```java
import java.util.Stack;

public class FullBalancedSymbolsChecker {

   public static boolean isBalanced(String str) {
       Stack<Character> stack = new Stack<>();

       for (char ch : str.toCharArray()) {
           // Handle opening brackets
           if (ch == '(' || ch == '{' || ch == '[' || ch == '<') {
               stack.push(ch);
           }
           // Handle closing brackets
           else if (ch == ')') {
               if (stack.isEmpty() || stack.pop() != '(') return false;
           } else if (ch == '}') {
               if (stack.isEmpty() || stack.pop() != '{') return false;
           } else if (ch == ']') {
               if (stack.isEmpty() || stack.pop() != '[') return false;
           } else if (ch == '>') {
               if (stack.isEmpty() || stack.pop() != '<') return false;
           }

           // Handle quotes (both ' and ")
           else if (ch == '\'' || ch == '"') {
               if (!stack.isEmpty() && stack.peek() == ch) {
                   stack.pop(); // Closing quote
               } else {
                   stack.push(ch); // Opening quote
               }
           }
       }

       // In the end, stack should be empty if everything matched
       return stack.isEmpty();
   }

   public static void main(String[] args) {

       System.out.println(isBalanced("{ [ ( ) ] }"));
       System.out.println(isBalanced("<< { \" [ ( ) ] } >>"));
       System.out.println(isBalanced("'<{[()]}'\">"));
       System.out.println(isBalanced("\"'([{}])'\""));


       System.out.println(isBalanced("{ [ ( ] ) }"));            // false
       System.out.println(isBalanced("\" [ { ] }"));             // false
       System.out.println(isBalanced("<< { ' ( [ ) } >>"));      // false
   }
```

```
}
```

```
C:\Users\Karthik\.jdks\openjdk-24.0.1\bin\java.e
true
false
false
true
false
false
false
```

```
import java.util.Stack;
public class BrowserNavigator {
   private Stack<String> backwardStack = new Stack<>();
   private Stack<String> forwardStack = new Stack<>();
   private String currentPage = "Home"; // starting page

   public void visit(String url) {
       System.out.println("Navigating to a new webpage: " + url);
       if (currentPage != null) {
           backwardStack.push(currentPage);
       }
       currentPage = url;
       forwardStack.clear();
       displayCurrentPage();
   }

   public void backward() {
       if (backwardStack.isEmpty()) {
           System.out.println("Cannot go back. No previous pages.");
           return;
       }
       System.out.println("Going back one page.");
       forwardStack.push(currentPage);
       currentPage = backwardStack.pop();
       displayCurrentPage();
   }

   public void forward() {
       if (forwardStack.isEmpty()) {
           System.out.println("Cannot go forward. No forward pages.");
           return;
       }
       System.out.println("Going forward one page.");
       backwardStack.push(currentPage);
```

```
        currentPage = forwardStack.pop();
        displayCurrentPage();
    }

    public void displayCurrentPage() {
        System.out.println("Current page: " + currentPage);
    } }
class BrowserDriver {
    public static void main(String[] args) {
        BrowserNavigator browser = new BrowserNavigator();

        browser.displayCurrentPage();  // Home
        browser.visit("https://example.com");
        browser.visit("https://openai.com");
        browser.visit("https://github.com");

        browser.backward(); // GitHub → OpenAI
        browser.backward(); // OpenAI → Example
        browser.forward();  // Example → OpenAI
        browser.visit("https://stackoverflow.com"); // clears forward stack

        browser.backward(); // StackOverflow → OpenAI
        browser.backward(); // OpenAI → Example
        browser.forward();  // Example → OpenAI
    }
}
```

```
C:\Users\Karthik\.jdks\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Program Fi
Current page: Home
Navigating to a new webpage: https://example.com
Current page: https://example.com
Navigating to a new webpage: https://openai.com
Current page: https://openai.com
Navigating to a new webpage: https://github.com
Current page: https://github.com
Going back one page.
Current page: https://openai.com
Going back one page.
Current page: https://example.com
Going forward one page.
Current page: https://openai.com
Navigating to a new webpage: https://stackoverflow.com
Current page: https://stackoverflow.com
Going back one page.
Current page: https://openai.com
Going back one page.
```