Scenario 1: Temperature Converter Utility

Problem Statement: You are developing a temperature conversion utility in Java. Implement temperature conversion logic using a method-local inner class to encapsulate the conversion formulas.

Requirements:

1. **Conversion Method:**
   - Define a method convertTemperature in a TemperatureConverter class that converts temperatures between Celsius and Fahrenheit.
2. **Method-Local Inner Class:**
   - Implement a method-local inner class Converter within convertTemperature.
   - Converter should encapsulate methods to convert Celsius to Fahrenheit and vice versa using the formulas:
     - Celsius to Fahrenheit: F=95×C+32F = \frac{9}{5} \times C + 32F=59 ×C+32
     - Fahrenheit to Celsius: C=59×(F−32)C = \frac{5}{9} \times (F - 32)C=95 ×(F−32)
3. **Testing:**
   - Instantiate TemperatureConverter in a main class and call convertTemperature with sample temperatures.
   - Verify that Converter correctly converts temperatures between Celsius and Fahrenheit based on the provided values.

Scenario 2: Simple Calculator Operations

Problem Statement: You are developing a simple calculator program in Java. Implement basic arithmetic operations using a method-local inner class to handle calculation logic.

Requirements:

1. **Calculator Method:**
   - Define a method performOperation in a Calculator class that performs addition, subtraction, multiplication, and division.
2. **Method-Local Inner Class:**
   - Implement a method-local inner class OperationHandler within performOperation.
   - OperationHandler should encapsulate methods to perform each arithmetic operation (addition, subtraction, multiplication, division).
3. **Testing:**
   - Instantiate Calculator in a main class and call performOperation with operands and operations.
   - Verify that OperationHandler correctly performs arithmetic operations and returns results based on the provided inputs.

Scenario 3: Text Processing Utility

Problem Statement: You are developing a text processing utility in Java. Implement text manipulation logic using a method-local inner class to handle formatting tasks.

Requirements:

1. **Text Processing Method:**
   - Define a method `processText` in a `TextProcessor` class that manipulates text such as trimming whitespace, converting to uppercase, and counting characters.
2. **Method-Local Inner Class:**
   - Implement a method-local inner class `TextManipulator` within `processText`.
   - `TextManipulator` should encapsulate methods to perform tasks like trimming whitespace, converting text to uppercase, and counting characters.
3. **Testing:**
   - Instantiate `TextProcessor` in a main class and call `processText` with sample text inputs.
   - Verify that `TextManipulator` correctly processes the text and returns results based on the specified operations.