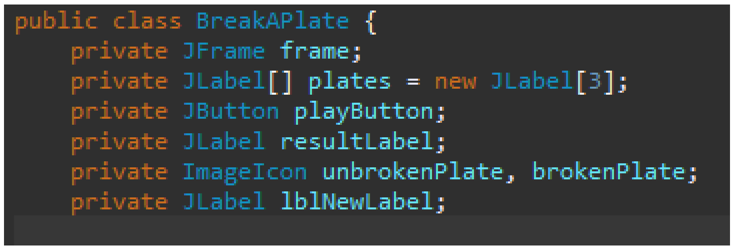
Reflection log

Class Definition:

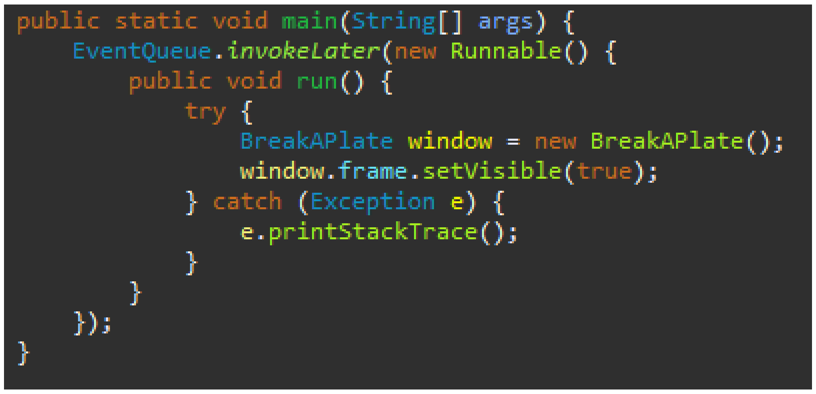
The `BreakAPlate` class creates a simple graphical user interface (GUI) game where users can play a game involving breaking plates. The game provides visual feedback based on user interactions and random outcomes.



```
public class BreakAPlate {
    private JFrame frame;
    private JLabel[] plates = new JLabel[3];
    private JButton playButton;
    private JLabel resultLabel;
    private ImageIcon unbrokenPlate, brokenPlate;
    private JLabel lblNewLabel;
```

Main Method:

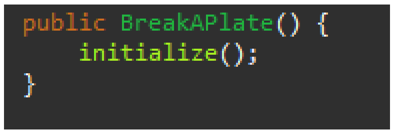
The main method serves as the application's entry point. It employs `EventQueue.invokeLater` to ensure that the GUI is constructed on the Event Dispatch Thread, adhering to best practices for thread safety and responsiveness in Java Swing applications.



```
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                BreakAPlate window = new BreakAPlate();
                window.frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}
```

Constructor:

The constructor (`BreakAPlate()`) initializes the GUI by calling the `initialize()` method, setting up all necessary components for user interaction.



```
public BreakAPlate() {
    initialize();
}
```

Initialize Method:

The `initialize` method sets up the main window (`JFrame`) and the UI components:

- JFrame: The main window of the application.
- JPanel: A panel that holds the plates and buttons.
- JLabel: Displays the plates, results, and images related to the game.
- JButton: A button for users to play the game.

```
private void initialize() {
    frame = new JFrame();
    frame.setBounds(100, 100, 450, 408);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(new BorderLayout());
```

Components:

- JLabel Array (plates): An array of labels representing the plates, which can be displayed as broken or unbroken.
- JButton (playButton): A button that triggers the game action when clicked.
- JLabel (resultLabel): Displays the result of the game, informing the user about their winnings.
- ImageIcon: Two icons representing the unbroken and broken states of a plate, enhancing the visual aspect of the game.

```
unbrokenPlate = new ImageIcon("unbroken_plate.png"); // Replace with actual path
brokenPlate = new ImageIcon("broken_plate.png");     // Replace with actual path


// Plate panel
JPanel platePanel = new JPanel();
for (int i = 0; i < plates.length; i++) {
    plates[i] = new JLabel(unbrokenPlate);
    platePanel.add(plates[i]);
}


// Result label
resultLabel = new JLabel(" ");
resultLabel.setHorizontalAlignment(SwingConstants.CENTER);


frame.getContentPane().add(platePanel, BorderLayout.CENTER);
platePanel.setLayout(null);

// C:\\Users\\1100062345\\Downloads\\plates_all_broken.gif
lblNewLabel = new JLabel("");
lblNewLabel.setBounds(85, 25, 308, 80);
lblNewLabel.setIcon(new ImageIcon("C:\\Users\\1100062345\\Downloads\\plates.gif"));
platePanel.add(lblNewLabel);
ChangeImage(lblNewLabel, "plates");

JLabel lblNewLabel_3 = new JLabel("");
lblNewLabel_3.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_3.setBounds(85, 219, 308, 80);
platePanel.add(lblNewLabel_3);
frame.getContentPane().add(resultLabel, BorderLayout.NORTH);
```

Button ActionListener:

An `ActionListener` is attached to the `playButton`. When clicked, it generates random values to determine which plates are broken, updates the display, and sets the result message based on the outcome of the game. It also changes the button text to "Play Again" after the first play.

```java
JPanel buttonPanel = new JPanel();
buttonPanel.setBounds(184, 142, 96, 33);
platePanel.add(buttonPanel);
playButton = new JButton("Play");
playButton.addActionListener(new ActionListener() {
```

Game Logic:

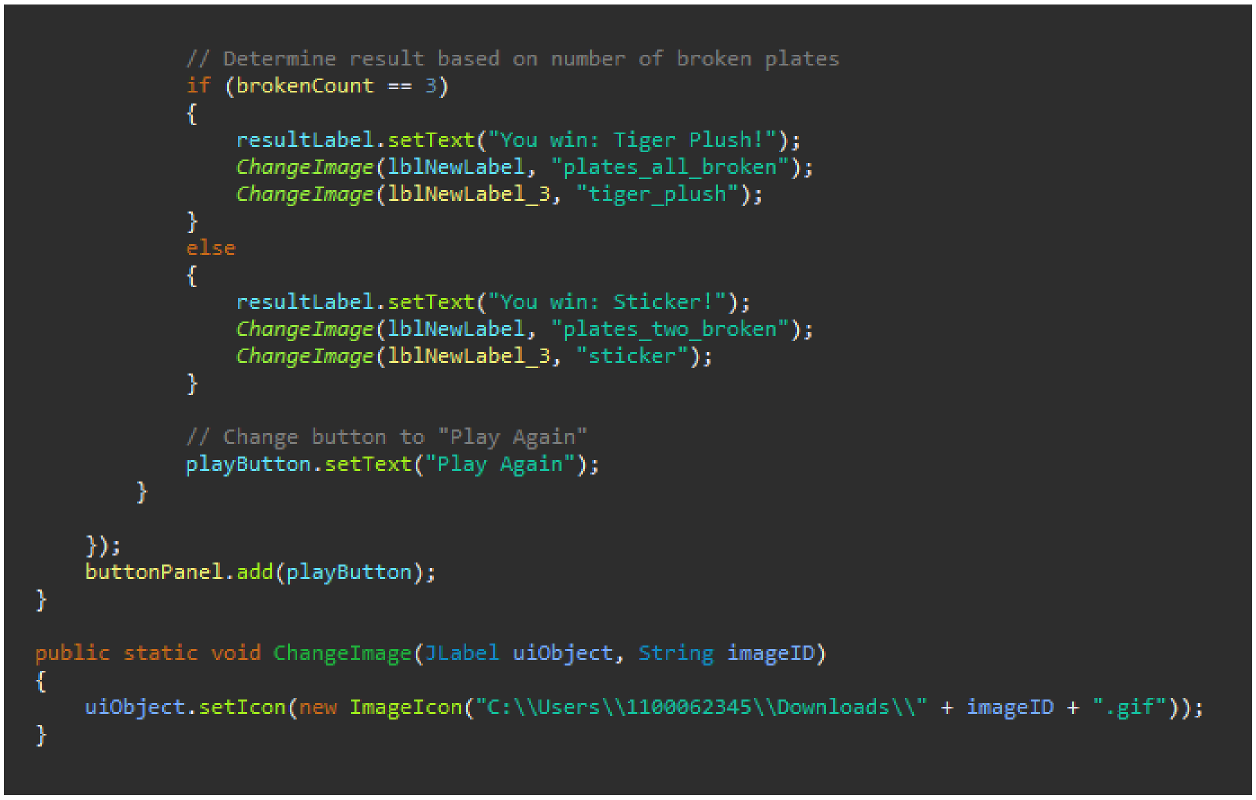
The core logic for the game resides in the `actionPerformed` method of the button's listener:

- Random Plate Status: It generates random values to randomly determine whether each plate is broken or unbroken.
- Result Evaluation: Depending on how many plates are broken, it updates the `resultLabel` with different outcomes (e.g., winning a tiger plush or a sticker) and changes relevant images to reflect the game's state.

```java
public void actionPerformed(ActionEvent e) {
    Random random = new Random();
    int brokenCount = 0;

    ChangeImage(lblNewLabel, "plates_all_broken");

     // Generate random values and update plate images
    for (int i = 0; i < plates.length; i++) {
        int randValue = random.nextInt(2);  // Generate 0 or 1
        if (randValue == 1) {
            plates[i].setIcon(brokenPlate);
            brokenCount++;
        } else {
            plates[i].setIcon(unbrokenPlate);
        }
    }


    // Determine result based on number of broken plates
    if (brokenCount == 3)
    {
        resultLabel.setText("You win: Tiger Plush!");
        ChangeImage(lblNewLabel, "plates_all_broken");
        ChangeImage(lblNewLabel_3, "tiger_plush");
    }
    else
    {
        resultLabel.setText("You win: Sticker!");
        ChangeImage(lblNewLabel, "plates_two_broken");
        ChangeImage(lblNewLabel_3, "sticker");
    }

    // Change button to "Play Again"
    playButton.setText("Play Again");
}
```

User Feedback:

The `resultLabel` provides immediate feedback on game results, and additional labels display images corresponding to the outcomes, helping users engage with the game dynamically.



```
            // Determine result based on number of broken plates
            if (brokenCount == 3)
            {
                resultLabel.setText("You win: Tiger Plush!");
                ChangeImage(lblNewLabel, "plates_all_broken");
                ChangeImage(lblNewLabel_3, "tiger_plush");
            }
            else
            {
                resultLabel.setText("You win: Sticker!");
                ChangeImage(lblNewLabel, "plates_two_broken");
                ChangeImage(lblNewLabel_3, "sticker");
            }

            // Change button to "Play Again"
            playButton.setText("Play Again");
        }

    });
    buttonPanel.add(playButton);
}

public static void ChangeImage(JLabel uiObject, String imageID)
{
    uiObject.setIcon(new ImageIcon("C:\\Users\\1100062345\\Downloads\\" + imageID + ".gif"));
}
```