

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

Folder src/main

7 printable files

(file list disabled)

src/main/java/org/example/INumberleModel.java

```
1 package org.example;
2
3
4 public interface INumberleModel{
5     int Attempt = 6;
6
7
8     void StartGame();
9     boolean processInput(String input);
10    boolean isGameOver();
11    boolean isGameWon();
12
13    void setFlag3(boolean flag3);
14
15    boolean input(String input);
16    void setFlag(boolean flag1);
17    void setTargetword(String targetword);
18    String getTargetWord();
19    StringBuilder getCurrentGuess();
20    int getRemainingAttempts();
21    void setRemainingAttempts(int val);
22    boolean startNewGame();
23
24
25    void Delete();
26 }
27
28
29
```

译 极速翻译 检测到当前网页不是中文网页，是否要翻译成中文？

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

src/main/java/org/example/Junit.java

```
1 package org.example;
2 import org.junit.After;
3 import org.junit.Assert;
4 import org.junit.Test;
5
6 import org.junit.Before;
7
8 import java.io.ByteArrayInputStream;
9 import java.io.InputStream;
10
11
12 import static org.junit.Assert.*;
13
14
15 public class Junit {
16     private NumberleModel model;
17     private NumberleView view;
18     @Before
19     public void Setup(){
20         model=new NumberleModel();
21     }
22     @Test
23     public void TestNumber(){
24         model.setTargetword("2*4=4*2");
25         model.input("2*4=4*2");
26         assertEquals("2*4=4*2",model.getTargetWord());
27         assertTrue(model.Gamewon);
28
29     }
30     @Test
31     public void ProcessInput(){
32         model.StartGame();
33         assertFalse(model.processInput(""));
34         assertFalse(model.processInput("3*2=2*8"));
35         assertFalse(model.isGameWon());
36     }
37 }
```

译 极速翻译 检测到当前网页不是中文网页，是否要翻译成中文？

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
39     model.StartGame();
40     assertEquals(0, model.getCurrentGuess().length());
41 }
42
43 @Test
44 public void StartNewGame() {
45     model.StartGame();
46     String oldTargetWord = model.getTargetWord();
47     model.startNewGame();
48     assertEquals(oldTargetWord, model.getTargetWord());
49 }
50
51
52
53 @Test
54 public void testProcessInput_InvalidEquation() {
55     model.setFlag3(true);
56     model.StartGame();
57     assertFalse(model.processInput("1+2=2+3"));
58     assertFalse(model.isGameWon());
59 }
60
61
62 }
63
64
```

src\main\java\org\example\Main.java

```
1 package org.example;
2
3 public class Main {
4     public static void main(String[] args) {
5         INumberleModel model = new NumberleModel();
6         NumberleController controller=new NumberleController(model);
7         NumberleView view = new NumberleView(model, controller);
8
9     }
```

译 极速翻译 检测到当前网页不是中文网页，是否要翻译成中文？

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

11 | }

src\main\java\org\example\NumberleCLI.java

```
1 package org.example;
2 import java.util.Scanner;
3 import net.objecthunter.exp4j.Expression;
4 import net.objecthunter.exp4j.ExpressionBuilder;
5
6
7
8 public class NumberleCLI {
9     private static final String GREEN = "\u001B[32m";
10    private static final String YELLOW = "\u001B[33m";
11
12    private static final String White = "\u001B[36m";
13    private static final String GRAY = "\u001B[90m";
14
15
16    private static final INumberleModel model = new NumberleModel();
17    private static final Scanner scanner = new Scanner(System.in);
18
19    private static void GameStart() {
20        while (!model.isGameOver()) {
21            GameStatue();
22            String input = InputInformation();
23            if (input.length()<7){
24                System.out.println("the input length is too short ");
25            }
26            boolean containsOperator = false; // Flag to check if an operator is present
27
28            for (char c : input.toCharArray()) {
29                if ("+-*/".indexOf(c) != -1) {
30                    containsOperator = true; // Set flag if an operator is found
31                    break;
32                }
33            }
34            boolean equal=input.contains("=");
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)[不再提示](#)

```
36     boolean containsMultiple=input.contains("+-*/*");
37     boolean containMultiples=input.contains("+-/");
38     if (!equal){
39
40         System.out.println("no equal please try again ");
41     } else if (containMultiples||containsMultiple) {
42         System.out.println("the Multiple operator");
43     }else if (!containsOperator){
44         System.out.println("at least one operator");
45     }
46     if (input.length()==7) {
47         if (correct(input)){
48             boolean correct = model.input(input);
49             ColoredInput(input, model.getTargetWord());
50             if (correct) {
51                 break;
52             }
53         }
54     }
55 }
56
57 }
58
59 if (model.isGameWon()) {
60     System.out.println("wow! you won the game ");
61 } else {
62     System.out.println("Game over ! your lose the game !");
63     System.out.println("The right answer is " + model.getTargetWord());
64 }
65 }
66
67 private static void printCategories() {
68
69     System.out.println("operator: + - * / =");
70     System.out.println("number: 0 1 2 3 4 5 6 7 8 9");
71
72 }
73
74 private static String InputInformation() {
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
77     }
78     private static void GameStatue() {
79         System.out.println("the target equation: " + model.getTargetWord());
80         System.out.println("you have : " + model.getRemainingAttempts()+" "+"chances");
81         System.out.println("your input guess is : " + model.getCurrentGuess());
82         printCategories();
83     }
84     private static void ColoredInput(String input, String target) {
85
86         StringBuilder SE = new StringBuilder();
87         for (int i = 0; i < input.length(); i++) {
88             char c = input.charAt(i);
89             if (i < target.length() && c == target.charAt(i)) {
90                 SE.append(GREEN).append(c);
91             } else if (target.contains(String.valueOf(c))) {
92                 SE.append(YELLOW).append(c);
93             } else {
94                 SE.append(GRAY).append(c);
95             }
96         }
97         SE.append(White);
98         System.out.println(SE.toString());
99     }
100    private static boolean correct(String input) {
101        String[] EQUAL = input.split("=");
102        if (EQUAL.length != 2) {
103            return false;
104        }
105        Expression expLeft = new ExpressionBuilder(EQUAL[0].trim()).build();
106        double leftResult = expLeft.evaluate();
107        Expression expRight = new ExpressionBuilder(EQUAL[1].trim()).build();
108        double rightResult = expRight.evaluate();
109        if (leftResult!=rightResult){
110            System.out.println("the left side is not equal right side ");
111        }
112        return Double.compare(leftResult, rightResult) == 0;
113    }
114    public static void main(String[] args) {
115        model.startNewGame();
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
118 }  
119
```

src/main/java/org/example/NumberleController.java

```
1 package org.example;  
2  
3 public class NumberleController {  
4  
5     private NumberleView view;  
6     private INumberleModel model;  
7  
8     public NumberleController(INumberleModel model) {  
9         this.model = model;  
10    }  
11    public int getRemainingAttempts() {  
12        return model.getRemainingAttempts();  
13    }  
14    public void startNewGame() {  
15        model.startNewGame();  
16    }  
17  
18  
19    public void SetFlag(Boolean Flag3){  
20        model.setFlag3(Flag3);  
21        if (Flag3==true){  
22            model.setTargetword("3*2+1=7");  
23        }  
24  
25    }  
26    public void setflag1(Boolean flag){  
27        model.setFlag(flag);;  
28        System.out.println("error,the equation is not valid,this will not count as one of tires. so you have "+"getRemainingAttempts()  
+" "+"chance ");  
29  
30  
31    }  
32    public void setFlag2(Boolean Flag){  
--
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
35         System.out.println(getTargetWord());
36     }
37
38 }
39 public void setView(NumberleView view) {
40     this.view = view;
41 }
42 public String getTargetWord() {
43     return model.getTargetWord();
44 }
45 public void setRemainingAttempts(int val) {
46     model.setRemainingAttempts(val);
47 }
48 public void processInput(String input) {
49     model.processInput(input);
50 }
51 public boolean isGameOver() {
52     return model.isGameOver();
53 }
54 public boolean isGameWon() {
55     return model.isGameWon();
56 }
57
58 public void setTargetWord(){
59
60 }
61 }
62
```

src/main/java/org/example/NumberleModel.java

```
1 package org.example;
2
3 import java.io.File;
4 import java.util.ArrayList;
5 import java.util.Random;
6 import java.util.Scanner;
7 import java.io.FileNotFoundException;
```

译 极速翻译 检测到当前网页不是中文网页，是否要翻译成中文？

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
9
10 public class NumberleModel extends java.util.Observable implements INumberleModel{
11
12     private String Word;
13     public ArrayList<String> List=new ArrayList<>();
14     public int remainAttempts=6 ;
15     public boolean Gamewon;
16     public boolean Flag1=false;
17     public boolean Flag2=false ;
18     public boolean Flag3;
19
20     public String Formal_World="3+4-2=5";
21
22     private StringBuilder CurrentGuess;
23
24
25     //Start new game you should know this function just from txt file to select the words
26     //which is random to select
27     public void StartGame(){
28
29         assert List != null;
30         this.FileRead();
31         Random random=new Random();
32         int index =random.nextInt(List.size());
33         if (Flag1==true){
34             System.out.println("this is not correct answer ");
35         }
36         if (Flag2==true){
37             System.out.println(getTargetWord());
38         }
39         if (Flag3==true){
40             Word=Formal_World;
41         }else {
42             Word=List.get(index);
43         }
44         CurrentGuess=new StringBuilder(Word.length());
45         remainAttempts=Attempt;
46         Gamewon=false;
47         setChanged();
48     }
49 }
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

翻译成其他语言

不再提示

```
50 public boolean ProcessInput(String input){
51
52     assert input.length() <=7 : "Input length should be <= 7";
53     if (input.length() < 7){
54         return false;
55     }
56     CurrentGuess = new StringBuilder(input);
57
58     if (CurrentGuess.toString().equals(Word)) {
59         Gamewon = true;
60     }
61     setChanged();
62     notifyObservers();
63     return Gamewon;
64 }
65
66 public boolean input(String input){
67     assert input.length() <=7 : "Input length should be <= 7";
68     if (input.length() < 7){
69         System.out.println("input length must =7");
70     }
71     CurrentGuess = new StringBuilder(input);
72
73
74     if (CurrentGuess.toString().equals(Word)) {
75         Gamewon = true;
76     }
77     else {
78         remainAttempts--;
79     }
80     return Gamewon;
81 }
82
83
84
85
86
87 public void FileRead(){
88     try {
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
91         Scanner scanner =new Scanner(file);
92
93         while(scanner.hasNextLine()){
94             this.List.add(scanner.nextLine());
95
96         }
97         scanner.close();
98
99     } catch (FileNotFoundException e) {
100         System.out.println("File can not read,please try to fix it ");
101     }
102
103 }
104
105
106 @Override
107 public boolean processInput(String input) {
108     return ProcessInput(input);
109 }
110
111 @Override
112 public boolean isGameOver() {
113     assert remainAttempts >= 0 : "remainAttempts should be non-negative";
114     return remainAttempts <= 0 || Gamewon ;
115 }
116
117 @Override
118 public boolean isGameWon() {
119     return Gamewon;
120 }
121
122
123
124 @Override
125 public void setFlag(boolean flag2) {
126     this.Flag2=flag2;
127 }
128 @Override
129 public void setFlag3(boolean flag3) {
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
132
133
134
135
136 @Override
137 public void setTargetword(String targetword) {
138     assert targetword != null : "targetword should not be null";
139     assert targetword.length() >= 7 : "targetword length should be >= 7";
140     this.Word=targetword;
141 }
142
143 @Override
144 public String getTargetWord() {
145     return Word;
146 }
147
148 @Override
149 public StringBuilder getCurrentGuess() {
150     return CurrentGuess;
151 }
152
153 @Override
154 public int getRemainingAttempts() {
155     assert remainAttempts >= 0 : "remainAttempts should be non-negative";
156     return remainAttempts;
157 }
158
159 @Override
160 public void setRemainingAttempts(int val) {
161     assert val >= 0 : "val should be non-negative";
162     remainAttempts = val;
163
164 }
165
166
167 @Override
168 public boolean startNewGame() {
169     StartGame();
170     return false;
171 }
```

译 极速翻译 检测到当前网页不是中文网页，是否要翻译成中文？

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
173
174
175
176     @Override
177     public void Delete() {
178
179         if (CurrentGuess.length()>0){
180             CurrentGuess.deleteCharAt(CurrentGuess.length()-1);
181             setChanged();
182             notifyObservers();
183         }
184     }
185
186
187 }
188
```

src\main\java\org\example\Number1eView.java

```
1 package org.example;
2
3
4 import javax.swing.*;
5
6 import java.awt.*;
7 import java.awt.event.ActionEvent;
8 import java.awt.event.ActionListener;
9
10 import java.util.Observable;
11 import java.util.Observer;
12
13
14 import net.objecthunter.exp4j.Expression;
15 import net.objecthunter.exp4j.ExpressionBuilder;
16
17
18
19
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
21 private JPanel Panel;//game panel
22 private JLayeredPane layeredPane;
23 private NumberleController controller;// Controller
24 private final INumberleModel model;// Model
25 public JButton button[][]=new JButton[6][7]; // Game buttons
26 private JTextField inputTextField;
27 private int Row=0;
28
29 public static final int ROWS = 6;// Total rows
30 public static final int COLS = 7;// Total columns
31 public static final int MAX_INPUT_LENGTH = 7;// Maximum length of input text field
32 private static final Color CORRECT_COLOR = Color.GREEN; // Correct color
33 //The entered text contains letters but in the wrong order
34 private static final Color INCORRECT_COLOR = Color.YELLOW;
35 private final Color defaultColor = UIManager.getColor("Button.background");// Default button color
36 //
37 private JPanel row1;// First row panel
38 private JPanel row2;// Second row panel
39
40
41
42 public NumberleView(INumberleModel model, NumberleController controller){
43     // Parameter validation
44     assert model != null; // Ensure model parameter is not null
45     assert controller != null; // Ensure controller parameter is not null
46
47     // Assigning controller and model
48     this.controller = controller; // Assign controller parameter to the controller field
49     this.model = model; // Assign model parameter to the model field
50
51     // Start new game
52     controller.startNewGame(); // Invoke startNewGame method of the controller
53
54     // Initialize GUI components
55     initialize(); // Call the initialize method to set up the GUI components
56
57     // Show game start reminder
58     showGameStartReminder(); // Display a reminder or message indicating the game has started
59
60     // ...
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
62
63     // Set view for controller
64     this.controller.setView(this); // Set this view instance as the view for the controller
65
66     // Update view with initial model state
67     update((Number1eModel)this.model, null); // Call update method to synchronize view with initial model state
68 }
69 public void initialize() {
70
71     // Set the look and feel of the user interface to MetalLookAndFeel
72     try {
73         UIManager.setLookAndFeel("com.sun.java.swing.plaf.motif.MotifLookAndFeel");
74     } catch (Exception e) {
75         e.printStackTrace();
76     }
77
78     // Set the title of the frame
79     setTitle("Game");
80
81     // Set default close operation to exit the application when the frame is closed
82     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
83
84     // Set the frame size to 800x800 pixels
85     setSize(750, 750);
86     setLocationRelativeTo(null);
87     // Make the frame non-resizable
88     setResizable(false);
89
90     // Set the layout of the frame to BorderLayout
91     setLayout(new BorderLayout());
92
93     // Create a panel with a 6x7 grid layout for buttons
94     Panel = new JPanel(new GridLayout(6, 7));
95
96     // Create buttons and add them to the panel
97     for (int i = 0; i < 6; i++) {
98         for (int j = 0; j < 7; j++) {
99             button[i][j] = new JButton("");
100             button[i][j].setPreferredSize(new Dimension(100, 100));
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
103         // Add the panel to the frame's center position
104         add(Panel, BorderLayout.CENTER);
105         Panel.setVisible(true);
106     }
107
108     // Create an input panel with a text field
109     JPanel inputPanel = new JPanel();
110     inputTextField = new JTextField(7);
111     inputPanel.add(inputTextField);
112     add(inputPanel, BorderLayout.NORTH);
113     inputPanel.setVisible(false);
114
115     // Get text from input text field
116     String input = inputTextField.getText();
117
118     // Assert that input text field is not null or empty
119     assert (input) != null & !input.isEmpty();
120
121     // Create a keyboard panel with numeric buttons and operator buttons
122     JPanel keyBoardPanel = new JPanel();
123     keyBoardPanel.setLayout(new BoxLayout(keyBoardPanel, BoxLayout.Y_AXIS));
124     row1 = new JPanel();
125     row1.setLayout(new FlowLayout());
126     for (int i = 1; i <= 9; i++) {
127         JButton button = new JButton(Integer.toString(i));
128         button.setFont(new Font("Serif", Font.PLAIN, 30));
129         button.addActionListener(new ButtonClickListener());
130         row1.add(button);
131     }
132     JButton zeroButton = new JButton("0");
133     zeroButton.setFont(new Font("Serif", Font.PLAIN, 30));
134     zeroButton.addActionListener(new ButtonClickListener());
135     row1.add(zeroButton);
136
137     row2 = new JPanel();
138     row2.setLayout(new FlowLayout());
139     String[] operators = {"Remove", "+", "-", "*", "/", "=", "Enter"};
140     for (String operator : operators) {
141         JButton button = new JButton(operator);
142         button.setFont(new Font("Serif", Font.PLAIN, 30));
143         button.addActionListener(new ButtonClickListener());
144         row2.add(button);
145     }
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
144         row2.add(button);
145     }
146     keyBoardPanel.add(row1);
147     keyBoardPanel.add(row2);
148
149     // Add the keyboard panel to the frame's south position
150     add(keyBoardPanel, BorderLayout.SOUTH);
151
152     // Create a button panel with various control buttons
153     JPanel ButtonPanel = new JPanel();
154     JButton Information = new JButton("How to play");
155
156     JButton FixFormual = new JButton("Fixed_Formula");
157     JButton display=new JButton("Display");
158
159
160     display.addActionListener(e -> {
161         display();
162     });
163
164     // Add action listeners to control buttons
165     FixFormual.addActionListener(e -> {
166         Fixed();
167     });
168
169     Information.addActionListener(e -> {
170         HowtoPlayGame();
171     });
172
173
174
175     // Add buttons to the button panel
176     ButtonPanel.add(Information);
177     ButtonPanel.add(FixFormual);
178     ButtonPanel.add(display);
179
180
181     // Add the button panel to the frame's north position
182     add(ButtonPanel, BorderLayout.NORTH);
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
185     setVisible(true);
186
187     // Call the change method
188     change();
189 }
190 public void Fixed() {
191     controller.SetFlag(true);
192     System.out.println(controller.getTargetWord());
193 }
194
195 /**
196  * Removes all components from row2 panel and replaces them with operator buttons in a specific order.
197  * This method is called when the "Swap" button is clicked.
198  */
199
200 public void display(){
201     controller.setFlag2(true);
202 }
203 /**
204  * Updates the buttons based on the input text.
205  * This method is called when there is a change in the input text field.
206  */
207 public void change() {
208     String input = inputTextField.getText();
209     updateButtonsFromInput(input);
210 }
211
212
213
214
215
216 /**
217  * ActionListener implementation for handling button clicks.
218  */
219 private class ButtonClickListener implements ActionListener {
220     @Override
221     public void actionPerformed(ActionEvent e) {
222         JButton source = (JButton) e.getSource();
223         String buttonText = source.getText();
```

译 极速翻译 检测到当前网页不是中文网页，是否要翻译成中文？

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
226         clearDisplay();
227         break;
228     case "Enter":
229         Enter();
230
231         break;
232     default:
233         if (inputTextField.getText().length() < MAX_INPUT_LENGTH) {
234             updateDisplay(buttonText);
235             updateButtonsFromInput(inputTextField.getText());
236         }
237         controller.isGameOver();
238     }
239 }
240 }
241 /**
242  * Updates the color and text of buttons based on the target word and user input.
243  * @param target The target word as an array of characters.
244  * @param input The user input as a string.
245  */
246 private void updateButtonsFromTarget(char[] target, String input) {
247     char[] userInput = input.toCharArray();
248
249     for (int col = 0; col < 7; col++) {
250
251         char targetChar = target[col];
252         char inputChar = userInput[col];
253
254         if (targetChar == inputChar) {
255             button[Row][col].setBackground(Color.GREEN);
256         } else if (new String(target).indexOf(inputChar) != -1) {
257             button[Row][col].setBackground(Color.ORANGE);
258         } else {
259             button[Row][col].setBackground(Color.GRAY);
260         }
261
262         button[Row][col].setText(String.valueOf(inputChar));
263
264     }
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

翻译成其他语言

不再提示

```
267
268     for (Component component : row1.getComponents()) {
269         if (component instanceof JButton) {
270             JButton keyboardButton = (JButton) component;
271             if (keyboardButton.getText().equals(buttonText)) {
272                 if (targetChar == inputChar) {
273                     keyboardButton.setBackground(Color.GREEN);
274                 } else if (new String(target).indexOf(inputChar) != -1) {
275                     keyboardButton.setBackground(Color.ORANGE);
276                 } else {
277                     keyboardButton.setBackground(Color.GRAY);
278                 }
279                 break;
280             }
281         }
282     }
283
284
285     for (Component component : row2.getComponents()) {
286         if (component instanceof JButton) {
287             JButton keyboardButton = (JButton) component;
288             if (keyboardButton.getText().equals(buttonText)) {
289                 if (targetChar == inputChar) {
290                     keyboardButton.setBackground(Color.GREEN);
291                 } else if (new String(target).indexOf(inputChar) != -1) {
292                     keyboardButton.setBackground(Color.ORANGE);
293                 } else {
294                     keyboardButton.setBackground(Color.GRAY);
295                 }
296                 break;
297             }
298         }
299     }
300 }
301
302
303
304 /**
305  * Updates the text of buttons based on the user input.
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
308 private void updateButtonsFromInput(String input) {
309     int rowIndex = Row;
310
311     int colIndex = 0;
312     char[] chars = input.toCharArray();
313
314     for (char c : chars) {
315         if (Character.isDigit(c) || "+-*/=".indexOf(c) != -1) {
316             if (colIndex < 7) {
317                 button[rowIndex][colIndex].setText(String.valueOf(c));
318                 colIndex++;
319
320             }
321         }
322     }
323     for (int clearIndex = colIndex; clearIndex < 7; clearIndex++) {
324         button[rowIndex][clearIndex].setText("");
325     }
326
327 }
328
329 /**
330  * Handles the action when the "Enter" button is clicked.
331  */
332 public void Enter() {
333     String input = inputTextField.getText();
334     // Check if input length is less than 7
335     if (input.length() < 7) {
336         controller.setflag1(true);
337         TooShort();
338         return; // Return to avoid further processing
339     }
340
341     boolean containsOperator = false; // Flag to check if an operator is present
342
343     for (char c : input.toCharArray()) {
344         if ("+-*/".indexOf(c) != -1) {
345             containsOperator = true; // Set flag if an operator is found
346             break;
347         }
348     }
349 }
```

译 极速翻译 检测到当前网页不是中文网页，是否要翻译成中文？

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
349
350
351
352
353
354     if (input.length() == 7) {
355         boolean containsEquals = input.contains("=");
356         boolean containsMultiple=input.contains("+-*");
357         boolean containMultiples=input.contains("+-/");
358         boolean containMultiplless=input.contains("+-*");
359         boolean contains=input.contains("+=++");
360
361         if (!containsEquals) {
362             NoEqualSign();
363         } else if (!containsOperator) {
364             Symbol();
365         } else if (containsMultiple||containMultiples||containMultiplless){
366             MultiplyOperator();
367
368         } else if ( contains){
369             MultiplyOperator();
370         }
371         else {
372             boolean isCorrect = validateEquation(input);
373             if (isCorrect) {
374                 char[]target=controller.getTargetWord().toCharArray();
375                 validateEquation(input);
376
377                 controller.setRemainingAttempts(controller.getRemainingAttempts()-input.length()/7);
378
379                 updateButtonsFromInput(input); // Update buttons based on the input4
380                 updateButtonsFromTarget(target,input);
381                 Row++;
382
383                 if (Row>5){
384                     Row=0;
385                 }
386                 controller.processInput(input);
387
388                 ...

```

译 极速翻译 检测到当前网页不是中文网页，是否要翻译成中文？

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
390         Youlost();
391     }
392 }
393 else{
394
395     NotEqual();
396 }
397 }
398 }
399
400 }
401
402
403
404
405
406
407 /**
408  * Validates the equation provided by the user.
409  *
410  * @param input The equation input by the user.
411  * @return True if the equation is valid, false otherwise.
412  */
413 private boolean validateEquation(String input) {
414     String[] parts = input.split("="); // Split the input into left and right sides of the equation
415     if (parts.length != 2) {
416         return false; // Return false if the equation doesn't contain exactly one equals sign
417     }
418
419     // Evaluate the left side of the equation
420     Expression expLeft = new ExpressionBuilder(parts[0].trim()).build();
421     double leftResult = expLeft.evaluate();
422
423     // Evaluate the right side of the equation
424     Expression expRight = new ExpressionBuilder(parts[1].trim()).build();
425     double rightResult = expRight.evaluate();
426
427     // Compare the results of the left and right sides of the equation
428     return Double.compare(leftResult, rightResult) == 0;
429 }
```

译 极速翻译 检测到当前网页不是中文网页，是否要翻译成中文？

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
431
432
433
434
435 /**
436  * Clears the display by removing the last character from the input text field.
437  */
438 public void clearDisplay() {
439     String currentText = inputTextField.getText();
440     if (!currentText.isEmpty()) {
441         // Remove the last character from the current text
442         String newText = currentText.substring(0, currentText.length() - 1);
443         inputTextField.setText(newText);
444         // Update buttons based on the new input text
445         updateButtonsFromInput(newText);
446     }
447 }
448
449
450 /**
451  * Updates the display by appending the given text to the input text field.
452  * If the length of the resulting text exceeds 7 characters, it truncates the text appropriately.
453  *
454  * @param text The text to be added to the input text field.
455  */
456 public void updateDisplay(String text) {
457     // Check if adding the text will exceed the maximum length
458     if (inputTextField.getText().length() + text.length() > 7) {
459         // Truncate the text to fit within the maximum length
460         text = text.substring(0, 7 - inputTextField.getText().length());
461     }
462     // Append the text to the input text field
463     inputTextField.setText(inputTextField.getText() + text);
464 }
465
466
467
468
469 /**
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
472     */
473     public void TooShort() {
474         // Create a reminder frame
475         JFrame reminderFrame = new JFrame("reminder");
476         reminderFrame.setBounds(625,300, 400, 200);
477         reminderFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
478
479         // Create a label for the reminder message
480         JLabel reminderLabel = new JLabel("Too Short !");
481         reminderLabel.setFont(new Font("Serif", Font.BOLD, 25));
482         reminderLabel.setHorizontalAlignment(JLabel.CENTER);
483
484         // Create a panel to hold the reminder label
485         JPanel reminderPanel = new JPanel(new BorderLayout());
486         reminderPanel.add(reminderLabel, BorderLayout.CENTER);
487         reminderFrame.getContentPane().add(reminderPanel);
488
489         // Make the reminder frame visible
490         reminderFrame.setVisible(true);
491
492         // Set a timer to automatically close the reminder frame after 2 seconds
493         Timer timer = new Timer(2000, e -> {
494             reminderFrame.dispose();
495         });
496         timer.setRepeats(false);
497         timer.start();
498     }
499
500     /**
501     * Displays a reminder frame indicating that there must be at least one mathematical symbol
502     * +- */ The reminder frame is displayed for 2 seconds before automatically closing.
503     *
504     */
505
506     public void MultiplyOperator() {
507         // Create a reminder frame
508         JFrame reminderFrame = new JFrame("reminder");
509         reminderFrame.setBounds(625,300, 400, 200);
510         reminderFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
511     }
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
513     JLabel reminderLabel = new JLabel("Multiple math symbols in a row ");
514     reminderLabel.setFont(new Font("Serif", Font.BOLD, 25));
515     reminderLabel.setHorizontalAlignment(JLabel.CENTER);
516
517     // Create a panel to hold the reminder label
518     JPanel reminderPanel = new JPanel(new BorderLayout());
519     reminderPanel.add(reminderLabel, BorderLayout.CENTER);
520     reminderFrame.getContentPane().add(reminderPanel);
521
522     // Make the reminder frame visible
523     reminderFrame.setVisible(true);
524
525     // Set a timer to automatically close the reminder frame after 2 seconds
526     Timer timer = new Timer(2000, e -> {
527         reminderFrame.dispose();
528     });
529     timer.setRepeats(false);
530     timer.start();
531 }
532 /**
533  * Displays a reminder frame indicating that there
534  * they must be at least one sign
535  *
536  *
537  */
538 public void Symbol() {
539     // Create a reminder frame
540     JFrame reminderFrame = new JFrame("reminder");
541     reminderFrame.setBounds(625, 300, 400, 200);
542     reminderFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
543
544     // Create a label for the reminder message
545     JLabel reminderLabel = new JLabel("There must be at least one sign +-*/ !");
546     reminderLabel.setFont(new Font("Serif", Font.BOLD, 25));
547     reminderLabel.setHorizontalAlignment(JLabel.CENTER);
548
549     // Create a panel to hold the reminder label
550     JPanel reminderPanel = new JPanel(new BorderLayout());
551     reminderPanel.add(reminderLabel, BorderLayout.CENTER);
552     reminderFrame.getContentPane().add(reminderPanel);
553 }
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

翻译成其他语言

不再提示

```
554 // Make the reminder frame visible
555 reminderFrame.setVisible(true);
556
557 // Set a timer to automatically close the reminder frame after 2 seconds
558 Timer timer = new Timer(2000, e -> {
559     reminderFrame.dispose();
560 });
561 timer.setRepeats(false);
562 timer.start();
563 }
564
565 /**
566  * Displays a reminder frame explaining how to play the game.
567  * The reminder frame contains instructions on how to guess the hidden mathematical equation in 6 tries
568  * and how the color of the tiles changes to show the proximity to the correct solution.
569  * The reminder frame is displayed until closed by the user.
570  */
571 public void HowtoPlayGame() {
572     // Create a frame to display the instructions
573     JFrame reminderFrame = new JFrame("How to play");
574     reminderFrame.setBounds(420, 150, 850, 850);
575     reminderFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
576
577     // Create a text area to display the instructions
578     JTextArea textArea = new JTextArea();
579     textArea.setLineWrap(true); // Enable automatic line wrapping
580     textArea.setWrapStyleWord(true);
581     textArea.setText("You have to guess the hidden math equation in 6 tries and color of the tiles " +
582         "To start playing, enter any mathematical equation. For example: 5+7*5=40.\n" + "if you enter the 5+5*2=10" +
583         "the first part and second part ,four part,seven parts will become green because your position is correct "
584
585     );
586
587     // Set font and size for the text area
588     textArea.setFont(new Font("Serif", Font.ROMAN_BASELINE, 30));
589
590     // Create a panel to hold the text area
591     JPanel reminderPanel = new JPanel(new BorderLayout());
592     reminderPanel.add(textArea, BorderLayout.CENTER);
593     ...
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
595     // Make the reminder frame visible
596     reminderFrame.setVisible(true);
597 }
598
599
600 /**
601  * Displays a reminder frame indicating that the left side of the equation is not equal to the right side.
602  * Displays a reminder frame indicating that the left side of the equation is not equal to the right side.
603  */
604 public void NotEqual() {
605     // Create a frame to display the reminder
606     JFrame reminderFrame = new JFrame("reminder");
607     reminderFrame.setBounds(625,300, 500, 400);
608     reminderFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
609
610     // Create a label to display the reminder message
611     JLabel reminderLabel = new JLabel("The left side is not equal to the right");
612     reminderLabel.setFont(new Font("Serif", Font.BOLD, 25));
613     reminderLabel.setHorizontalAlignment(JLabel.CENTER);
614
615     // Create a panel to hold the label
616     JPanel reminderPanel = new JPanel(new BorderLayout());
617     reminderPanel.add(reminderLabel, BorderLayout.CENTER);
618     reminderFrame.getContentPane().add(reminderPanel);
619
620     // Make the reminder frame visible
621     reminderFrame.setVisible(true);
622
623     // Close the reminder frame after a delay
624     Timer timer = new Timer(2000, e -> {
625         reminderFrame.dispose();
626     });
627     timer.setRepeats(false);
628     timer.start();
629 }
630
631 /**
632  * Displays a reminder frame indicating the game start and prompts the user to guess the first equation.
633  * The reminder frame is displayed until closed by the user.
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)[不再提示](#)

```
636 // Create a frame to display the reminder
637 JFrame reminderFrame = new JFrame("reminder");
638 reminderFrame.setBounds(625, 300, 450, 300);
639 reminderFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
640
641 // Create a label to display the reminder message
642 JLabel reminderLabel = new JLabel("Guess the first equation!");
643 reminderLabel.setFont(new Font("Serif", Font.BOLD, 25));
644 reminderLabel.setHorizontalAlignment(JLabel.CENTER);
645
646 // Create a panel to hold the label
647 JPanel reminderPanel = new JPanel(new BorderLayout());
648 reminderPanel.add(reminderLabel, BorderLayout.CENTER);
649 reminderFrame.getContentPane().add(reminderPanel);
650
651 // Make the reminder frame visible
652 reminderFrame.setVisible(true);
653
654 // Close the reminder frame after a delay
655 Timer timer = new Timer(2000, e -> {
656     reminderFrame.dispose();
657 });
658 timer.setRepeats(false);
659 timer.start();
660 }
661
662 /**
663  * Displays a reminder frame indicating that there is no equal (=) sign in the input.
664  * The reminder frame is displayed until closed by the user.
665  */
666 public void NoEqualSign() {
667     // Create a frame to display the reminder
668     JFrame reminderFrame = new JFrame("reminder");
669     reminderFrame.setBounds(625, 300, 400, 200);
670     reminderFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
671
672     // Create a label to display the reminder message
673     JLabel reminderLabel = new JLabel("No equal (=) sign!");
674     reminderLabel.setFont(new Font("Serif", Font.BOLD, 25));
675     reminderLabel.setHorizontalAlignment(JLabel.CENTER);
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
677 // Create a panel to hold the label
678 JPanel reminderPanel = new JPanel(new BorderLayout());
679 reminderPanel.add(reminderLabel, BorderLayout.CENTER);
680 reminderFrame.getContentPane().add(reminderPanel);
681
682 // Make the reminder frame visible
683 reminderFrame.setVisible(true);
684
685 // Close the reminder frame after a delay
686 Timer timer = new Timer(2000, e -> {
687     reminderFrame.dispose();
688 });
689 timer.setRepeats(false);
690 timer.start();
691 }
692
693 /**
694  * Displays a reminder frame indicating that the player has lost the game.
695  * Provides an option to replay the game.
696  */
697 public void Youlost() {
698     // Create a frame to display the reminder
699     JFrame reminderFrame = new JFrame("reminder");
700     reminderFrame.setBounds(625, 300, 400, 200);
701     reminderFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
702
703     // Create a button for replaying the game
704     Button Replay = new Button("Replay");
705     Replay.addActionListener(e -> {
706         controller.startNewGame();
707         System.out.println("you replay the game so the number become "+ " "+ controller.getTargetWord());
708         for (int row = 0; row < ROWS; row++) {
709             for (int col = 0; col < COLS; col++) {
710                 button[row][col].setText("");
711                 button[row][col].setBackground(defaultColor);
712             }
713         }
714
715         // Enable buttons for input
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)[不再提示](#)

```
718         component.setEnabled(true);
719         component.setBackground(defaultColor);
720     }
721 }
722 // Enable operator buttons
723 for (Component component : row2.getComponents()) {
724     if (component instanceof JButton) {
725         component.setEnabled(true);
726         component.setBackground(defaultColor);
727     }
728 }
729 // Close the reminder frame
730 reminderFrame.dispose();
731 });
732 Replay.setBounds(10, 20, 10, 10);
733
734 // Create a label to display the reminder message
735 JLabel reminderLabel = new JLabel("You lost!");
736 reminderLabel.setFont(new Font("Serif", Font.BOLD, 25));
737 reminderLabel.setHorizontalAlignment(JLabel.CENTER);
738
739 // Create a panel to hold the label and replay button
740 JPanel reminderPanel = new JPanel(new BorderLayout());
741 reminderPanel.add(reminderLabel, BorderLayout.CENTER);
742 reminderPanel.add(Replay, BorderLayout.NORTH);
743
744 // Add the panel to the reminder frame
745 reminderFrame.getContentPane().add(reminderPanel);
746
747 // Make the replay button and reminder frame visible
748 Replay.setVisible(true);
749 reminderFrame.setVisible(true);
750 }
751
752 /**
753  * Displays a reminder frame indicating that the player has won the game.
754  * Provides an option to replay the game.
755  */
756 public void Youwon() {
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
759     reminderFrame.setBounds(625, 300, 400, 200);
760     reminderFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
761
762     // Create a button for restarting the game
763     Button Restart = new Button("Replay");
764     Restart.addActionListener(e -> {
765         Row = 0;
766         // Reset the game board
767         for (int row = 0; row < ROWS; row++) {
768             for (int col = 0; col < COLS; col++) {
769                 button[row][col].setText("");
770                 button[row][col].setBackground(defaultColor);
771             }
772         }
773         // Enable buttons for input
774         for (Component component : row1.getComponents()) {
775             if (component instanceof JButton) {
776                 component.setEnabled(true);
777                 component.setBackground(defaultColor);
778             }
779         }
780         // Enable operator buttons
781         for (Component component : row2.getComponents()) {
782             if (component instanceof JButton) {
783                 component.setEnabled(true);
784                 component.setBackground(defaultColor);
785             }
786         }
787         // Close the reminder frame
788         reminderFrame.dispose();
789     });
790     Restart.setBounds(10, 20, 10, 10);
791
792     // Create a label to display the reminder message
793     JLabel reminderLabel = new JLabel("You won!");
794     reminderLabel.setFont(new Font("Serif", Font.BOLD, 25));
795     reminderLabel.setHorizontalAlignment(JLabel.CENTER);
796
797     // Create a panel to hold the label and restart button
```

译 极速翻译 检测到当前网页不是中文网页, 是否要翻译成中文?

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)

```
800     reminderPanel.add(Restart, BorderLayout.NORTH);
801
802     // Add the panel to the reminder frame
803     reminderFrame.getContentPane().add(reminderPanel);
804
805     // Make the restart button and reminder frame visible
806     Restart.setVisible(true);
807     reminderFrame.setVisible(true);
808 }
809
810 @Override
811 /**
812  * update the GUI ,If the view is change the observable will notice changed
813  *
814  */
815     public void update(Observable o, Object arg) {
816
817
818         if (controller.isGameOver()) {
819             if (controller.isGameWon()) {
820                 Youwon();
821                 controller.startNewGame();
822                 System.out.println("The new number is"+" " +controller.getTargetWord());
823
824             }
825
826             for (Component component : row1.getComponents()) {
827                 if (component instanceof JButton) {
828                     component.setEnabled(false);
829                 }
830             }
831             for (Component component : row2.getComponents()) {
832                 if (component instanceof JButton) {
833                     component.setEnabled(false);
834                 }
835             }
836         }
837     }
838 }
```

 极速翻译 检测到当前网页不是中文网页，是否要翻译成中文？

翻译

不翻译

[翻译成其他语言](#)

[不再提示](#)