Namespace BOOSEapp

Classes

AppCanvas

Implements the BOOSE ICanvas interface for drawing operations Handles all graphic operations like drawing shapes, lines, and managing colours This is the actual canvas that the BOOSE system uses for rendering

CommandException

Custom exceprion for command related errors in the custom BOOSE interpreter Used when commands have problems with parameters or execution

Form1

Main application window for the BOOSE drawing interpreter Handles UI, user input, and coordinates between parser and canvas

MoveToCommand

Handles the moveto command, it moves the cursor wuthout drawing anything Usage: moveto 200,100

MyCommandFactory

Creates command objects based on command names This is the "command creator" is knows how to make each type of command

MyParser

MyStoredProgram

Custom program storage that holds and executes drawing commands This is where all the commands live before they get executed

Interfaces

ICommandHandler

Interface all command classes must implement Defines the basic structre for every drawing command

Namespace BOOSEapp.commands

Classes

CircleCommand

Handles the circle command - It draws a circle at current position Usage: circle 50

DrawToCommand

Handles the drawto command - draws a line from current position to a new position Usage: drawto 110,150

PenColourCommand

Handles the pen command - changes the pen colour Usage: pen 255,0,0 (for Red colour)

RectCommand

Handles the rect command - draws a rectangle at current position Usage: rect 50,100

Namespace BOOSEUnitTests

Classes

CommandFactoryTests

Unit test for MyCommandFactory to verify it creates command objects based on command types correctly

CommandsTests

Unit tests for individual command classes to verify successful execution and parameters are handled properly

IntegrationTests

Integration tests to verify multiple components work together correctly in complete system workflows

ParserTests

Unit tests for MyParser to verify it correctly pares the program text into executable commands