```java
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.TextField;
import javafx.scene.input.KeyCode;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;

public class SimpleCalculator extends Application {

    private TextField display = new TextField();
    private String currentInput = "";
    private double firstNumber = 0;
    private String operator = "";
    private boolean startNewNumber = true;

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("Simple Calculator");

        display.setEditable(false);
        display.setAlignment(Pos.CENTER_RIGHT);
        display.setStyle("-fx-font-size: 18px; -fx-pref-height: 50px;");

        GridPane grid = createButtonGrid();

        VBox root = new VBox(10, display, grid);
        root.setPadding(new Insets(10));
        root.setStyle("-fx-background-color: #f0f0f0;");

        Scene scene = new Scene(root, 300, 400);

        // Keyboard input support
        scene.setOnKeyPressed(event -> {
            if (event.getCode().isDigitKey() || event.getCode() == KeyCode.PERIOD) {
                appendToDisplay(event.getText());
            } else if (event.getCode() == KeyCode.ENTER) {
                calculate();
            } else if (event.getCode() == KeyCode.BACK_SPACE) {
                backspace();
            } else {
                switch (event.getText()) {
                    case "+":
                    case "-":
                    case "*":
                    case "/":
                        setOperator(event.getText());
                        break;
                }
            }
        });

        primaryStage.setScene(scene);
        primaryStage.show();
    }

    private GridPane createButtonGrid() {
        GridPane grid = new GridPane();
        grid.setHgap(10);
        grid.setVgap(10);
        grid.setAlignment(Pos.CENTER);

        String[][] buttons = {
            {"7", "8", "9", "/"},
            {"4", "5", "6", "*"},
            {"1", "2", "3", "-"},
            {"0", ".", "=", "+"},
            {"C"}
        };

        for (int row = 0; row < buttons.length; row++) {
            for (int col = 0; col < buttons[row].length; col++) {
                String text = buttons[row][col];
                Button button = new Button(text);
                button.setPrefSize(60, 60);
                button.setStyle("-fx-font-size: 16px;");
                button.setOnAction(e -> handleButtonClick(text));
                grid.add(button, col, row);
            }
        }
        return grid;
    }

    private void handleButtonClick(String value) {
        switch (value) {
            case "C":
                clear();
                break;
            case "=":
                calculate();
                break;
            case "+":
            case "-":
            case "*":
            case "/":
                setOperator(value);
                break;
            default:
                appendToDisplay(value);
                break;
        }
    }

    private void appendToDisplay(String value) {
        if (startNewNumber) {
            currentInput = "";
            startNewNumber = false;
        }
        currentInput += value;
        display.setText(currentInput);
    }

    private void setOperator(String op) {
        try {
            firstNumber = Double.parseDouble(currentInput);
            operator = op;
            startNewNumber = true;
        } catch (NumberFormatException e) {
            display.setText("Error");
        }
    }

    private void calculate() {
        try {
            double secondNumber = Double.parseDouble(currentInput);
            double result = 0;
            switch (operator) {
                case "+": result = firstNumber + secondNumber; break;
                case "-": result = firstNumber - secondNumber; break;
                case "*": result = firstNumber * secondNumber; break;
```

```java
                case "/":
                    if (secondNumber == 0) {
                        display.setText("Cannot divide by zero");
                        return;
                    }
                    result = firstNumber / secondNumber;
                    break;
                default:
                    return;
            }
            currentInput = String.valueOf(result);
            display.setText(currentInput);
            startNewNumber = true;
        } catch (NumberFormatException e) {
            display.setText("Error");
        }
    }

    private void clear() {
        currentInput = "";
        operator = "";
        firstNumber = 0;
        display.setText("");
        startNewNumber = true;
    }

    private void backspace() {
        if (!startNewNumber && currentInput.length() > 0) {
            currentInput = currentInput.substring(0, currentInput.length() - 1);
            display.setText(currentInput);
        }
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```