

[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 import java.util.ArrayList;
4
5 public interface INumberleModel{
6     int MAX_ATTEMPTS = 6;
7
8
9     void initialize();
10    boolean processInput(String input);
11    boolean isGameOver();
12    boolean isGameWon();
13
14    void setFlag3(boolean flag3);
15
16
17    void setFlag(boolean flag1);
18    void setTargetword(String targetword);
19    String getTargetWord();
20    StringBuilder getCurrentGuess();
21    int getRemainingAttempts();
22    void setRemainingAttempts(int val);
23    boolean startNewGame();
24    String DefaultWord();
25
26    void remove( );
27
28 }
```

 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](#)

31
32

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

```
1 package org.example;
2 import org.junit.Assert;
3 import org.junit.Test;
4
5 import org.junit.Before;
6
7 import java.io.ByteArrayInputStream;
8 import java.io.InputStream;
9 import java.util.ArrayList;
10 import java.util.Scanner;
11
12 import static org.junit.Assert.*;
13 import static org.junit.Assert.*;
14
15 public class Junit {
16     private NumberleModel model;
17     @Before
18     public void Setup(){
19         model=new NumberleModel();
20     }
21
22     @Test
23     public void TestNumber(){
24         model.setTargetword("1+2=2+1");
25         assertEquals("1+2=2+1",model.getTargetWord());
26         Assert.assertNotNull(true);
27     }
28
29     @Test
30     public void testProcessInput(){
31         model.initialize();
32         assertFalse(model.processInput(""));
33         assertFalse(model.processInput(null));
```

[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     }
36     @Test
37     public void testRemove_EmptyGuess() {
38         model.initialize();
39         model.remove();
40         assertEquals(0, model.getCurrentGuess().length());
41     }
42     @Test
43     public void testStartNewGame() {
44         model.initialize();
45         String oldTargetWord = model.getTargetWord();
46         model.startNewGame();
47         assertNotEquals(oldTargetWord, model.getTargetWord());
48     }
49     @Test
50     public void testRemainingAttempts() {
51         model.initialize();
52         int initialAttempts = model.getRemainingAttempts();
53         model.setRemainingAttempts(initialAttempts - 1);
54         assertEquals(initialAttempts - 1, model.getRemainingAttempts()); // 测试剩余尝试次数
55     }
56     @Test
57     public void testProcessInput_InvalidEquation() {
58         model.setFlag3(true);
59         model.initialize();
60         assertFalse(model.processInput("invalid equation")); // 测试无效的等式输入
61         assertFalse(model.isGameWon());
62     }
63     @Test
64     public void testProcessInput_InvalidInput() {
65         String invalidInput = "invalid";
66         InputStream inputStream = new ByteArrayInputStream(invalidInput.getBytes());
67         System.setIn(inputStream);
68
69         assertFalse(model.processInput(invalidInput)); // 测试无效输入
70         assertFalse(model.isGameWon());
71     }
72
73
```

 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](#)

76
77

src/main/java/org/example/Main.java

```
1 package org.example;
2
3
4 import javax.script.ScriptEngineManager;
5
6 public class Main {
7     public static void main(String[] args) {
8         javax.swing.SwingUtilities.invokeLater(
9
10             new Runnable() {
11
12                 public void run() {
13                     createAndShowGUI();
14                 }
15             }
16         );
17     } public static void createAndShowGUI() {
18         INumberleModel model = new NumberleModel();
19         NumberleController controller=new NumberleController(model);
20         NumberleView view = new NumberleView(model, controller);
21         ScriptEngineManager manager = new ScriptEngineManager();
22
23
24     }
25 }
```

src/main/java/org/example/NumberleCLI.java

```
1 package org.example;
2
3 import java.util.Scanner;
4 import net.objecthunter.exp4j.Expression;
```

[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](#)

```
6
7
8
9 public class NumberleCLI {
10     private static final String ANSI_RESET = "\u001B[0m";
11     private static final String ANSI_GREEN = "\u001B[32m";
12     private static final String ANSI_YELLOW = "\u001B[33m";
13     private static final String ANSI_GRAY = "\u001B[90m";
14
15     private static final INumberleModel model = new NumberleModel();
16     private static final Scanner scanner = new Scanner(System.in);
17
18     public static void main(String[] args) {
19         model.startNewGame();
20         playGame();
21     }
22
23     private static boolean validateExpression(String input) {
24         String[] parts = input.split("=");
25         if (parts.length != 2) {
26             return false;
27         }
28         Expression expLeft = new ExpressionBuilder(parts[0].trim()).build();
29         double leftResult = expLeft.evaluate();
30         Expression expRight = new ExpressionBuilder(parts[1].trim()).build();
31         double rightResult = expRight.evaluate();
32
33         return Double.compare(leftResult, rightResult) == 0;
34     }
35     private static void playGame() {
36         while (!model.isGameOver()) {
37             printGameStatus();
38             String input = getInput();
39             if (validateExpression(input)) {
40                 boolean isCorrect = model.processInput(input);
41                 printColoredInput(input, model.getTargetWord());
42                 if (isCorrect) {
43
44                     break;
```

[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](#)

```
47         System.out.println("incorrect format please try again. ");
48     }
49 }
50 if (model.isGameWon()) {
51     System.out.println("you won!");
52 } else {
53     System.out.println("Game over ! your lose !");
54     System.out.println("The right answer is " + model.getTargetWord());
55 }
56 }
57
58 private static void printCategories() {
59     System.out.println("number: 0 1 2 3 4 5 6 7 8 9");
60     System.out.println("operator: + - * /");
61     System.out.println("other symbol: =");
62 }
63
64 private static String getInput() {
65     System.out.print("please enter your guess: ");
66     return scanner.nextLine();
67 }
68 private static void printGameStatus() {
69     System.out.println("the target equation: " + model.getTargetWord());
70     System.out.println("you only have : " + model.getRemainingAttempts()+" "+"chances");
71     System.out.println("your input guess is : " + model.getCurrentGuess());
72     printCategories();
73 }
74 private static void printColoredInput(String input, String target) {
75     StringBuilder sb = new StringBuilder();
76     sb.setLength(0);
77     for (int i = 0; i < input.length(); i++) {
78         char c = input.charAt(i);
79         if (i < target.length() && c == target.charAt(i)) {
80             sb.append(ANSI_GREEN).append(c);
81         } else if (target.contains(String.valueOf(c))) {
82             sb.append(ANSI_YELLOW).append(c);
83         } else {
84             sb.append(ANSI_GRAY).append(c);
85         }
86     }
87 }
```

 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](#)

```
88     System.out.println(sb.toString());
89 }
90 }
91
```

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("It's not correct answer,please try again");
29
30    }
31    public void setFlag2(Boolean Flag){
```

 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](#)

```
33     System.out.println(getTargetWord());
34 }
35 public void setView(NumberleView view) {
36     this.view = view;
37 }
38 public String getTargetWord() {
39     return model.getTargetWord();
40 }
41 public void setRemainingAttempts(int val) {
42     model.setRemainingAttempts(val);
43 }
44 public void processInput(String input) {
45     model.processInput(input);
46 }
47 public boolean isGameOver() {
48     return model.isGameOver();
49 }
50 public boolean isGameWon() {
51     return model.isGameWon();
52 }
53
54 }
55
```

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

```
1 package org.example;
2
3 import net.objecthunter.exp4j.ExpressionBuilder;
4
5 import java.io.File;
6 import java.util.ArrayList;
7 import java.util.Random;
8 import java.util.Scanner;
9 import java.io.FileNotFoundException;
10
11 // NumberleModel.java
12 public class NumberleModel extends java.util.Observable implements INumberleModel{
```

[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](#)

```
14 private String TargetWord;
15 public ArrayList<String> NumberList=new ArrayList<>();
16 public int remainAttempts=6 ;
17 public boolean gamewon;
18 public boolean Flag1=false;
19 public boolean Flag2=false ;
20 public boolean Flag3;
21
22 public String DefaultNumberal="3+4-2=5";
23
24 private StringBuilder CurrentGuess;
25
26
27
28 public void initialize(){
29
30     assert NumberList != null && !NumberList.isEmpty() : "NumberList should not be null or empty";
31     this.LoadList();
32     Random random=new Random();
33     int index =random.nextInt(NumberList.size());
34     if (Flag1==true){
35         System.out.println("this is not ");
36     }
37     if (Flag2==true){
38         System.out.println(getTargetWord());
39     }
40     if (Flag3==true){
41         TargetWord=DefaultNumberal;
42     }else {
43         TargetWord=NumberList.get(index);
44     }
45     CurrentGuess=new StringBuilder(TargetWord.length());
46     remainAttempts=MAX_ATTEMPTS;
47     gamewon=false;
48     setChanged();
49     notifyObservers();
50
51
52
53
```

[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](#)

```
55 public boolean ProcessInput(String input){
56     assert(input) != null;
57     assert input.length() <=7 : "Input length should be <= 7";
58     if (input.length() < 7){
59         return false;
60     }
61     CurrentGuess = new StringBuilder(input);
62
63     if (CurrentGuess.toString().equals(TargetWord)) {
64         gamewon = true;
65     }
66     setChanged();
67     notifyObservers();
68     return gamewon;
69 }
70
71 public void remove(){
72     if (CurrentGuess.length()>0){
73         CurrentGuess.deleteCharAt(CurrentGuess.length()-1);
74         setChanged();
75         notifyObservers();
76     }
77 }
78
79
80 public void LoadList(){
81     try {
82         File file=new File("C:/Users/93678/IdeaProjects/Numbera/src/equations.txt");
83         assert file.exists():"File should be exists";
84         Scanner scanner =new Scanner(file);
85
86         while(scanner.hasNextLine()){
87             this.NumberList.add(scanner.nextLine());
88         }
89         scanner.close();
90
91     } catch (FileNotFoundException e) {
92         System.out.println("This File can't read ");
93     }
```

 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](#)

```
96     }
97
98
99     @Override
100     public boolean processInput(String input) {
101         return ProcessInput(input);
102     }
103
104     @Override
105     public boolean isGameOver() {
106         assert remainAttempts >= 0 : "remainAttempts should be non-negative";
107         return remainAttempts <= 0 || gamewon ;
108     }
109
110     @Override
111     public boolean isGameWon() {
112         return gamewon;
113     }
114
115
116
117     @Override
118     public void setFlag(boolean flag2) {
119         this.Flag2=flag2;
120     }
121     @Override
122     public void setFlag3(boolean flag3) {
123         this.Flag3=flag3;
124     }
125
126
127
128
129     @Override
130     public void setTargetword(String targetword) {
131         assert targetword != null : "targetword should not be null";
132         assert targetword.length() >= 7 : "targetword length should be >= 7";
133         this.TargetWord=targetword;
134     }
135
```

[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](#)

```
137     public String getTargetWord() {
138         return TargetWord;
139     }
140
141     @Override
142     public StringBuilder getCurrentGuess() {
143         return CurrentGuess;
144     }
145
146     @Override
147     public int getRemainingAttempts() {
148         assert remainAttempts >= 0 : "remainAttempts should be non-negative";
149         return remainAttempts;
150     }
151
152     @Override
153     public void setRemainingAttempts(int val) {
154         assert val >= 0 : "val should be non-negative";
155         remainAttempts = val;
156
157     }
158
159
160     @Override
161     public boolean startNewGame() {
162         initialize();
163         return false;
164     }
165
166     @Override
167     public String DefaultWord() {
168         return null;
169     }
170
171 }
172
173 }
```

src/main/java/org/example/Number1eView.java

[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](#)

```
1 package org.example;
2
3
4 import javax.swing.*;
5 import javax.swing.plaf.basic.BasicOptionPaneUI;
6 import java.awt.*;
7 import java.awt.event.ActionEvent;
8 import java.awt.event.ActionListener;
9 import java.sql.SQLOutput;
10 import java.util.Observable;
11 import java.util.Observer;
12 import javax.swing.JOptionPane;
13
14
15 import net.objecthunter.exp4j.Expression;
16 import net.objecthunter.exp4j.ExpressionBuilder;
17
18
19
20
21 public class NumberleView extends JFrame implements Observer {
22     private JPanel panel;//game panel
23     private JLayeredPane layeredPane;
24     private NumberleController controller;// Controller
25     private final INumberleModel model;// Model
26     public JButton button[][]=new JButton[6][7]; // Game buttons
27     private JTextField inputTextField;
28     private int currentRow=0;
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     private boolean isDarkTheme = false;//Dark theme 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](#)

```
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     // Register view as observer to model
61     ((NumberleModel)this.model).addObserver(this); // Register this view as an observer to the model
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((NumberleModel)this.model, null); // Call update method to synchronize view with initial model state
68 }
69 public void initialize() {
70     // Set flag2 in controller to true
71     controller.setFlag2(true);
72
73     // Set the look and feel of the user interface to MetalLookAndFeel
74     try {
75         UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
76     } catch (Exception e) {
77         e.printStackTrace();
78     }
79
80     // Set the title of the frame
```

[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](#)

```
82
83 // Set default close operation to exit the application when the frame is closed
84 setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
85
86 // Set the frame size to 800x800 pixels
87 setSize(800, 800);
88
89 // Make the frame non-resizable
90 setResizable(false);
91
92 // Set the layout of the frame to BorderLayout
93 setLayout(new BorderLayout());
94
95 // Create a panel with a 6x7 grid layout for buttons
96 panel = new JPanel(new GridLayout(6, 7));
97
98 // Create buttons and add them to the panel
99 for (int i = 0; i < 6; i++) {
100     for (int j = 0; j < 7; j++) {
101         button[i][j] = new JButton("");
102         button[i][j].setPreferredSize(new Dimension(100, 100));
103         panel.add(button[i][j]);
104     }
105     // Add the panel to the frame's center position
106     add(panel, BorderLayout.CENTER);
107     panel.setVisible(true);
108 }
109
110 // Create an input panel with a text field
111 JPanel inputPanel = new JPanel();
112 JTextField inputTextField = new JTextField(7);
113 inputPanel.add(inputTextField);
114 add(inputPanel, BorderLayout.NORTH);
115 inputPanel.setVisible(false);
116
117 // Get text from input text field
118 String input = inputTextField.getText();
119
120 // Assert that input text field is not null or empty
```

[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](#)

```
123 // Create a keyboard panel with numeric buttons and operator buttons
124 JPanel keyBoardPanel = new JPanel();
125 keyBoardPanel.setLayout(new BoxLayout(keyBoardPanel, BoxLayout.Y_AXIS));
126 row1 = new JPanel();
127 row1.setLayout(new FlowLayout());
128 for (int i = 1; i <= 9; i++) {
129     JButton button = new JButton(Integer.toString(i));
130     button.setFont(new Font("Serif", Font.PLAIN, 30));
131     button.addActionListener(new ButtonClickListener());
132     row1.add(button);
133 }
134 JButton zeroButton = new JButton("0");
135 zeroButton.setFont(new Font("Serif", Font.PLAIN, 30));
136 zeroButton.addActionListener(new ButtonClickListener());
137 row1.add(zeroButton);
138
139 row2 = new JPanel();
140 row2.setLayout(new FlowLayout());
141 String[] operators = {"Delete", "+", "-", "*", "/", "=", "Enter"};
142 for (String operator : operators) {
143     JButton button = new JButton(operator);
144     button.setFont(new Font("Serif", Font.PLAIN, 30));
145     button.addActionListener(new ButtonClickListener());
146     row2.add(button);
147 }
148 keyBoardPanel.add(row1);
149 keyBoardPanel.add(row2);
150
151 // Add the keyboard panel to the frame's south position
152 add(keyBoardPanel, BorderLayout.SOUTH);
153
154 // Create a button panel with various control buttons
155 JPanel ButtonPanel = new JPanel();
156 JButton Information = new JButton("Information");
157 JButton Setting = new JButton("Setting");
158 JButton Swap = new JButton("Swap");
159 JButton restart = new JButton("Restart");
160 JButton FixFormual = new JButton("Fixed_Formula");
161
```

[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](#)

```
164         Fixed();
165     });
166     Setting.addActionListener(e -> {
167         Setting();
168     });
169     Information.addActionListener(e -> {
170         HowtoPlayGame();
171     });
172     Swap.addActionListener(e -> {
173         Swap();
174     });
175     restart.addActionListener(e -> {
176         // Start a new game when the restart button is clicked
177         controller.startNewGame();
178         // Print the new target word to the console
179         System.out.println("The new word is become " + " " + controller.getTargetWord());
180         currentRow = 0;
181         // Reset button text and background color
182         for (int row = 0; row < ROWS; row++) {
183             for (int col = 0; col < COLS; col++) {
184                 button[row][col].setText("");
185                 button[row][col].setBackground(defaultColor);
186             }
187         }
188         // Enable buttons in row1
189         for (Component component : row1.getComponents()) {
190             if (component instanceof JButton) {
191                 component.setEnabled(true);
192                 component.setBackground(defaultColor);
193             }
194         }
195         // Enable buttons in row2
196         for (Component component : row2.getComponents()) {
197             if (component instanceof JButton) {
198                 component.setEnabled(true);
199                 component.setBackground(defaultColor);
200             }
201         }
202     });
203 }
```

[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](#)

```
205     ButtonPanel.add(Setting);
206     ButtonPanel.add(Information);
207     ButtonPanel.add(Swap);
208     ButtonPanel.add(restart);
209     ButtonPanel.add(FixFormual);
210
211     // Add the button panel to the frame's north position
212     add(ButtonPanel, BorderLayout.NORTH);
213
214     // Make the frame visible
215     setVisible(true);
216
217     // Call the change method
218     change();
219 }
220 public void Fixed() {
221     controller.SetFlag(true);
222     System.out.println(controller.getTargetWord());
223 }
224
225 /**
226  * Removes all components from row2 panel and replaces them with operator buttons in a specific order.
227  * This method is called when the "Swap" button is clicked.
228  */
229 public void Swap() {
230     // Remove all components from row2 panel
231     row2.removeAll();
232
233     JButton deleteButton = null;
234     JButton enterButton = null;
235
236     // Find the "Delete" and "Enter" buttons in row2
237     for (Component component : row2.getComponents()) {
238         if (component != null && component instanceof JButton) {
239             JButton button = (JButton) component;
240             if (button.getText().equals("Delete")) {
241                 deleteButton = button;
242             } else if (button.getText().equals("Enter")) {
243                 enterButton = 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](#)

```
246     }
247
248     // Add operator buttons in a specific order to row2 panel
249     row2.setLayout(new FlowLayout());
250     for (String operator : new String[]{"Enter", "+", "-", "*", "/", "=", "Delete"}) {
251         JButton button;
252         if (operator.equals("Delete")) {
253             button = deleteButton != null ? deleteButton : new JButton("Delete");
254         } else if (operator.equals("Enter")) {
255             button = enterButton != null ? enterButton : new JButton("Enter");
256         } else {
257             button = new JButton(operator);
258         }
259         button.setFont(new Font("Serif", Font.PLAIN, 30));
260         button.addActionListener(new ButtonClickListener());
261         row2.add(button);
262     }
263
264     // Revalidates and repaints row2 panel to reflect changes
265     row2.revalidate();
266     row2.repaint();
267 }
268
269 /**
270  * Updates the buttons based on the input text.
271  * This method is called when there is a change in the input text field.
272  */
273 public void change() {
274     String input = inputTextField.getText();
275     updateButtonsFromInput(input);
276 }
277
278
279 public void Setting() {
280     try {
281         if (isDarkTheme) {
282             // Set the look and feel to MetalLookAndFeel for dark theme
283             UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
284             isDarkTheme = false;
```

[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](#)

```
287         UIManager.setLookAndFeel("javax.swing.plaf.nimbus.NimbusLookAndFeel");
288         isDarkTheme = true;
289     }
290 } catch (Exception e) {
291     e.printStackTrace();
292 }
293
294 // Update the UI components to reflect the theme change
295 SwingUtilities.updateComponentTreeUI(this);
296 for (Component component : getComponents()) {
297     SwingUtilities.updateComponentTreeUI(component);
298 }
299 }
300 /**
301  * ActionListener implementation for handling button clicks.
302  */
303 private class ButtonClickListener implements ActionListener {
304     @Override
305     public void actionPerformed(ActionEvent e) {
306         JButton source = (JButton) e.getSource();
307         String buttonText = source.getText();
308         switch (buttonText) {
309             case "Delete":
310                 clearDisplay();
311                 break;
312             case "Enter":
313                 Enter();
314
315                 break;
316             default:
317                 if (inputTextField.getText().length() < MAX_INPUT_LENGTH) {
318                     updateDisplay(buttonText);
319                     updateButtonsFromInput(inputTextField.getText());
320                 }
321                 controller.isGameOver();
322         }
323     }
324 }
325 /**
```

[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](#)

```
328     * @param input The user input as a string.
329     */
330     private void updateButtonsFromTarget(char[] target, String input) {
331         char[] userInput = input.toCharArray();
332
333         for (int col = 0; col < 7; col++) {
334
335             char targetChar = target[col];
336             char inputChar = userInput[col];
337
338             if (targetChar == inputChar) {
339                 button[currentRow][col].setBackground(Color.GREEN);
340             } else if (new String(target).indexOf(inputChar) != -1) {
341                 button[currentRow][col].setBackground(Color.ORANGE);
342             } else {
343                 button[currentRow][col].setBackground(Color.GRAY);
344             }
345
346             button[currentRow][col].setText(String.valueOf(inputChar));
347
348
349             String buttonText = button[currentRow][col].getText();
350
351
352             for (Component component : row1.getComponents()) {
353                 if (component instanceof JButton) {
354                     JButton keyboardButton = (JButton) component;
355                     if (keyboardButton.getText().equals(buttonText)) {
356                         if (targetChar == inputChar) {
357                             keyboardButton.setBackground(Color.GREEN);
358                         } else if (new String(target).indexOf(inputChar) != -1) {
359                             keyboardButton.setBackground(Color.ORANGE);
360                         } else {
361                             keyboardButton.setBackground(Color.GRAY);
362                         }
363                         break;
364                     }
365                 }
366             }
367         }
368     }
369 }
```

[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](#)

```
369         for (Component component : row2.getComponents()) {
370             if (component instanceof JButton) {
371                 JButton keyboardButton = (JButton) component;
372                 if (keyboardButton.getText().equals(buttonText)) {
373                     if (targetChar == inputChar) {
374                         keyboardButton.setBackground(Color.GREEN);
375                     } else if (new String(target).indexOf(inputChar) != -1) {
376                         keyboardButton.setBackground(Color.ORANGE);
377                     } else {
378                         keyboardButton.setBackground(Color.GRAY);
379                     }
380                     break;
381                 }
382             }
383         }
384     }
385 }
386
387 /**
388  * Updates the text of buttons based on the user input.
389  * @param input The user input as a string.
390  */
391 private void updateButtonsFromInput(String input) {
392     int rowIndex = currentRow;
393
394     int colIndex = 0;
395     char[] chars = input.toCharArray();
396
397     for (char c : chars) {
398         if (Character.isDigit(c) || "+-*/=".indexOf(c) != -1) {
399             if (colIndex < 7) {
400                 button[rowIndex][colIndex].setText(String.valueOf(c));
401                 colIndex++;
402             }
403         }
404     }
405
406     for (int clearIndex = colIndex; clearIndex < 7; clearIndex++) {
407         button[rowIndex][clearIndex].setText("");
408     }
409 }
```

[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](#)

```
410
411     }
412     /**
413      * Handles the action when the "Enter" button is clicked.
414      */
415     public void Enter() {
416         String input = inputTextField.getText();
417
418         // Check if input length is less than 7
419         if (input.length() < 7) {
420             controller.setflag1(true);
421             TooShort();
422
423             return; // Return to avoid further processing
424         }
425
426         boolean containsOperator = false; // Flag to check if an operator is present
427
428         for (char c : input.toCharArray()) {
429             if ("+-*/".indexOf(c) != -1) {
430                 containsOperator = true; // Set flag if an operator is found
431                 break;
432             }
433         }
434
435         if (input.length() == 7) {
436             boolean containsEquals = input.contains("=");
437             if (!containsEquals) {
438
439                 NoEqualSign();
440             } else if (!containsOperator) {
441                 Symbol();
442             } else {
443                 boolean isCorrect = validateEquation(input);
444                 if (isCorrect) {
445                     char[] target = controller.getTargetWord().toCharArray();
446                     validateEquation(input);
447
448                     controller.setRemainingAttempts(controller.getRemainingAttempts() - input.length() / 7);
```

[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](#)

```
451         updateButtonsFromTarget(target,input);
452         currentRow++;
453
454         if (currentRow>5){
455             currentRow=0;
456         }
457         controller.processInput(input);
458
459         inputTextField.setText(""); // Clear the input text field
460         if (controller.getRemainingAttempts()==0){
461             Youlost();
462         }
463     }
464     else{
465
466         NotEqual();
467     }
468 }
469 }
470
471 }
472
473
474
475
476
477
478 /**
479  * Validates the equation provided by the user.
480  *
481  * @param input The equation input by the user.
482  * @return True if the equation is valid, false otherwise.
483  */
484 private boolean validateEquation(String input) {
485     String[] parts = input.split("="); // Split the input into left and right sides of the equation
486     if (parts.length != 2) {
487         return false; // Return false if the equation doesn't contain exactly one equals sign
488     }
489
490     // For the rest of the method, we will assume that the input is a valid equation
491     // and we will only return false if the equation is not valid.
```

[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](#)

```
492     double leftResult = expLeft.evaluate();
493
494     // Evaluate the right side of the equation
495     Expression expRight = new ExpressionBuilder(parts[1].trim()).build();
496     double rightResult = expRight.evaluate();
497
498     // Compare the results of the left and right sides of the equation
499     return Double.compare(leftResult, rightResult) == 0;
500 }
501
502
503
504
505
506 /**
507  * Clears the display by removing the last character from the input text field.
508  */
509 public void clearDisplay() {
510     String currentText = inputTextField.getText();
511     if (!currentText.isEmpty()) {
512         // Remove the last character from the current text
513         String newText = currentText.substring(0, currentText.length() - 1);
514         inputTextField.setText(newText);
515         // Update buttons based on the new input text
516         updateButtonsFromInput(newText);
517     }
518 }
519
520
521 /**
522  * Updates the display by appending the given text to the input text field.
523  * If the length of the resulting text exceeds 7 characters, it truncates the text appropriately.
524  *
525  * @param text The text to be added to the input text field.
526  */
527 public void updateDisplay(String text) {
528     // Check if adding the text will exceed the maximum length
529     if (inputTextField.getText().length() + text.length() > 7) {
530         // Truncate the text to fit within the maximum length
```

[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](#)

```
533 // Append the text to the input text field
534 inputTextField.setText(inputTextField.getText() + text);
535 }
536
537
538
539
540 /**
541  * Displays a reminder frame indicating that the input is too short.
542  * The reminder frame is displayed for 2 seconds before automatically closing.
543  */
544 public void TooShort() {
545     // Create a reminder frame
546     JFrame reminderFrame = new JFrame("reminder");
547     reminderFrame.setBounds(200, 200, 400, 200);
548     reminderFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
549
550     // Create a label for the reminder message
551     JLabel reminderLabel = new JLabel("Too Short !");
552     reminderLabel.setFont(new Font("Serif", Font.BOLD, 25));
553     reminderLabel.setHorizontalAlignment(JLabel.CENTER);
554
555     // Create a panel to hold the reminder label
556     JPanel reminderPanel = new JPanel(new BorderLayout());
557     reminderPanel.add(reminderLabel, BorderLayout.CENTER);
558     reminderFrame.getContentPane().add(reminderPanel);
559
560     // Make the reminder frame visible
561     reminderFrame.setVisible(true);
562
563     // Set a timer to automatically close the reminder frame after 2 seconds
564     Timer timer = new Timer(2000, e -> {
565         reminderFrame.dispose();
566     });
567     timer.setRepeats(false);
568     timer.start();
569 }
570
571 /**
```

[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](#)

```
574     *
575     */
576     public void Symbol() {
577         // Create a reminder frame
578         JFrame reminderFrame = new JFrame("reminder");
579         reminderFrame.setBounds(200, 200, 500, 200);
580         reminderFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
581
582         // Create a label for the reminder message
583         JLabel reminderLabel = new JLabel("There must be at least one sign +-*/ !");
584         reminderLabel.setFont(new Font("Serif", Font.BOLD, 25));
585         reminderLabel.setHorizontalAlignment(JLabel.CENTER);
586
587         // Create a panel to hold the reminder label
588         JPanel reminderPanel = new JPanel(new BorderLayout());
589         reminderPanel.add(reminderLabel, BorderLayout.CENTER);
590         reminderFrame.getContentPane().add(reminderPanel);
591
592         // Make the reminder frame visible
593         reminderFrame.setVisible(true);
594
595         // Set a timer to automatically close the reminder frame after 2 seconds
596         Timer timer = new Timer(2000, e -> {
597             reminderFrame.dispose();
598         });
599         timer.setRepeats(false);
600         timer.start();
601     }
602
603     /**
604     * Displays a reminder frame explaining how to play the game.
605     * The reminder frame contains instructions on how to guess the hidden mathematical equation in 6 tries
606     * and how the color of the tiles changes to show the proximity to the correct solution.
607     * The reminder frame is displayed until closed by the user.
608     */
609     public void HowtoPlayGame() {
610         // Create a frame to display the instructions
611         JFrame reminderFrame = new JFrame("How to play");
612         reminderFrame.setBounds(0, 0, 1000, 1000);
613         reminderFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
614         reminderFrame.setVisible(true);
615     }
616 }
```

[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](#)

```
615 // Create a text area to display the instructions
616 JTextArea textArea = new JTextArea();
617 textArea.setLineWrap(true); // Enable automatic line wrapping
618 textArea.setWrapStyleWord(true);
619 textArea.setText("You have to guess the hidden math equation in 6 tries and color of the tiles " +
620     "change to show how close you are.\n" +
621     "To start playing, enter any mathematical equation. For example: 4+7*5=39.\n" +
622     "Numbers that are not in the target equation are colored gray, numbers that are in the " +
623     "equation but in the wrong spot are colored orange, and correct numbers are colored green.\n" +
624     "Guess the target equation correctly to win the game!");
625
626 // Set font and size for the text area
627 textArea.setFont(new Font("Serif", Font.ROMAN_BASELINE, 30));
628
629 // Create a panel to hold the text area
630 JPanel reminderPanel = new JPanel(new BorderLayout());
631 reminderPanel.add(textArea, BorderLayout.CENTER);
632 reminderFrame.getContentPane().add(reminderPanel);
633
634 // Make the reminder frame visible
635 reminderFrame.setVisible(true);
636 }
637
638
639 /**
640  * Displays a reminder frame indicating that the left side of the equation is not equal to the right side.
641  * Displays a reminder frame indicating that the left side of the equation is not equal to the right side.
642  * The reminder frame is displayed until closed by the user.
643  */
644 public void NotEqual() {
645     // Create a frame to display the reminder
646     JFrame reminderFrame = new JFrame("reminder");
647     reminderFrame.setBounds(200, 200, 500, 200);
648     reminderFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
649
650     // Create a label to display the reminder message
651     JLabel reminderLabel = new JLabel("The left side is not equal to the right");
652     reminderLabel.setFont(new Font("Serif", Font.BOLD, 25));
653     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](#)

```
656 JPanel reminderPanel = new JPanel(new BorderLayout());
657 reminderPanel.add(reminderLabel, BorderLayout.CENTER);
658 reminderFrame.getContentPane().add(reminderPanel);
659
660 // Make the reminder frame visible
661 reminderFrame.setVisible(true);
662
663 // Close the reminder frame after a delay
664 Timer timer = new Timer(2000, e -> {
665     reminderFrame.dispose();
666 });
667 timer.setRepeats(false);
668 timer.start();
669 }
670
671 /**
672  * Displays a reminder frame indicating the game start and prompts the user to guess the first equation.
673  * The reminder frame is displayed until closed by the user.
674  */
675 private void showGameStartReminder() {
676     // Create a frame to display the reminder
677     JFrame reminderFrame = new JFrame("reminder");
678     reminderFrame.setBounds(150, 200, 450, 300);
679     reminderFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
680
681     // Create a label to display the reminder message
682     JLabel reminderLabel = new JLabel("Guess the first equation!");
683     reminderLabel.setFont(new Font("Serif", Font.BOLD, 25));
684     reminderLabel.setHorizontalAlignment(JLabel.CENTER);
685
686     // Create a panel to hold the label
687     JPanel reminderPanel = new JPanel(new BorderLayout());
688     reminderPanel.add(reminderLabel, BorderLayout.CENTER);
689     reminderFrame.getContentPane().add(reminderPanel);
690
691     // Make the reminder frame visible
692     reminderFrame.setVisible(true);
693
694     // Close the reminder frame after a delay
695     Timer timer = new Timer(2000, e -> {
```

[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](#)

```
697     });
698     timer.setRepeats(false);
699     timer.start();
700 }
701
702 /**
703  * Displays a reminder frame indicating that there is no equal (=) sign in the input.
704  * The reminder frame is displayed until closed by the user.
705  */
706 public void NoEqualSign() {
707     // Create a frame to display the reminder
708     JFrame reminderFrame = new JFrame("reminder");
709     reminderFrame.setBounds(200, 200, 400, 200);
710     reminderFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
711
712     // Create a label to display the reminder message
713     JLabel reminderLabel = new JLabel("No equal (=) sign!");
714     reminderLabel.setFont(new Font("Serif", Font.BOLD, 25));
715     reminderLabel.setHorizontalAlignment(JLabel.CENTER);
716
717     // Create a panel to hold the label
718     JPanel reminderPanel = new JPanel(new BorderLayout());
719     reminderPanel.add(reminderLabel, BorderLayout.CENTER);
720     reminderFrame.getContentPane().add(reminderPanel);
721
722     // Make the reminder frame visible
723     reminderFrame.setVisible(true);
724
725     // Close the reminder frame after a delay
726     Timer timer = new Timer(2000, e -> {
727         reminderFrame.dispose();
728     });
729     timer.setRepeats(false);
730     timer.start();
731 }
732
733 /**
734  * Displays a reminder frame indicating that the player has lost the game.
735  * Provides an option to replay the game.
```

[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](#)

```
738 // Create a frame to display the reminder
739 JFrame reminderFrame = new JFrame("reminder");
740 reminderFrame.setBounds(200, 200, 400, 200);
741 reminderFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
742
743 // Create a button for replaying the game
744 Button Replay = new Button("Replay");
745 Replay.addActionListener(e -> {
746     controller.startNewGame();
747
748     for (int row = 0; row < ROWS; row++) {
749         for (int col = 0; col < COLS; col++) {
750             button[row][col].setText("");
751             button[row][col].setBackground(defaultColor);
752         }
753     }
754
755     // Enable buttons for input
756     for (Component component : row1.getComponents()) {
757         if (component instanceof JButton) {
758             component.setEnabled(true);
759             component.setBackground(defaultColor);
760         }
761     }
762     // Enable operator buttons
763     for (Component component : row2.getComponents()) {
764         if (component instanceof JButton) {
765             component.setEnabled(true);
766             component.setBackground(defaultColor);
767         }
768     }
769     // Close the reminder frame
770     reminderFrame.dispose();
771 });
772 Replay.setBounds(10, 20, 10, 10);
773
774 // Create a label to display the reminder message
775 JLabel reminderLabel = new JLabel("You lost!");
776 reminderLabel.setFont(new Font("Serif", Font.BOLD, 25));
777 reminderLabel.setHorizontalAlignment(SwingConstants.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](#)

```
779 // Create a panel to hold the label and replay button
780 JPanel reminderPanel = new JPanel(new BorderLayout());
781 reminderPanel.add(reminderLabel, BorderLayout.CENTER);
782 reminderPanel.add(Replay, BorderLayout.NORTH);
783
784 // Add the panel to the reminder frame
785 reminderFrame.getContentPane().add(reminderPanel);
786
787 // Make the replay button and reminder frame visible
788 Replay.setVisible(true);
789 reminderFrame.setVisible(true);
790 }
791
792 /**
793  * Displays a reminder frame indicating that the player has won the game.
794  * Provides an option to replay the game.
795  */
796 public void Youwon() {
797     // Create a frame to display the reminder
798     JFrame reminderFrame = new JFrame("reminder");
799     reminderFrame.setBounds(200, 200, 400, 200);
800     reminderFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
801
802     // Create a button for restarting the game
803     Button Restart = new Button("Replay");
804     Restart.addActionListener(e -> {
805         currentRow = 0;
806         // Reset the game board
807         for (int row = 0; row < ROWS; row++) {
808             for (int col = 0; col < COLS; col++) {
809                 button[row][col].setText("");
810                 button[row][col].setBackground(defaultColor);
811             }
812         }
813         // Enable buttons for input
814         for (Component component : row1.getComponents()) {
815             if (component instanceof JButton) {
816                 component.setEnabled(true);
817                 component.setBackground(defaultColor);
818             }
819         }
820     });
821 }
```

[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](#)

```
820         // Enable operator buttons
821         for (Component component : row2.getComponents()) {
822             if (component instanceof JButton) {
823                 component.setEnabled(true);
824                 component.setBackground(defaultColor);
825             }
826         }
827         // Close the reminder frame
828         reminderFrame.dispose();
829     });
830     Restart.setBounds(10, 20, 10, 10);
831
832     // Create a label to display the reminder message
833     JLabel reminderLabel = new JLabel("You won!");
834     reminderLabel.setFont(new Font("Serif", Font.BOLD, 25));
835     reminderLabel.setHorizontalAlignment(JLabel.CENTER);
836
837     // Create a panel to hold the label and restart button
838     JPanel reminderPanel = new JPanel(new BorderLayout());
839     reminderPanel.add(reminderLabel, BorderLayout.CENTER);
840     reminderPanel.add(Restart, BorderLayout.NORTH);
841
842     // Add the panel to the reminder frame
843     reminderFrame.getContentPane().add(reminderPanel);
844
845     // Make the restart button and reminder frame visible
846     Restart.setVisible(true);
847     reminderFrame.setVisible(true);
848 }
849
850 @Override
851
852 public void update(Observable o, Object arg) {
853
854
855     if (controller.isGameOver()) {
856         if (controller.isGameWon()) {
857             Youwon();
858             controller.startNewGame();
859         }
860     }
861 }
```

 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](#)

```
861     }
862
863     for (Component component : row1.getComponents()) {
864         if (component instanceof JButton) {
865             component.setEnabled(false);
866         }
867     }
868     for (Component component : row2.getComponents()) {
869         if (component instanceof JButton) {
870             component.setEnabled(false);
871         }
872     }
873 }
874 }
875 }
876 }
```