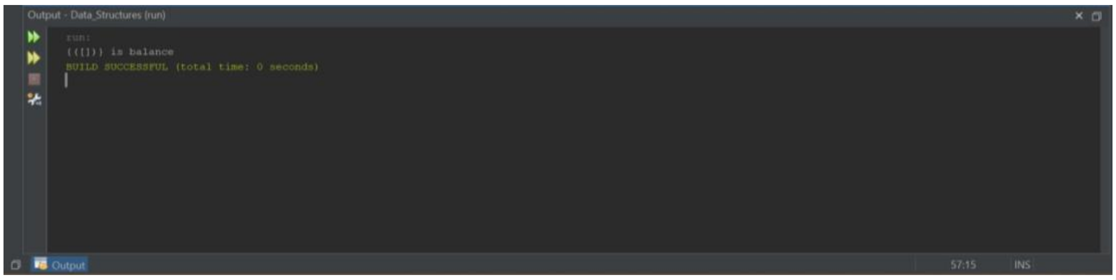


```java
package lab4;
public class stack_main {

    public static boolean checkBalance(String expr){
         boolean test = true;

         Stack stack = new Stack();
         for (int i = 0; i < expr.length(); i++) {
            char character = expr.charAt(i);
            E x=new E(character);
            if (x.getValue() == '(' || x.getValue() == '[' || x.getValue() == '{') {
                stack.push(x);
                continue;
            }
              char check;

            switch (x.getValue()) {
            case ')':
                check = stack.pop().getValue();

                if (check == '(' || check == '['){
                    test = false;

                }
                break;

            case '}':
                check = stack.pop().getValue();

                if (check == '(' || check == '['){
                   test = false;

                }
```

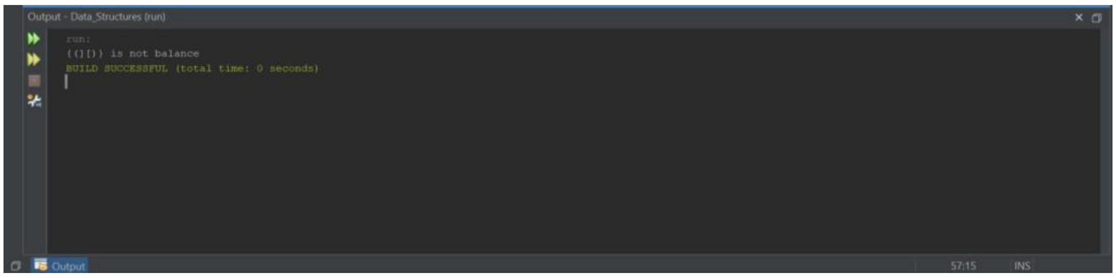


```
                    if (check == '{' || check == '['){
                        test = false;

                    }
                    break;

                case ']':
                    check = stack.pop().getValue();

                    if (check == '(' || check == '{'){
                         test = false;

                    }
                    break;
                }
            }
            return test;

        }

        public static void main(String[] args) {
              String brackets  ="{([])}";
             if(checkBalance(brackets )){
                System.out.println(brackets  + " is balance");
            }else{
                System.out.println(brackets  + " is not balance");
            }

        }

    }
```

The output:



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



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