

```
public static String removeDoubleCharacters(String str) {  
    if (str.isEmpty()) {  
        return str;  
    }  
    StringBuilder result = new StringBuilder();  
    char prev = str.charAt(0);  
    result.append(prev);  
    for (int i = 1; i < str.length(); i++) {  
        char cur = str.charAt(i);  
        if (prev != cur) {  
            result.append(str.charAt(i));  
        }  
        prev = cur;  
    }  
    return result.toString();  
}
```

The diagram illustrates the control flow of the provided Java code. Nodes are represented by colored circles: purple for the initial state and the 'if' block, red for the 'return str;' statement, yellow for the 'StringBuilder result' initialization and the 'return result.toString();' statement, orange for the loop body, and grey for the 'if (str.isEmpty())' condition. Lines connect these nodes to show the sequence of execution: from the start to the 'if' block, then to the 'return str;' statement, then to the 'StringBuilder result' initialization, followed by the loop body, and finally to the 'return result.toString();' statement.