```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Random;

/**
 * ArithmeticGame.java
 * This class creates the main JFrame for the arithmetic game.
 * It handles the UI layout, question generation, and event handling.
 */
public class ArithmeticGame extends JFrame implements ActionListener {

    // --- UI Components ---
    // Labels for displaying numbers and operators
    private JLabel lblNum1, lblNum2, lblOperator, lblEquals;
    // Labels for score and feedback
    private JLabel lblScore, lblCorrect, lblIncorrect, lblFeedback;

    // Text field for user input
    private JTextField txtAnswer;

    // Buttons for game control
    private JButton btnSubmit, btnNext, btnExit, btnGo;

    // Radio buttons for selecting operation
    private JRadioButton rdoAdd, rdoSubtract, rdoMultiply, rdoDivide, rdoModulo;

    // Radio buttons for selecting difficulty level
    private JRadioButton rdoLevel1, rdoLevel2, rdoLevel3;

    // Button groups to ensure only one radio button is selected per group
    private ButtonGroup operationGroup;
    private ButtonGroup levelGroup;

    // Panels for organizing components
    private JPanel mainPanel, questionPanel, operationPanel, levelPanel, controlPanel,
scorePanel;

    // --- Game Logic Variables ---
    private Random random = new Random();
    private int num1, num2, correctAnswer;
    private int correctCount = 0;
    private int incorrectCount = 0;
```

```
private String currentOperation = "+";
private int minRange = 1;
private int maxRange = 100;

/**
 * Constructor: Sets up the entire game window.
 */
public ArithmeticGame() {
    // --- 1. Set up the Frame (Main Window) ---
    setTitle("Arithmetic Game");
    setSize(650, 450); // Set window size
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // Exit app when 'X' is clicked
    setLocationRelativeTo(null); // Center the window on screen
    setLayout(new BorderLayout(10, 10)); // Use BorderLayout

    // --- 2. Initialize Components ---

    // Question Panel (Top)
    questionPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 15, 20));
    lblNum1 = createDisplayLabel("167");
    lblOperator = createDisplayLabel("+");
    lblNum2 = createDisplayLabel("142");
    lblEquals = createDisplayLabel("=");
    txtAnswer = new JTextField(5); // Text field for 5 columns
    txtAnswer.setFont(new Font("Inter", Font.BOLD, 48));
    txtAnswer.setHorizontalAlignment(JTextField.CENTER);

    questionPanel.add(lblNum1);
    questionPanel.add(lblOperator);
    questionPanel.add(lblNum2);
    questionPanel.add(lblEquals);
    questionPanel.add(txtAnswer);

    // Main Panel (Center) - will hold operations, levels, and feedback
    mainPanel = new JPanel(new BorderLayout(10, 10));
    mainPanel.setBorder(BorderFactory.createEmptyBorder(10, 20, 10, 20));

    // Operation Panel
    operationPanel = new JPanel(new GridLayout(6, 1)); // 6 rows, 1 column
    operationPanel.setBorder(BorderFactory.createTitledBorder("Operation"));
    rdoAdd = new JRadioButton("Addition", true); // Default selected
    rdoSubtract = new JRadioButton("Subtraction");
    rdoMultiply = new JRadioButton("Multiplication");
    rdoDivide = new JRadioButton("Division");
```

```
rdoModulo = new JRadioButton("Modulo");

// Group operation radio buttons
operationGroup = new ButtonGroup();
operationGroup.add(rdoAdd);
operationGroup.add(rdoSubtract);
operationGroup.add(rdoMultiply);
operationGroup.add(rdoDivide);
operationGroup.add(rdoModulo);

operationPanel.add(new JLabel("Operation:"));
operationPanel.add(rdoAdd);
operationPanel.add(rdoSubtract);
operationPanel.add(rdoMultiply);
operationPanel.add(rdoDivide);
operationPanel.add(rdoModulo);

// Level Panel
levelPanel = new JPanel(new GridLayout(5, 1)); // 5 rows, 1 column
levelPanel.setBorder(BorderFactory.createTitledBorder("Level"));
rdoLevel1 = new JRadioButton("Level 1 (1-100)", true); // Default selected
rdoLevel2 = new JRadioButton("Level 2 (101-200)");
rdoLevel3 = new JRadioButton("Level 3 (201-300)");
btnGo = new JButton("GO");

// Group level radio buttons
levelGroup = new ButtonGroup();
levelGroup.add(rdoLevel1);
levelGroup.add(rdoLevel2);
levelGroup.add(rdoLevel3);

levelPanel.add(new JLabel("Level:"));
levelPanel.add(rdoLevel1);
levelPanel.add(rdoLevel2);
levelPanel.add(rdoLevel3);
levelPanel.add(btnGo);

// Feedback Label (below the question)
lblFeedback = new JLabel("Select an operation and level, then press 'GO' to start!");
lblFeedback.setFont(new Font("Inter", Font.PLAIN, 14));
lblFeedback.setHorizontalAlignment(JLabel.CENTER);

// Add sub-panels to the main center panel
mainPanel.add(operationPanel, BorderLayout.WEST);
```

```
mainPanel.add(levelPanel, BorderLayout.CENTER);
mainPanel.add(lblFeedback, BorderLayout.SOUTH);

// Control Panel (Right)
controlPanel = new JPanel(new BorderLayout(10, 10));
controlPanel.setBorder(BorderFactory.createEmptyBorder(0, 0, 10, 20));

// Score Panel (part of Control Panel)
scorePanel = new JPanel(new GridLayout(3, 2, 5, 5)); // 3 rows, 2 columns
scorePanel.setBorder(BorderFactory.createTitledBorder("Score:"));
lblCorrect = createScoreLabel("0");
lblIncorrect = createScoreLabel("0");

scorePanel.add(new JLabel("Correct:", JLabel.CENTER));
scorePanel.add(new JLabel("Incorrect:", JLabel.CENTER));
scorePanel.add(lblCorrect);
scorePanel.add(lblIncorrect);

// Button Panel (part of Control Panel)
JPanel buttonPanel = new JPanel(new GridLayout(3, 1, 5, 10)); // 3 rows, 1 col
btnSubmit = new JButton("Submit");
btnNext = new JButton("Next");
btnExit = new JButton("Exit");

buttonPanel.add(btnSubmit);
buttonPanel.add(btnNext);
buttonPanel.add(btnExit);

controlPanel.add(scorePanel, BorderLayout.NORTH);
controlPanel.add(buttonPanel, BorderLayout.SOUTH);

// --- 3. Add Listeners ---
btnSubmit.addActionListener(this);
btnNext.addActionListener(this);
btnExit.addActionListener(this);
btnGo.addActionListener(this);

// Also add listener to 'Enter' key in text field
txtAnswer.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        checkAnswer(); // Same action as clicking Submit
    }
});
```

```
    // --- 4. Add Components to Frame ---
    add(questionPanel, BorderLayout.NORTH);
    add(mainPanel, BorderLayout.CENTER);
    add(controlPanel, BorderLayout.EAST);

    // --- 5. Finalize and Show ---
    generateQuestion(); // Generate the first question
    setVisible(true); // Make the window visible
}

/**
 * Helper method to create the big display labels for numbers.
 */
private JLabel createDisplayLabel(String text) {
    JLabel label = new JLabel(text);
    label.setFont(new Font("Inter", Font.BOLD, 48));
    label.setPreferredSize(new Dimension(120, 80));
    label.setHorizontalAlignment(JLabel.CENTER);
    label.setOpaque(true);
    label.setBackground(new Color(0, 120, 150)); // Teal color
    label.setForeground(Color.WHITE);
    label.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    return label;
}

/**
 * Helper method to create the score labels.
 */
private JLabel createScoreLabel(String text) {
    JLabel label = new JLabel(text);
    label.setFont(new Font("Inter", Font.BOLD, 24));
    label.setHorizontalAlignment(JLabel.CENTER);
    label.setOpaque(true);
    label.setBackground(Color.WHITE);
    label.setBorder(BorderFactory.createLineBorder(Color.GRAY));
    return label;
}

/**
 * Main event handler for all button clicks.
 */
@Override
public void actionPerformed(ActionEvent e) {
```

```
    Object source = e.getSource();

    if (source == btnGo) {
        // User pressed "GO" - update settings and generate new question
        updateSettings();
        generateQuestion();
        lblFeedback.setText("New question generated!");
    } else if (source == btnSubmit) {
        // User pressed "Submit"
        checkAnswer();
    } else if (source == btnNext) {
        // User pressed "Next"
        generateQuestion();
        lblFeedback.setText("Here is the next question.");
    } else if (source == btnExit) {
        // User pressed "Exit"
        System.exit(0);
    }
}

/**
 * Reads the selected radio buttons to update game settings.
 */
private void updateSettings() {
    // Update Level
    if (rdoLevel1.isSelected()) {
        minRange = 1;
        maxRange = 100;
    } else if (rdoLevel2.isSelected()) {
        minRange = 101;
        maxRange = 200;
    } else if (rdoLevel3.isSelected()) {
        minRange = 201;
        maxRange = 300;
    }

    // Update Operation
    if (rdoAdd.isSelected()) {
        currentOperation = "+";
    } else if (rdoSubtract.isSelected()) {
        currentOperation = "-";
    } else if (rdoMultiply.isSelected()) {
        currentOperation = "*";
    } else if (rdoDivide.isSelected()) {
```

```
        currentOperation = "/";
    } else if (rdoModulo.isSelected()) {
        currentOperation = "%";
    }

    lblOperator.setText(currentOperation);
}

/**
 * Generates a new random question based on current settings.
 */
private void generateQuestion() {
    // Generate two random numbers in the selected range
    num1 = random.nextInt(maxRange - minRange + 1) + minRange;
    num2 = random.nextInt(maxRange - minRange + 1) + minRange;

    // Handle special cases for division and modulo
    if (currentOperation.equals("/") || currentOperation.equals("%")) {
        if (num2 == 0) {
            num2 = 1; // Avoid division by zero
        }
        // Ensure num1 is always greater than num2 for simpler division/modulo
        if (num1 < num2) {
            int temp = num1;
            num1 = num2;
            num2 = temp;
        }
    }

    // Calculate the correct answer
    switch (currentOperation) {
        case "+":
            correctAnswer = num1 + num2;
            break;
        case "-":
            correctAnswer = num1 - num2;
            break;
        case "*":
            correctAnswer = num1 * num2;
            break;
        case "/":
            correctAnswer = num1 / num2; // Integer division
            break;
        case "%":
```

```
            correctAnswer = num1 % num2;
            break;
    }

    // Update the display labels
    lblNum1.setText(String.valueOf(num1));
    lblNum2.setText(String.valueOf(num2));

    // Clear previous answer and feedback
    txtAnswer.setText("");
    lblFeedback.setText("Enter your answer and click 'Submit'.");
    txtAnswer.requestFocus(); // Put the cursor in the answer box
}

/**
 * Checks the user's answer against the correct answer.
 */
private void checkAnswer() {
    try {
        // Get user's answer from the text field
        String userAnswerStr = txtAnswer.getText().trim();
        if (userAnswerStr.isEmpty()) {
            lblFeedback.setText("Please enter an answer.");
            return;
        }

        int userAnswer = Integer.parseInt(userAnswerStr);

        // Compare user's answer with the correct answer
        if (userAnswer == correctAnswer) {
            correctCount++;
            lblFeedback.setText("Correct! Well done.");
            lblFeedback.setForeground(new Color(0, 128, 0)); // Dark green
        } else {
            incorrectCount++;
            lblFeedback.setText("Incorrect. The correct answer is " + correctAnswer);
            lblFeedback.setForeground(Color.RED);
        }

        // Update score labels
        lblCorrect.setText(String.valueOf(correctCount));
        lblIncorrect.setText(String.valueOf(incorrectCount));

    } catch (NumberFormatException e) {
```

```
            // Handle cases where the user types non-numeric characters
            lblFeedback.setText("Please enter a valid whole number.");
            lblFeedback.setForeground(Color.RED);
        }
    }
}
```

```
import javax.swing.SwingUtilities;

/**
 * GameTester.java
 * This class contains the main method to run the Arithmetic Game.
 * It ensures the GUI is created on the Event Dispatch Thread (EDT).
 */
public class GameTester {

    public static void main(String[] args) {

        // Use SwingUtilities.invokeLater to create and show the GUI
        // This is the standard and safest way to start a Swing application.
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                // Create an instance of our game window
                new ArithmeticGame();
            }
        });
    }
}
```