

# Introduction

# Namespace TestProjectSampurna

## Classes

### [CircleCommandTest](#)

Contains unit tests for the [CircleCommand](#) class. These tests verify correct parameter parsing, validation, and execution behaviour when drawing circles on the canvas.

### [ClearCommandTest](#)

Unit tests for the [ClearCommand](#) class. Ensures the command validates parameters correctly and triggers a canvas clear operation when executed.

### [DrawToCommandTest](#)

Contains unit tests for the [DrawToCommand](#) class. These tests verify correct parameter parsing, validation, and execution behaviour when drawing lines on the canvas.

### [IfsAndLoopstest](#)

End-to-end unit tests for IF / IF-ELSE / WHILE / FOR commands. These tests execute real BOOSE programs using ExtendedParser and validate behaviour via canvas output.

### [MethodAndCallCommandTests](#)

Contains end-to-end unit tests for BOOSE method definition and invocation.

### [MockCanvas](#)

A mock implementation of BOOSE.ICanvas used for unit testing. It records method calls, argument values, and internal drawing state without performing any real graphical operations.

### [MoveToCommandTest](#)

Contains unit tests for the [MoveToCommand](#) class. These tests verify correct parameter parsing, validation, and execution behaviour when moving the canvas cursor.

### [MultilineProgramTest](#)

Provides unit tests for executing multiline BOOSE programs. Verifies drawing commands, colour changes, pen commands, and conditional or sequential execution using MockCanvas.

### [PenCommandTest](#)

Contains unit tests for the [PenCommand](#) class. These tests verify correct parameter parsing, validation, and execution behaviour when setting the pen colour.

### [RectangleCommandTest](#)

Contains unit tests for the [RectangleCommand](#) class. These tests verify correct parameter parsing, validation, and execution behaviour when drawing rectangles on the canvas.

### [ResetCommandTest](#)

Unit tests for the [resetCommand](#) class. Ensures correct validation, compilation behavior, and execution effects on both the stored program and the canvas.

### [TriangleCommandTest](#)

### [VariableTest](#)

Minimal variable facility tests required for: Int, Real, Boolean, Array, Poke, Peek.

### [WriteCommandTest](#)

Contains unit tests for the [WriteCommand](#) class. These tests verify correct parameter validation and execution behaviour when writing text to the canvas.