ASE_Assignment Namespace

⌄ Classes

AppCanvas	Provides functionality for drawing shapes, text, and graphics on a virtual canvas.
AppCommandFactory	Factory class for creating command objects based on the specified command type.
AppMethod	Represents a custom method handling in the BOOSE framework.
AppParser	Responsible for parsing program text into executable commands for the canvas.
ArrayCommand	Represents a command for handling array operations on the canvas. Inherits from the base Array class provided by the BOOSE framework.
CircleCommand	Represents a command to draw a circle on a canvas.
ClearCommand	Represents a command to clear the canvas.
DrawToCommand	Represents a command to draw a line from the current position to a specified coordinate on a canvas.
ElseCommand	Represents a command for handling the `Else` branch of an `If-Else` structure.
EndCommand	Represents a command for handling the end of a process or operation.
ForCommand	Represents a command for handling `For` loop functionality.
Form1	Represents the main form for the ASE Assignment application.
IfCommand	Represents a command that handles conditional logic, specifically implementing the functionality of an "if" statement. This command is designed to work with the BOOSE framework and provides an override of restrictions for greater flexibility.

IntCommand	Represents a command for handling integer-related operations on the canvas. Inherits from the base Int class provided by the BOOSE framework.
MoveToCommand	Represents a command to move the pen to a specified position on the canvas without drawing.
PenColourCommand	Represents a command to change the pen color on the canvas.
RealCommand	Represents a command for handling real number (floating-point) operations on the canvas. Inherits from the base Real class provided by the BOOSE framework.
RectCommand	Represents a command to draw a rectangle on the canvas.
ResetCommand	Represents a command to reset the canvas to its initial state.
TriCommand	Represents a command to draw a triangle on the canvas.
WhileCommand	Represents a command for handling `While` loop functionality.
WriteCommand	Represents a command to write text onto the canvas.