

Namespace BOOSEUnitTesting

Classes

[DrawToCommandTesting](#)

[MoveToCommandTests](#)

[MultiLineProgramTest](#)

Class DrawToCommandTesting

Namespace: [BOOSEUnitTesting](#)

Assembly: BOOSEUnitTesting.dll

```
[TestClass]
public class DrawToCommandTesting
```

Inheritance

[object](#) ← DrawToCommandTesting

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Methods

CheckParametersIncorrectAmount()

```
[TestMethod]
[ExpectedException(typeof(ParserException))]
public void CheckParametersIncorrectAmount()
```

CheckParametersInvalidXInput()

```
[TestMethod]
public void CheckParametersInvalidXInput()
```

CheckParametersInvalidXInputType()

```
[TestMethod]
[ExpectedException(typeof(ParserException))]
public void CheckParametersInvalidXInputType()
```

CheckParametersInvalidYInput()

```
[TestMethod]
public void CheckParametersInvalidYInput()
```

CheckParametersInvalidYInputType()

```
[TestMethod]
[ExpectedException(typeof(ParserException))]
public void CheckParametersInvalidYInputType()
```

CheckParametersNothingEntered()

```
[TestMethod]
[ExpectedException(typeof(ParserException))]
public void CheckParametersNothingEntered()
```

CheckParametersValidInputs()

```
public void CheckParametersValidInputs()
```

Execute_OutOfBounds_DoesNotCrash()

```
[TestMethod]
public void Execute_OutOfBounds_DoesNotCrash()
```

Execute_ValidCoordinates_UpdatesCanvasPosition()

```
[TestMethod]
public void Execute_ValidCoordinates_UpdatesCanvasPosition()
```

SetValidParameters()

```
[TestMethod]
public void SetValidParameters()
```

Set_EmptyParams.ThrowsCommandException()

```
[TestMethod]
[ExpectedException(typeof(ParserException))]
public void Set_EmptyParams.ThrowsCommandException()
```

Set_InvalidParamCount.ThrowsCommandException()

```
[TestMethod]
[ExpectedException(typeof(ParserException))]
public void Set_InvalidParamCount.ThrowsCommandException()
```

Setup()

```
[TestInitialize]
public void Setup()
```

Class MoveToCommandTests

Namespace: [BOOSEUnitTesting](#)

Assembly: BOOSEUnitTesting.dll

```
[TestClass]
public sealed class MoveToCommandTests
```

Inheritance

[object](#) ← MoveToCommandTests

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Methods

CheckParametersIncorrectAmount()

```
[TestMethod]
[ExpectedException(typeof(ParserException))]
public void CheckParametersIncorrectAmount()
```

CheckParametersInvalidXInput()

```
[TestMethod]
public void CheckParametersInvalidXInput()
```

CheckParametersInvalidXInputType()

```
[TestMethod]
[ExpectedException(typeof(ParserException))]
public void CheckParametersInvalidXInputType()
```

CheckParametersInvalidYInput()

```
[TestMethod]
public void CheckParametersInvalidYInput()
```

CheckParametersInvalidYInputType()

```
[TestMethod]
[ExpectedException(typeof(ParserException))]
public void CheckParametersInvalidYInputType()
```

CheckParametersNothingEntered()

```
[TestMethod]
[ExpectedException(typeof(ParserException))]
public void CheckParametersNothingEntered()
```

CheckParametersValidInputs()

```
[TestMethod]
public void CheckParametersValidInputs()
```

Execute_OutOfBounds_DoesNotCrash()

```
[TestMethod]
public void Execute_OutOfBounds_DoesNotCrash()
```

Execute_ValidCoordinates_UpdatesCanvasPosition()

```
[TestMethod]
public void Execute_ValidCoordinates_UpdatesCanvasPosition()
```

SetValidParameters()

```
[TestMethod]
public void SetValidParameters()
```

Set_EmptyParams.ThrowsCommandException()

```
[TestMethod]
[ExpectedException(typeof(ParserException))]
public void Set_EmptyParams.ThrowsCommandException()
```

Set_InvalidParamCount.ThrowsCommandException()

```
[TestMethod]
[ExpectedException(typeof(ParserException))]
public void Set_InvalidParamCount.ThrowsCommandException()
```

Setup()

```
[TestInitialize]
public void Setup()
```

Class MultiLineProgramTest

Namespace: [BOOSEUnitTesting](#)

Assembly: BOOSEUnitTesting.dll

```
[TestClass]
public sealed class MultiLineProgramTest
```

Inheritance

[object](#) ← MultiLineProgramTest

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Methods

InvalidMultilineProgram1()

```
[TestMethod]
public void InvalidMultilineProgram1()
```

Setup()

```
[TestInitialize]
public void Setup()
```

ValidMultilineProgram1()

```
[TestMethod]
public void ValidMultilineProgram1()
```

Namespace BOOSE_App

Classes

[AppCanvas](#)

Class used for a drawable canvas via underlying bitmap. Supports basic drawing operations which are built upon in other classes.

[AppCommandFactory](#)

Factory class responsible for creating command objects for BOOSE application. Commands are linked to [AppCanvas](#) allowing drawing operations to be used avoiding their inherited restrictions.

[CircleCommand](#)

Command class for drawing circles onto canvas at current pen position. Bypasses BOOSE restrictions.

[CommandParser](#)

Parses user input into valid BOOSE commands. Works with command factory to create command objects and stores them in BOOSE.StoredProgram for execution.

[DrawToCommand](#)

Command class for drawing a line on canvas from current pen position.

[Form1](#)

[Form1](#) operates as the main class for the application. It provides a windows form GUI to store and execute user commands, which interact with the drawable canvas.

[MoveToCommand](#)

Command class for moving pen on canvas without drawing.

[PenCommand](#)

[RectangleCommand](#)

Command class for drawing rectangles onto canvas at current pen position.

[WriteCommand](#)

Command class for drawing text onto canvas at current pen position. Uses BOOSE.ICommand interface

Class AppCanvas

Namespace: [BOOSE_App](#)

Assembly: BOOSE_App.dll

Class used for a drawable canvas via underlying bitmap. Supports basic drawing operations which are built upon in other classes.

```
public class AppCanvas : ICanvas
```

Inheritance

[object](#) ← AppCanvas

Implements

ICanvas

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

AppCanvas(int, int)

Constructor for [AppCanvas](#) which sets up size parameters, pen properties and bitmap to canvas.

```
public AppCanvas(int width, int height)
```

Parameters

width [int](#)

The width of the canvas.

height [int](#)

The height of the canvas.

Fields

background_colour

```
protected Color background_colour
```

Field Value

[Color↗](#)

height

```
public int height
```

Field Value

[int↗](#)

width

```
public int width
```

Field Value

[int↗](#)

xpos

```
public int xpos
```

Field Value

[int↗](#)

ypos

```
public int ypos
```

Field Value

[int↗](#)

Properties

PenColour

```
public virtual object PenColour { get; set; }
```

Property Value

[object↗](#)

Xpos

Returns X position of pen on canvas.

Sets X position of pen on canvas. Catches BOOSE.CanvasException if value isn't within canvas boundaries.

```
public int Xpos { get; set; }
```

Property Value

[int↗](#)

Ypos

Returns Y position of pen on canvas.

Sets Y position of pen on canvas. Catches BOOSE.CanvasException if value isn't within canvas boundaries.

```
public int Ypos { get; set; }
```

Property Value

[int](#)

Methods

Circle(int, bool)

Draws a circle at current pen position.

```
public void Circle(int radius, bool filled)
```

Parameters

[radius](#) [int](#)

Radius of circle to be drawn.

[filled](#) [bool](#)

Determines whether circle is filled or not. Currently not implemented.

Remarks

Updates the pen position after drawing.

Clear()

Clears canvas.

```
public virtual void Clear()
```

DrawTo(int, int)

Draws a line on canvas from current pen position to desired x and y co-ordinates. Throws BOOSE.CanvasException if parameter isn't within canvas boundaries.

```
public void DrawTo(int x, int y)
```

Parameters

x [int](#)

X co-ordinate value of Pen.

y [int](#)

Y co-ordinate value of Pen.

Remarks

Updates the pen position to the specified co-ordinates after drawing.

MoveTo(int, int)

Moves pen to desired x and y co-ordinates without drawing. Throws BOOSE.CanvasException if parameter isn't within canvas boundaries.

```
public void MoveTo(int x, int y)
```

Parameters

x [int](#)

X co-ordinate value of Pen.

y [int](#)

Y co-ordinate value of Pen.

Rect(int, int, bool)

Draws rectangle at the current pen position.

```
public void Rect(int width, int height, bool filled)
```

Parameters

width [int](#)

Width of rectangle.

height [int](#)

Height of rectangle.

filled [bool](#)

Determines whether rectangle is filled or not. Currently not implemented.

Remarks

Updates the pen position after drawing.

Reset()

Resets pen position to default values (0,0).

```
public virtual void Reset()
```

Set(int, int)

Resets graphic context for existing bitmap.

```
public virtual void Set(int width, int height)
```

Parameters

width [int](#)

Width of canvas.

height [int](#)

Width of bitmap.

SetColour(int, int, int)

Sets colour of pen via RGB values. Throws BOOSE.CanvasException if a value is above 255.

```
public void SetColour(int red, int green, int blue)
```

Parameters

red [int](#)

Red colour value.

green [int](#)

Green colour value.

blue [int](#)

Blue colour value.

Tri(int, int)

Draws triangle at current pen position with left corner as base point.

```
public virtual void Tri(int width, int height)
```

Parameters

width [int](#)

Width of triangle.

height [int](#)

Height of triangle.

Remarks

Triangle is drawn with three points the current position, the top point and the base right point. Updates pen position after drawing.

WriteText(string)

Draws text at current pen position.

```
public virtual void WriteText(string text)
```

Parameters

text [string](#)

String to be drawn on canvas.

Remarks

Updates the pen positions after drawing.

draw(Graphics)

Method had to be implemented due to interface. But is not in use as of right now.

```
public void draw(Graphics g)
```

Parameters

g [Graphics](#)

Exceptions

[NotImplementedException](#)

getBitmap()

Returns bitmap.

```
public virtual object getBitmap()
```

Returns

object ↗

Bitmap which is used for drawing operations.

Class AppCommandFactory

Namespace: [BOOSE_App](#)

Assembly: BOOSE_App.dll

Factory class responsible for creating command objects for BOOSE application. Commands are linked to [AppCanvas](#) allowing drawing operations to be used avoiding their inherited restrictions.

```
public class AppCommandFactory : CommandFactory, ICommandFactory
```

Inheritance

[object](#) ← CommandFactory ← AppCommandFactory

Implements

ICommandFactory

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

AppCommandFactory(AppCanvas)

Constructor for [AppCommandFactory](#) setting up canvas for commands.

```
public AppCommandFactory(AppCanvas canvas)
```

Parameters

canvas [AppCanvas](#)

The canvas which commands will operate on.

Fields

canvas

Canvas object that commands use.

```
public AppCanvas canvas
```

Field Value

[AppCanvas](#)

Methods

MakeCommand(string)

Creates command objects based on provided name.

```
public override ICommand MakeCommand(string commandType)
```

Parameters

commandType [string](#) ↗

The name of the command to but be created.

Returns

ICommand

Command for execution.

Exceptions

FactoryException

Thrown when the command entered doesn't exist.

Class CircleCommand

Namespace: [BOOSE_App](#)

Assembly: BOOSE_App.dll

Command class for drawing circles onto canvas at current pen position. Bypasses BOOSE restrictions.

```
public class CircleCommand : ICommand
```

Inheritance

[object](#) ← CircleCommand

Implements

ICommand

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

CircleCommand(AppCanvas)

Constructor for command setting up canvas for drawing.

```
public CircleCommand(AppCanvas canvas)
```

Parameters

canvas [AppCanvas](#)

Fields

canvas

Canvas that will be drawn onto.

```
public AppCanvas canvas
```

Field Value

[AppCanvas](#)

filled

Determines whether circle will be filled.

```
public bool filled
```

Field Value

[bool](#)

program

Stored command from parser for execution.

```
public StoredProgram program
```

Field Value

StoredProgram

radius

The radius of the circle to be drawn.

```
public int radius
```

Field Value

[int](#)

Methods

CheckParameters(string[])

Checks command parameters are valid.

```
public void CheckParameters(string[] Parameters)
```

Parameters

Parameters [string](#)[]

Array containing command parameters.

Exceptions

ParserException

Thrown if there is not one parameter stored as expected or the radius parameter is not a valid integer.

CommandException

Thrown if radius value is below one making it invalid.

Compile()

Method not implemented.

```
public void Compile()
```

Execute()

Executes circle command.

```
public void Execute()
```

Remarks

Calls circle method from canvas using parsed parameter. Ignores restrictions implemented by BOOSE.

Set(StoredProgram, string)

Sets command parameters.

```
public void Set(StoredProgram Program, string Params)
```

Parameters

Program StoredProgram

Reference for the command stored by the parser.

Params string ↗

Reference for command parameters stored by the parser.

Class CommandParser

Namespace: [BOOSE App](#)

Assembly: BOOSE_App.dll

Parses user input into valid BOOSE commands. Works with command factory to create command objects and stores them in BOOSE.StoredProgram for execution.

```
public class CommandParser : IParser
```

Inheritance

[object](#) ← CommandParser

Implements

IParser

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

CommandParser(AppCommandFactory, StoredProgram)

Constructor for [CommandParser](#) that initialises command factory and stored program.

```
public CommandParser(AppCommandFactory factory, StoredProgram program)
```

Parameters

factory [AppCommandFactory](#)

Factory used to create commands.

program StoredProgram

Program used to store commands.

Fields

factory

Factory used to create command objects based on parsed user input.

```
public AppCommandFactory factory
```

Field Value

[AppCommandFactory](#)

storedProgram

Stores command objects so they can be executed.

```
public StoredProgram storedProgram
```

Field Value

StoredProgram

Methods

ParseCommand(string)

Parses a single line of user input into a command object.

```
public ICommand ParseCommand(string Line)
```

Parameters

Line [string](#) ↗

Raw user input string.

Returns

ICommand

A compiled BOOSE.ICommand object containing command and command parameters for execution.

Exceptions

ParserException

Class DrawToCommand

Namespace: [BOOSE_App](#)

Assembly: BOOSE_App.dll

Command class for drawing a line on canvas from current pen position.

```
public class DrawToCommand : ICommand
```

Inheritance

[object](#) ← DrawToCommand

Implements

ICommand

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

DrawToCommand(AppCanvas)

Constructor for command that sets up canvas for drawing.

```
public DrawToCommand(AppCanvas canvas)
```

Parameters

canvas [AppCanvas](#)

Fields

canvas

Canvas to be drawn onto.

```
public AppCanvas canvas
```

Field Value

[AppCanvas](#)

program

Command stored by parser.

```
public StoredProgram program
```

Field Value

StoredProgram

X

x and y co-ordinates of pen position on canvas.

```
public int x
```

Field Value

[int](#)

y

x and y co-ordinates of pen position on canvas.

```
public int y
```

Field Value

[int](#)

Methods

CheckParameters(string[])

```
public void CheckParameters(string[] Parameters)
```

Parameters

Parameters `string[]`

Compile()

Method not implemented.

```
public void Compile()
```

Execute()

Executes drawto command.

```
public void Execute()
```

Remarks

Calls drawto method from canvas using command parameters.

Set(StoredProgram, string)

Sets command parameters.

```
public void Set(StoredProgram Program, string Params)
```

Parameters

Program `StoredProgram`

Reference for the command stored by the parser.

Params [string](#)

Reference for command parameters stored by the parser.

Class Form1

Namespace: [BOOSE_App](#)

Assembly: BOOSE_App.dll

[Form1](#) operates as the main class for the application. It provides a windows form GUI to store and execute user commands, which interact with the drawable canvas.

```
public class Form1 : Form, IDropTarget, ISynchronizeInvoke, IWin32Window,  
IBindableComponent, IComponent, IDisposable, IContainerControl
```

Inheritance

```
object ↳ ← MarshalByRefObject ↳ ← Component ↳ ← Control ↳ ← ScrollableControl ↳ ←  
ContainerControl ↳ ← Form ↳ ← Form1
```

Implements

```
IDropTarget, ISynchronizeInvoke, IWin32Window, IBindableComponent, IComponent,  
IDisposable, IContainerControl
```

Inherited Members

```
Form.SetVisibleCore(bool), Form.Activate(), Form.ActivateMdiChild(Form),  
Form.AddOwnedForm(Form), Form.AdjustFormScrollbars(bool), Form.Close(),  
Form.CreateAccessibilityInstance(), Form.CreateControlsInstance(), Form.CreateHandle(),  
Form.DefWndProc(ref Message), Form.ProcessMnemonic(char), Form.CenterToParent(),  
Form.CenterToScreen(), Form.LayoutMdi(MdiLayout), Form.OnActivated(EventArgs),  
Form.OnBackgroundImageChanged(EventArgs),  
Form.OnBackgroundImageLayoutChanged(EventArgs), Form.OnClosing(CancelEventArgs),  
Form.OnClosed(EventArgs), Form.OnFormClosing(FormClosingEventArgs),  
Form.OnFormClosed(FormClosedEventArgs), Form.OnCreateControl(),  
Form.OnDeactivate(EventArgs), Form.OnEnabledChanged(EventArgs), Form.OnEnter(EventArgs),  
Form.OnFontChanged(EventArgs), Form.OnGotFocus(EventArgs),  
Form.OnHandleCreated(EventArgs), Form.OnHandleDestroyed(EventArgs),  
Form.OnHelpButtonClicked(CancelEventArgs), Form.OnLayout(LayoutEventArgs),  
Form.OnLoad(EventArgs), Form.OnMaximizedBoundsChanged(EventArgs),  
Form.OnMaximumSizeChanged(EventArgs), Form.OnMinimumSizeChanged(EventArgs),  
Form.OnInputLanguageChanged(InputLanguageChangedEventArgs),  
Form.OnInputLanguageChanging(InputLanguageChangingEventArgs),  
Form.OnVisibleChanged(EventArgs), Form.OnMdiChildActivate(EventArgs),  
Form.OnMenuStart(EventArgs), Form.OnMenuComplete(EventArgs),  
Form.OnPaint(PaintEventArgs), Form.OnResize(EventArgs),
```

[Form.OnDpiChanged\(DpiChangedEventArgs\)](#) , [Form.OnGetDpiScaledSize\(int, int, ref Size\)](#) ,
[Form.OnRightToLeftLayoutChanged\(EventArgs\)](#) , [Form.OnShown\(EventArgs\)](#) ,
[Form.OnTextChanged\(EventArgs\)](#) , [Form.ProcessCmdKey\(ref Message, Keys\)](#) ,
[Form.ProcessDialogKey\(Keys\)](#) , [Form.ProcessDialogChar\(char\)](#) ,
[Form.ProcessKeyPreview\(ref Message\)](#) , [Form.ProcessTabKey\(bool\)](#) ,
[Form.RemoveOwnedForm\(Form\)](#) , [Form.Select\(bool, bool\)](#) ,
[Form.ScaleMinAxisSize\(float, float, bool\)](#) ,
[Form.GetScaledBounds\(Rectangle, SizeF, BoundsSpecified\)](#) ,
[Form.ScaleControl\(SizeF, BoundsSpecified\)](#) , [Form.SetBoundsCore\(int, int, int, int, BoundsSpecified\)](#) ,
[Form.SetClientSizeCore\(int, int\)](#) , [Form.SetDesktopBounds\(int, int, int, int\)](#) ,
[Form.SetDesktopLocation\(int, int\)](#) , [Form.Show\(IWin32Window\)](#) , [Form.ShowDialog\(\)](#) ,
[Form.ShowDialog\(IWin32Window\)](#) , [Form.ToString\(\)](#) , [Form.UpdateDefaultButton\(\)](#) ,
[Form.OnResizeBegin\(EventArgs\)](#) , [Form.OnResizeEnd\(EventArgs\)](#) ,
[Form.OnStyleChanged\(EventArgs\)](#) , [Form.ValidateChildren\(\)](#) ,
[Form.ValidateChildren\(ValidationConstraints\)](#) , [Form.WndProc\(ref Message\)](#) , [Form.AcceptButton](#) ,
[Form.ActiveForm](#) , [Form.ActiveMdiChild](#) , [Form.AllowTransparency](#) , [Form.AutoScroll](#) ,
[Form.AutoSize](#) , [Form.AutoSizeMode](#) , [Form.AutoValidate](#) , [Form.BackColor](#) ,
[Form.FormBorderStyle](#) , [Form.CancelButton](#) , [Form.ClientSize](#) , [Form.ControlBox](#) ,
[Form.CreateParams](#) , [Form.DefaultImeMode](#) , [Form.DefaultSize](#) , [Form.DesktopBounds](#) ,
[Form/DesktopLocation](#) , [Form/DialogResult](#) , [Form/HelpButton](#) , [Form/Icon](#) , [Form/IsMdiChild](#) ,
[Form/IsMdiContainer](#) , [Form/IsRestrictedWindow](#) , [Form/KeyPreview](#) , [Form/Location](#) ,
[Form/MaximizedBounds](#) , [Form/MaximumSize](#) , [Form/MainMenuStrip](#) , [Form/MinimumSize](#) ,
[Form/MaximizeBox](#) , [Form/MdiChildren](#) , [Form/MdiChildrenMinimizedAnchorBottom](#) ,
[Form/MdiParent](#) , [Form/MinimizeBox](#) , [Form/Modal](#) , [Form/Opacity](#) , [Form/OwnedForms](#) ,
[Form/Owner](#) , [Form/RestoreBounds](#) , [Form/RightToLeftLayout](#) , [Form>ShowInTaskbar](#) ,
[Form>ShowIcon](#) , [Form>ShowWithoutActivation](#) , [Form/Size](#) , [Form/SizeGripStyle](#) ,
[Form/StartPosition](#) , [Form/Text](#) , [Form/TopLevel](#) , [Form/TopMost](#) , [Form/TransparencyKey](#) ,
[Form/WindowState](#) , [Form/AutoSizeChanged](#) , [Form/AutoValidateChanged](#) ,
[Form/HelpButtonClicked](#) , [Form/MaximizedBoundsChanged](#) , [Form/MaximumSizeChanged](#) ,
[Form/MinimumSizeChanged](#) , [Form/Activated](#) , [Form/Deactivate](#) , [Form/FormClosing](#) ,
[Form/FormClosed](#) , [Form/Load](#) , [Form/MdiChildActivate](#) , [Form/MenuComplete](#) ,
[Form/MenuStart](#) , [Form/InputLanguageChanged](#) , [Form/InputLanguageChanging](#) ,
[Form/RightToLeftLayoutChanged](#) , [Form/Shown](#) , [Form/DpiChanged](#) , [Form/ResizeBegin](#) ,
[Form/ResizeEnd](#) , [ContainerControl.OnAutoValidateChanged\(EventArgs\)](#) ,
[ContainerControl.OnMove\(EventArgs\)](#) , [ContainerControl.OnParentChanged\(EventArgs\)](#) ,
[ContainerControl.PerformLayout\(\)](#) , [ContainerControl.RescaleConstantsForDpi\(int, int\)](#) ,
[ContainerControl/Validate\(\)](#) , [ContainerControl/Validate\(bool\)](#) ,
[ContainerControl/AutoScaleDimensions](#) , [ContainerControl/AutoScaleFactor](#) ,
[ContainerControl/AutoScaleMode](#) , [ContainerControl/BindingContext](#) ,
[ContainerControl/CanEnableIme](#) , [ContainerControl/ActiveControl](#) ,

[ContainerControl.CurrentAutoScaleDimensions](#) , [ContainerControl.ParentForm](#) ,
[ScrollableControl.ScrollStateAutoScrolling](#) , [ScrollableControl.ScrollStateHScrollVisible](#) ,
[ScrollableControl.ScrollStateVScrollVisible](#) , [ScrollableControl.ScrollStateUserHasScrolled](#) ,
[ScrollableControl.ScrollStateFullDrag](#) , [ScrollableControl.GetScrollState\(int\)](#) ,
[ScrollableControl.OnMouseWheel\(MouseEventArgs\)](#) ,
[ScrollableControl.OnRightToLeftChanged\(EventArgs\)](#) ,
[ScrollableControl.OnPaintBackground\(PaintEventArgs\)](#) ,
[ScrollableControl.OnPaddingChanged\(EventArgs\)](#) , [ScrollableControl.SetDisplayRectLocation\(int, int\)](#) ,
[ScrollableControl.ScrollControlIntoView\(Control\)](#) , [ScrollableControl.ScrollToControl\(Control\)](#) ,
[ScrollableControl.OnScroll\(ScrollEventArgs\)](#) , [ScrollableControl.SetAutoScrollMargin\(int, int\)](#) ,
[ScrollableControl.SetScrollState\(int, bool\)](#) , [ScrollableControl.AutoScrollMargin](#) ,
[ScrollableControl.AutoScrollPosition](#) , [ScrollableControl.AutoScrollMinSize](#) ,
[ScrollableControl.DisplayRectangle](#) , [ScrollableControl.HScroll](#) , [ScrollableControl.HorizontalScroll](#) ,
[ScrollableControl.VScroll](#) , [ScrollableControl.VerticalScroll](#) , [ScrollableControl.Scroll](#) ,
[Control.GetAccessibilityObjectById\(int\)](#) , [Control.SetAutoSizeMode\(AutoSizeMode\)](#) ,
[Control.GetAutoSizeMode\(\)](#) , [Control.GetPreferredSize\(Size\)](#) ,
[Control.AccessibilityNotifyClients\(AccessibleEvents, int\)](#) ,
[Control.AccessibilityNotifyClients\(AccessibleEvents, int, int\)](#) , [Control.BeginInvoke\(Delegate\)](#) ,
[Control.BeginInvoke\(Action\)](#) , [Control.BeginInvoke\(Delegate, params object\[\]\)](#) ,
[Control.BringToFront\(\)](#) , [Control.Contains\(Control\)](#) , [Control.CreateGraphics\(\)](#) ,
[Control.CreateControl\(\)](#) , [Control.DestroyHandle\(\)](#) , [Control.DoDragDrop\(object, DragDropEffects\)](#) ,
[Control.DoDragDrop\(object, DragDropEffects, Bitmap, Point, bool\)](#) ,
[Control.DrawToBitmap\(Bitmap, Rectangle\)](#) , [Control.EndInvoke\(IAsyncResult\)](#) , [Control.FindForm\(\)](#) ,
[Control.GetTopLevel\(\)](#) , [Control.RaiseKeyEvent\(object, KeyEventArgs\)](#) ,
[Control.RaiseMouseEvent\(object, MouseEventArgs\)](#) , [Control.Focus\(\)](#) ,
[Control.FromChildHandle\(nint\)](#) , [Control.FromHandle\(nint\)](#) ,
[Control.GetChildAtPoint\(Point, GetChildAtPointSkip\)](#) , [Control.GetChildAtPoint\(Point\)](#) ,
[Control.GetContainerControl\(\)](#) , [Control.GetNextControl\(Control, bool\)](#) ,
[Control.GetStyle\(ControlStyles\)](#) , [Control.Hide\(\)](#) , [Control.InitLayout\(\)](#) , [Control.Invalidate\(Region\)](#) ,
[Control.Invalidate\(Region, bool\)](#) , [Control.Invalidate\(\)](#) , [Control.Invalidate\(bool\)](#) ,
[Control.Invalidate\(Rectangle\)](#) , [Control.Invalidate\(Rectangle, bool\)](#) , [Control.Invoke\(Action\)](#) ,
[Control.Invoke\(Delegate\)](#) , [Control.Invoke\(Delegate, params object\[\]\)](#) ,
[Control.Invoke<T>\(Func<T>\)](#) , [Control.InvokePaint\(Control, PaintEventArgs\)](#) ,
[Control.InvokePaintBackground\(Control, PaintEventArgs\)](#) , [Control.IsKeyLocked\(Keys\)](#) ,
[Control.IsInputChar\(char\)](#) , [Control.IsInputKey\(Keys\)](#) , [Control.IsMnemonic\(char, string\)](#) ,
[Control.LogicalToDeviceUnits\(int\)](#) , [Control.LogicalToDeviceUnits\(Size\)](#) ,
[Control.ScaleBitmapLogicalToDevice\(ref Bitmap\)](#) , [Control.NotifyInvalidate\(Rectangle\)](#) ,
[Control.InvokeOnClick\(Control, EventArgs\)](#) , [Control.OnAutoSizeChanged\(EventArgs\)](#) ,
[Control.OnBackColorChanged\(EventArgs\)](#) , [Control.OnBindingContextChanged\(EventArgs\)](#) ,
[Control.OnCausesValidationChanged\(EventArgs\)](#) , [Control.OnContextMenuStripChanged\(EventArgs\)](#) ,

[Control.OnCursorChanged\(EventArgs\)](#) , [Control.OnDataContextChanged\(EventArgs\)](#) ,
[Control.OnDockChanged\(EventArgs\)](#) , [Control.OnForeColorChanged\(EventArgs\)](#) ,
[Control.OnNotifyMessage\(Message\)](#) , [Control.OnParentBackColorChanged\(EventArgs\)](#) ,
[Control.OnParentBackgroundImageChanged\(EventArgs\)](#) ,
[Control.OnParentBindingContextChanged\(EventArgs\)](#) , [Control.OnParentCursorChanged\(EventArgs\)](#) ,
[Control.OnParentDataContextChanged\(EventArgs\)](#) , [Control.OnParentEnabledChanged\(EventArgs\)](#) ,
[Control.OnParentFontChanged\(EventArgs\)](#) , [Control.OnParentForeColorChanged\(EventArgs\)](#) ,
[Control.OnParentRightToLeftChanged\(EventArgs\)](#) , [Control.OnParentVisibleChanged\(EventArgs\)](#) ,
[Control.OnPrint\(PaintEventArgs\)](#) , [Control.OnTabIndexChanged\(EventArgs\)](#) ,
[Control.OnTabStopChanged\(EventArgs\)](#) , [Control.OnClick\(EventArgs\)](#) ,
[Control.OnClientSizeChanged\(EventArgs\)](#) , [Control.OnControlAdded\(ControlEventArgs\)](#) ,
[Control.OnControlRemoved\(ControlEventArgs\)](#) , [Control.OnLocationChanged\(EventArgs\)](#) ,
[Control.OnDoubleClick\(EventArgs\)](#) , [Control.OnDragEnter\(DragEventArgs\)](#) ,
[Control.OnDragOver\(DragEventArgs\)](#) , [Control.OnDragLeave\(EventArgs\)](#) ,
[Control.OnDragDrop\(DragEventArgs\)](#) , [Control.OnGiveFeedback\(GiveFeedbackEventArgs\)](#) ,
[Control.InvokeGotFocus\(Control, EventArgs\)](#) , [Control.OnHelpRequested\(HelpEventArgs\)](#) ,
[Control.OnInvalidate\(InvalidateEventArgs\)](#) , [Control.OnKeyDown\(KeyEventEventArgs\)](#) ,
[Control.OnKeyPress\(KeyPressEventEventArgs\)](#) , [Control.OnKeyUp\(KeyEventEventArgs\)](#) ,
[Control.OnLeave\(EventArgs\)](#) , [Control.InvokeLostFocus\(Control, EventArgs\)](#) ,
[Control.OnLostFocus\(EventArgs\)](#) , [Control.OnMarginChanged\(EventArgs\)](#) ,
[Control.OnMouseDoubleClick\(MouseEventArgs\)](#) , [Control.OnMouseClicked\(MouseEventArgs\)](#) ,
[Control.OnMouseCaptureChanged\(EventArgs\)](#) , [Control.OnMouseDown\(MouseEventArgs\)](#) ,
[Control.OnMouseEnter\(EventArgs\)](#) , [Control.OnMouseLeave\(EventArgs\)](#) ,
[Control.OnDpiChangedBeforeParent\(EventArgs\)](#) , [Control.OnDpiChangedAfterParent\(EventArgs\)](#) ,
[Control.OnMouseHover\(EventArgs\)](#) , [Control.OnMouseMove\(MouseEventArgs\)](#) ,
[Control.OnMouseUp\(MouseEventArgs\)](#) ,
[Control.OnQueryContinueDrag\(QueryContinueDragEventArgs\)](#) ,
[Control.OnRegionChanged\(EventArgs\)](#) , [Control.OnPreviewKeyDown\(PreviewKeyDownEventArgs\)](#) ,
[Control.OnSizeChanged\(EventArgs\)](#) , [Control.OnChangeUICues\(UICuesEventArgs\)](#) ,
[Control.OnSystemColorsChanged\(EventArgs\)](#) , [Control.OnValidating\(CancelEventArgs\)](#) ,
[Control.OnValidated\(EventArgs\)](#) , [Control.PerformLayout\(\)](#) , [Control.PerformLayout\(Control, string\)](#) ,
[Control.PointToClient\(Point\)](#) , [Control.PointToScreen\(Point\)](#) ,
[Control.PreProcessMessage\(ref Message\)](#) , [Control.PreProcessControlMessage\(ref Message\)](#) ,
[Control.ProcessKeyEventArgs\(ref Message\)](#) , [Control.ProcessKeyMessage\(ref Message\)](#) ,
[Control.RaiseDragEvent\(object, DragEventArgs\)](#) , [Control.RaisePaintEvent\(object, PaintEventArgs\)](#) ,
[Control.RecreateHandle\(\)](#) , [Control.RectangleToClient\(Rectangle\)](#) ,
[Control.RectangleToScreen\(Rectangle\)](#) , [Control.ReflectMessage\(nint, ref Message\)](#) ,
[Control.Refresh\(\)](#) , [Control.ResetMouseEventArgs\(\)](#) , [Control.ResetText\(\)](#) , [Control.ResumeLayout\(\)](#) ,
[Control.ResumeLayout\(bool\)](#) , [Control.Scale\(SizeF\)](#) , [Control.Select\(\)](#) ,
[Control.SelectNextControl\(Control, bool, bool, bool\)](#) , [Control.SendToBack\(\)](#) ,

[Control.SetBounds\(int, int, int, int\)](#) , [Control.SetBounds\(int, int, int, int, BoundsSpecified\)](#) ,
[Control.SizeFromClientSize\(Size\)](#) , [Control.SetStyle\(ControlStyles, bool\)](#) , [Control.SetTopLevel\(bool\)](#) ,
[Control.RtlTranslateAlignment\(HorizontalAlignment\)](#) ,
[Control.RtlTranslateAlignment\(LeftRightAlignment\)](#) ,
[Control.RtlTranslateAlignment\(ContentAlignment\)](#) ,
[Control.RtlTranslateHorizontal\(HorizontalAlignment\)](#) ,
[Control.RtlTranslateLeftRight\(LeftRightAlignment\)](#) , [Control.RtlTranslateContent\(ContentAlignment\)](#) ,
[Control.Show\(\)](#) , [Control.SuspendLayout\(\)](#) , [Control.Update\(\)](#) , [Control.UpdateBounds\(\)](#) ,
[Control.UpdateBounds\(int, int, int, int\)](#) , [Control.UpdateBounds\(int, int, int, int, int, int\)](#) ,
[Control.UpdateZOrder\(\)](#) , [Control.UpdateStyles\(\)](#) , [Control.OnImeModeChanged\(EventArgs\)](#) ,
[Control.AccessibilityObject](#) , [Control.AccessibleDefaultActionDescription](#) ,
[Control.AccessibleDescription](#) , [Control.AccessibleName](#) , [Control.AccessibleRole](#) ,
[Control.AllowDrop](#) , [Control.Anchor](#) , [Control.AutoScrollOffset](#) , [Control.LayoutEngine](#) ,
[Control.DataContext](#) , [Control.BackgroundImage](#) , [Control.BackgroundImageLayout](#) ,
[Control.Bottom](#) , [Control.Bounds](#) , [Control.CanFocus](#) , [Control.CanRaiseEvents](#) ,
[Control.CanSelect](#) , [Control.Capture](#) , [Control.CausesValidation](#) ,
[Control.CheckForIllegalCrossThreadCalls](#) , [Control.ClientRectangle](#) , [Control.CompanyName](#) ,
[Control.ContainsFocus](#) , [Control.ContextMenuStrip](#) , [Control.Controls](#) , [Control.Created](#) ,
[Control.Cursor](#) , [Control.DataBindings](#) , [Control.DefaultBackColor](#) , [Control.DefaultCursor](#) ,
[Control.DefaultFont](#) , [Control.DefaultForeColor](#) , [Control.DefaultMargin](#) ,
[Control.DefaultMaximumSize](#) , [Control.DefaultMinimumSize](#) , [Control.DefaultPadding](#) ,
[Control.DeviceDpi](#) , [Control.IsDisposed](#) , [Control.Disposing](#) , [Control.Dock](#) ,
[Control.DoubleBuffered](#) , [Control.Enabled](#) , [Control.Focused](#) , [Control.Font](#) ,
[Control.FontHeight](#) , [Control.ForeColor](#) , [Control.Handle](#) , [Control.HasChildren](#) , [Control.Height](#) ,
[Control.IsHandleCreated](#) , [Control.InvokeRequired](#) , [Control.Accessible](#) ,
[Control.IsAncestorSiteInDesignMode](#) , [Control.IsMirrored](#) , [Control.Left](#) , [Control.Margin](#) ,
[Control.ModifierKeys](#) , [Control.MouseButtons](#) , [Control.mousePosition](#) , [Control.Name](#) ,
[Control.Parent](#) , [Control.ProductName](#) , [Control.ProductVersion](#) , [Control.RecreatingHandle](#) ,
[Control.Region](#) , [Control.RenderRightToLeft](#) , [Control.ResizeRedraw](#) , [Control.Right](#) ,
[Control.RightToLeft](#) , [Control.ScaleChildren](#) , [Control.Site](#) , [Control.TabIndex](#) , [Control.TabStop](#) ,
[Control.Tag](#) , [Control.Top](#) , [Control.TopLevelControl](#) , [Control.ShowKeyboardCues](#) ,
[Control.ShowFocusCues](#) , [Control.UseWaitCursor](#) , [Control.Visible](#) , [Control.Width](#) ,
[Control.PreferredSize](#) , [Control.Padding](#) , [Control.ImeMode](#) , [Control.ImeModeBase](#) ,
[Control.PropagatingImeMode](#) , [Control.BackColorChanged](#) , [Control.BackgroundImageChanged](#) ,
[Control.BackgroundImageLayoutChanged](#) , [Control.BindingContextChanged](#) ,
[Control.CausesValidationChanged](#) , [Control.ClientSizeChanged](#) ,
[Control.ContextMenuStripChanged](#) , [Control.CursorChanged](#) , [Control.DockChanged](#) ,
[Control.EnabledChanged](#) , [Control.FontChanged](#) , [Control.ForeColorChanged](#) ,
[Control.LocationChanged](#) , [Control.MarginChanged](#) , [Control.RegionChanged](#) ,
[Control.RightToLeftChanged](#) , [Control.SizeChanged](#) , [Control.TabIndexChanged](#) ,

[Control.TabStopChanged](#) , [Control.TextChanged](#) , [Control.VisibleChanged](#) , [Control.Click](#) ,
[Control.ControlAdded](#) , [Control.ControlRemoved](#) , [Control.DataContextChanged](#) ,
[Control.DragDrop](#) , [Control.DragEnter](#) , [Control.DragOver](#) , [Control.DragLeave](#) ,
[Control.GiveFeedback](#) , [Control.HandleCreated](#) , [Control.HandleDestroyed](#) ,
[Control.HelpRequested](#) , [Control.Invalidated](#) , [Control.PaddingChanged](#) , [Control.Paint](#) ,
[Control.QueryContinueDrag](#) , [Control.QueryAccessibilityHelp](#) , [Control.DoubleClick](#) ,
[Control.Enter](#) , [Control.GotFocus](#) , [Control.KeyDown](#) , [Control.KeyPress](#) , [Control.KeyUp](#) ,
[Control.Layout](#) , [Control.Leave](#) , [Control.LostFocus](#) , [Control.MouseClick](#) ,
[Control.MouseDoubleClick](#) , [Control.MouseCaptureChanged](#) , [Control.MouseDown](#) ,
[Control.MouseEnter](#) , [Control.MouseLeave](#) , [Control.DpiChangedBeforeParent](#) ,
[Control.DpiChangedAfterParent](#) , [Control.MouseHover](#) , [Control.MouseMove](#) , [Control.MouseUp](#) ,
[Control.MouseWheel](#) , [Control.Move](#) , [Control.PreviewKeyDown](#) , [Control.Resize](#) ,
[Control.ChangeUICues](#) , [Control.StyleChanged](#) , [Control.SystemColorsChanged](#) ,
[Control.Validating](#) , [Control.Validated](#) , [Control.ParentChanged](#) , [Control.ImeModeChanged](#) ,
[Component.Dispose\(\)](#) , [Component.GetService\(Type\)](#) , [Component.Container](#) ,
[Component.DesignMode](#) , [Component.Events](#) , [Component.Disposed](#) ,
[MarshalByRefObject.GetLifetimeService\(\)](#) , [MarshalByRefObject.InitializeLifetimeService\(\)](#) ,
[MarshalByRefObject.MemberwiseClone\(bool\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#)

Constructors

Form1()

Initialises windows GUI form and GUI components. Sets up required BOOSE components and confirms BOOSE is working.

```
public Form1()
```

Methods

Dispose(bool)

Clean up any resources being used.

```
protected override void Dispose(bool disposing)
```

Parameters

disposing [bool](#)

true if managed resources should be disposed; otherwise, false.

Class MoveToCommand

Namespace: [BOOSE_App](#)

Assembly: BOOSE_App.dll

Command class for moving pen on canvas without drawing.

```
public class MoveToCommand : ICommand
```

Inheritance

[object](#) ← MoveToCommand

Implements

ICommand

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

MoveToCommand(AppCanvas)

Constructor for moveto command sets up canvas.

```
public MoveToCommand(AppCanvas canvas)
```

Parameters

canvas [AppCanvas](#)

Fields

canvas

Canvas to be operated on.

```
public AppCanvas canvas
```

Field Value

[AppCanvas](#)

program

Command stored by parser.

```
public StoredProgram program
```

Field Value

StoredProgram

X

x and y co-ordinates of pen position on canvas.

```
public int x
```

Field Value

[int](#)

y

x and y co-ordinates of pen position on canvas.

```
public int y
```

Field Value

[int](#)

Methods

CheckParameters(string[])

Checks command parameters are valid.

```
public void CheckParameters(string[] Parameters)
```

Parameters

Parameters [string](#)[]

An array containing command parameters

Exceptions

ParserException

Thrown if there is not two parameters stored as required. Also thrown if the parameters stored are not integers.

Compile()

Method not implemented.

```
public void Compile()
```

Execute()

Executes moveto command.

```
public void Execute()
```

Remarks

Calls moveto method from canvas using command parameters.

Set(StoredProgram, string)

Sets command parameters.

```
public void Set(StoredProgram Program, string Params)
```

Parameters

Program StoredProgram

Reference for the command stored by the parser.

Params string ↗

Reference for command parameters stored by the parser.

Class PenCommand

Namespace: [BOOSE App](#)

Assembly: BOOSE_App.dll

```
public class PenCommand : ICommand
```

Inheritance

[object](#) ← PenCommand

Implements

ICommand

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

PenCommand(AppCanvas)

```
public PenCommand(AppCanvas canvas)
```

Parameters

canvas [AppCanvas](#)

Fields

blue

```
public int blue
```

Field Value

[int](#)

canvas

```
public AppCanvas canvas
```

Field Value

[AppCanvas](#)

green

```
public int green
```

Field Value

[int↗](#)

program

```
public StoredProgram program
```

Field Value

StoredProgram

red

```
public int red
```

Field Value

[int↗](#)

Methods

CheckParameters(string[])

Checks command parameters are valid.

```
public void CheckParameters(string[] Parameters)
```

Parameters

Parameters [string](#)[]

An array of command parameters.

Exceptions

ParserException

Thrown if there is not three parameters stored as required. Also thrown if parameters are not integers.

CommandException

Thrown when command parameters values are below 0 or above 255 making them invalid.

Compile()

```
public void Compile()
```

Execute()

Executes pen command.

```
public void Execute()
```

Remarks

Calls setcolour method from canvas using parsed parameter.

Set(StoredProgram, string)

Sets command parameters.

```
public void Set(StoredProgram Program, string Params)
```

Parameters

Program StoredProgram

Reference for the command stored by the parser.

Params string ↗

Reference for command parameters stored by the parser.

Class RectangleCommand

Namespace: [BOOSE_App](#)

Assembly: BOOSE_App.dll

Command class for drawing rectangles onto canvas at current pen position.

```
public class RectangleCommand : ICommand
```

Inheritance

[object](#) ← RectangleCommand

Implements

ICommand

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

RectangleCommand(AppCanvas)

Constructor for command that sets up canvas for drawing.

```
public RectangleCommand(AppCanvas canvas)
```

Parameters

canvas [AppCanvas](#)

The canvas to be drawn onto.

Fields

canvas

Canvas that will be drawn onto.

```
public AppCanvas canvas
```

Field Value

[AppCanvas](#)

filled

Determines whether rectangle is filled or not

```
public bool filled
```

Field Value

[bool](#)

height

Height and width parameters of rectangle to be drawn.

```
public int height
```

Field Value

[int](#)

program

Stored command from parser for execution.

```
public StoredProgram program
```

Field Value

width

Height and width parameters of rectangle to be drawn.

```
public int width
```

Field Value

[int](#)

Methods

CheckParameters(string[])

Checks command parameters are valid.

```
public void CheckParameters(string[] Parameters)
```

Parameters

Parameters [string](#)[]

Array containing command parameters.

Exceptions

ParserException

Thrown if there is not two parameters as required. Also thrown if parameter values are not integers.

CommandException

Thrown if height or width parameters are invalid integer that can't be drawn.

Compile()

Method not implemented.

```
public void Compile()
```

Execute()

Executes rectangle command.

```
public void Execute()
```

Remarks

Calls rect method from canvas using parsed parameters. Ignores restrictions implemented by BOOSE.

Set(StoredProgram, string)

Sets command parameters.

```
public void Set(StoredProgram Program, string Params)
```

Parameters

Program StoredProgram

Reference for the command stored by the parser.

Params string ↗

Reference for command parameters stored by the parser.

Class WriteCommand

Namespace: [BOOSE App](#)

Assembly: BOOSE_App.dll

Command class for drawing text onto canvas at current pen position. Uses BOOSE.ICommand interface

```
public class WriteCommand : ICommand
```

Inheritance

[object](#) ← WriteCommand

Implements

ICommand

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

WriteCommand(AppCanvas)

Constructor for command that sets up canvas for drawing.

```
public WriteCommand(AppCanvas canvas)
```

Parameters

canvas [AppCanvas](#)

The canvas to be drawn onto.

Fields

canvas

Canvas that will be drawn onto.

```
public AppCanvas canvas
```

Field Value

[AppCanvas](#)

program

Stored command from parser for execution.

```
public StoredProgram program
```

Field Value

StoredProgram

text

The string to be written onto the canvas.

```
public string text
```

Field Value

[string](#)

Methods

CheckParameters(string[])

Checks command parameters are valid.

```
public void CheckParameters(string[] Parameters)
```

Parameters

Parameters `string[]`

Array containing command parameters.

Exceptions

ParserException

Thrown if there is not one parameter as required.

Compile()

Method not implemented.

```
public void Compile()
```

Execute()

Executes writeText command.

```
public void Execute()
```

Remarks

Calls write text method from canvas using parsed parameter. Ignores restrictions implemented by BOOSE.

Set(StoredProgram, string)

Sets command parameters.

```
public void Set(StoredProgram Program, string Params)
```

Parameters

Program StoredProgram

Reference for the command stored by the parser.

Params [string](#) ↗

Reference for command parameters stored by the parser.