Namespace ASEUnitTests

Classes

ASEUnitTests

This is the Unit testing class to test the functionalities of the program.

Class ASEUnitTests

Namespace: ASEUnitTests

Assembly: ASEUnitTests.dll

This is the Unit testing class to test the functionalities of the program.

```
[TestClass]
public class ASEUnitTests
```

Inheritance

object ← ASEUnitTests

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Methods

DrawTo_PenPosition_CorrectlyStored()

This test method is essentially identical to the workings of the MoveTo_PenPosition_Correctly stored test except this method tests the drawto command.

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

MoveTo_PenPosition_CorrectlyStored()

This test method implements a mock up of the program, and provides some values to be tested. Then the would-be command, which in this case is a move to command, is executed with the values given and finally this is compared to the actual position of the pen on the mockup canvas to ensure the command was properly parsed and executed.

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

MultilineProgram_CorrectlyExecuted()

This test method implements a slightly more robust mock up of the program in order to properly test a multi-line program, as such it also implements the BOOSE library classes; Parser, CommandFactory & StoredProgram, in order to do this and sets a test program with multiple commands, the test program is the parsed and ran to ensure that the pen ends up in the correct end coordinates meaning every command was successfully ran.

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

Namespace ASE_Assignment

Classes

ClearCanvas

This class is uses the BOOSE Command class to implement a custom command for use in the CustomCommandFactory class this particular class checks the how many parameters are in the parsed command and throws an exception if there is more/less than one parameter and then calls & executes the Clear method of the myCanvas class.

CustomCommandFactory

This class implements the original BOOSE CommandFactory class but adds the methods required that the original class does not contain or cannot handle. It contains two references; the canvas & the form. Most methods only use the canvas reference to use the canvas on the form. The form reference is used only by the 'fill' method as the toggle for the fill function is

FillToggle

This class is uses the BOOSE Command class to implement a custom command for use in the CustomCommandFactory class this particular class checks the how many parameters are in the parsed command and throws an exception if there is more or less than one parameter, then it changes a boolean value within the myCanvas class which affects many drawing classes when they are called. It also calls the fillToggleIndicator method on the main for to show the changes within the UI

ResetCanvas

This class is uses the BOOSE Command class to implement a custom command for use in the CustomCommandFactory class this particular class checks the how many parameters are in the parsed command and throws an exception if there is more or less than one parameter, it then calls & executes the Reset method in the myCanvas class.

Triangle

This class is uses the BOOSE Command class to implement a custom command for use in the CustomCommandFactory class this particular class checks the how many parameters are in the parsed command and throws an exception if there is more/less than two parameters and then assigns the different parameter values to a variable and then passes said values through to the Tri class in the myCanvas class

mainForm

This class handles the UI of the program and the events occuring when a user interacts with the program.

myCanvas

myWriteText

Class ClearCanvas

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

This class is uses the BOOSE Command class to implement a custom command for use in the CustomCommandFactory class this particular class checks the how many parameters are in the parsed command and throws an exception if there is more/less than one parameter and then calls & executes the Clear method of the myCanvas class.

```
public class ClearCanvas : Command, ICommand
```

Inheritance

object ← Command ← ClearCanvas

Implements

ICommand

Inherited Members

Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , Command.Compile() , Command.ProcessParameters(string) , Command.ToString() , Command.Program , Command.Name , Command.ParameterList , Command.Parameters , Command.Paramsint , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

ClearCanvas(myCanvas)

References the myCanvas class to allow access to its methods.

```
public ClearCanvas(myCanvas canvas)
```

Parameters

`canvas` myCanvas

Uses 'canvas' to reference the myCanvas class

Methods

CheckParameters(string[])

Checks the entered parameters and throws exception when they are out of permitted bounds.

```
public override void CheckParameters(string[] parameter)
```

Parameters

`parameter` string[]

This is the string of user entered values

Exceptions

CommandException

Thrown when the user enters too few or too many parameters

Execute()

Executes the command by calling the Clear method from the myCanvas class.

```
public override void Execute()
```

Class CustomCommandFactory

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

This class implements the original BOOSE CommandFactory class but adds the methods required that the original class does not contain or cannot handle. It contains two references; the canvas & the form. Most methods only use the canvas reference to use the canvas on the form. The form reference is used only by the 'fill' method as the toggle for the fill function is

```
public class CustomCommandFactory : CommandFactory, ICommandFactory
```

Inheritance

object ← CommandFactory ← CustomCommandFactory

Implements

ICommandFactory

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

CustomCommandFactory(myCanvas, mainForm)

References the myCanvas & mainForm classes to allow access to their methods.

```
public CustomCommandFactory(myCanvas canvas, mainForm form)
```

Parameters

`canvas` myCanvas

Uses 'canvas' to reference the myCanvas class

`form` mainForm

Uses 'form' to reference the mainForm class

Methods

MakeCommand(string)

Overrides the BOOSE CommandFactory to check whether the entered command is in this class before checking the original if nothing is found here

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

Parameters

`commandType` string

User inputted command.

Returns

ICommand

Returns to the original CommandFactory to continue searching for the correct command

Class FillToggle

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

This class is uses the BOOSE Command class to implement a custom command for use in the CustomCommandFactory class this particular class checks the how many parameters are in the parsed command and throws an exception if there is more or less than one parameter, then it changes a boolean value within the myCanvas class which affects many drawing classes when they are called. It also calls the fillToggleIndicator method on the main for to show the changes within the UI

```
public class FillToggle : Command, ICommand
```

Inheritance

object ← Command ← FillToggle

Implements

ICommand

Inherited Members

Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , Command.Compile() , Command.ProcessParameters(string) , Command.ToString() , Command.Program , Command.Name , Command.ParameterList , Command.Parameters , Command.Paramsint , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

FillToggle(myCanvas, mainForm)

References the myCanvas & mainForm classes to allow access to their methods.

```
public FillToggle(myCanvas canvas, mainForm form)
```

Parameters

`canvas` myCanvas

Uses 'canvas' to reference the myCanvas class

`form` mainForm

Uses 'form' to reference the mainForm class

Methods

CheckParameters(string[])

Checks the entered parameters and throws exception when they are out of permitted bounds.

```
public override void CheckParameters(string[] parameter)
```

Parameters

`parameter` string[]

This is the string of user entered values

Exceptions

CommandException

Thrown when the user enters too few or too many parameters

Execute()

Toggles the bool 'fillToggle' on the myCanvas class between true and false allowing the filling of shapes in the canvas class also calls the fillToggleIndicator in the myCanvas class to change the associated UI element

```
public override void Execute()
```

Class ResetCanvas

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

This class is uses the BOOSE Command class to implement a custom command for use in the CustomCommandFactory class this particular class checks the how many parameters are in the parsed command and throws an exception if there is more or less than one parameter, it then calls & executes the Reset method in the myCanvas class.

```
public class ResetCanvas : Command, ICommand
```

Inheritance

object ← Command ← ResetCanvas

Implements

ICommand

Inherited Members

Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , Command.Compile() , Command.ProcessParameters(string) , Command.ToString() , Command.Program , Command.Name , Command.ParameterList , Command.Parameters , Command.Paramsint , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

ResetCanvas(myCanvas)

References the myCanvas class to allow access to its methods.

```
public ResetCanvas(myCanvas canvas)
```

Parameters

`canvas` myCanvas

Uses 'canvas' to reference the myCanvas class

Methods

CheckParameters(string[])

Checks the entered parameters and throws exception when they are out of permitted bounds.

```
public override void CheckParameters(string[] parameter)
```

Parameters

`parameter` string[]

This is the string of user entered values

Exceptions

CommandException

Thrown when the user enters too few or too many parameters

Execute()

Executes the command by calling the Reset method from the myCanvas class.

```
public override void Execute()
```

Class Triangle

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

This class is uses the BOOSE Command class to implement a custom command for use in the CustomCommandFactory class this particular class checks the how many parameters are in the parsed command and throws an exception if there is more/less than two parameters and then assigns the different parameter values to a variable and then passes said values through to the Tri class in the myCanvas class

```
public class Triangle : Command, ICommand
```

Inheritance

object ← Command ← Triangle

Implements

ICommand

Inherited Members

Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , Command.Compile() , Command.ProcessParameters(string) , Command.ToString() , Command.Program , Command.Name , Command.ParameterList , Command.Parameters , Command.Paramsint , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

Triangle(myCanvas)

References the myCanvas class to allow access to its methods.

```
public Triangle(myCanvas canvas)
```

Parameters

`canvas` myCanvas

Uses 'canvas' to reference the myCanvas class

Methods

CheckParameters(string[])

Checks the entered parameters and throws exception when they are out of permitted bounds.

```
public override void CheckParameters(string[] parameter)
```

Parameters

`parameter` string[]

This is the string of user entered values

Exceptions

CommandException

Thrown when the user enters too few or too many parameters

Execute()

Sets the values of the parameters parsed into variables and then calls the Tri method from the myCanvas class

```
public override void Execute()
```

Class mainForm

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

This class handles the UI of the program and the events occuring when a user interacts with the program.

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

Inheritance

object ← MarshalByRefObject ← Component ← Control ← ScrollableControl ← ContainerControl ← Form ← mainForm

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.ScaleMinMaxSize(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.PerformAutoScale() , 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.OnInvalidated(InvalidateEventArgs) , Control.OnKeyDown(KeyEventArgs) , Control.OnKeyPress(KeyPressEventArgs) , Control.OnKeyUp(KeyEventArgs) , Control.OnLeave(EventArgs) , Control.InvokeLostFocus(Control, EventArgs) , Control.OnLostFocus(EventArgs) , Control.OnMarginChanged(EventArgs) , Control.OnMouseDoubleClick(MouseEventArgs) , Control.OnMouseClick(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, 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.IsAccessible ,
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

mainForm()

This method initialises all of the required aspects for the program to run correctly and displays a graphic, version number etc of the BOOSE library in the console

```
public mainForm()
```

Fields

DarkMode

```
public bool DarkMode
```

Field Value

bool

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.

currentPenColourIndicator()

This is similar to the previous method used to control the colour of a panel except this method is used to change the colour of a panel to the current colour of the pen in use, this provides an easy visual feedback for the user to keep track of the current colour as opposed to memorising RGB codes and also gives a preview of the colour

```
public void currentPenColourIndicator()
```

fillToggleIndicator()

This method is used to control the panel which is used to provide a visual indicator to the user whether the fill function is currently enabled or not. It simply checks the status of the bool 'fillToggle' located in the myCanvas class and changes the colour of the panel depending on this when it is called.

```
public void fillToggleIndicator()
```

Class myCanvas

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

```
public class myCanvas : ICanvas
```

Inheritance

object ← myCanvas

Implements

ICanvas

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

myCanvas(mainForm)

Initialises the canvas size, pen size and initial colour of the pen, also references the mainForm class to allow its methods to be accessed.

```
public myCanvas(mainForm form)
```

Parameters

`form` mainForm

Fields

fillToggle

```
public bool fillToggle
```

Field Value

bool

Properties

PenColour

Handles the storing of colour values.

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

Property Value

object

Xpos

Handles the storing of X coordinate values.

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

Property Value

int

Ypos

Handles the storing of Y coordinate values.

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

Property Value

int

Methods

Circle(int, bool)

Handles the drawing of circles onto the canvas when called, checking whether the 'fillToggle' bool is true or false to decide whether the shape should be filled or not.

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

Parameters

`radius` int

This value dictates the size of the circle, being multiplied by 2 to make the diameter

`filled` bool

A bool to decide whether the circle should be filled or not

Clear()

Simply clears the canvas by flooding the canvas with a blank colour

```
public void Clear()
```

DrawTo(int, int)

Handles the drawing of lines between two desired points, and moves the current pen position to the new destination afterwards.

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

Parameters

`x` int

The X coordinates of where the user desires the end point of the line to be

`y` int

The Y coordinates of where the user desires the end point of the line to be

Exceptions

CanvasException

This exception is called when the line is outside of the bounds of the canvas

MoveTo(int, int)

Handles the moving of the pen to the users desired location.

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

Parameters

`x` int

The desired X coordinates of the pen

`y` int

The desired Y coordinates of the pen

Exceptions

CanvasException

This exception is thrown when the line is outside the bounds of the canvas

Rect(int, int, bool)

Handles the drawing of rectangles onto the canvas

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

Parameters

`width` int

The desired width of the rectangle

`height` int

The desired height of the rectangle

`filled` bool

Whether the rectangle should be filled or not

Exceptions

CanvasException

This is thrown when the user tries to make a rectangle with no width or height value, or values exceeding the canvas size

Reset()

This method essentially resets the app back to default values; resetting coordinates, disposing of pens, disabling the fill option and setting the pen colour to black

```
public void Reset()
```

Set(int, int)

Used when the program first starts in order to set the size of the canvas, the bitmap that will be used as a canvas and the initial position of the pen.

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

Parameters

`width` int

Width of the canvas

`height` int

Height of the canvas

SetColour(int, int, int)

Handles the changing of the colour of the pen used to draw lines and 'brush' used to draw and fill shapes, also calls the currentPenColourIndicator in the mainForm class in order to update a UI element

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

Parameters

`red` int

The red value of the pen, using typical RGB limits of 0-255

`green` int

The green value of the pen, using typical RGB limits of 0-255

`blue` int

The blue value of the pen, using typical RGB limits of 0-255

Exceptions

CanvasException

Thrown when a value outside of the RGB limits is attempted to be entered

Tri(int, int)

Handles the drawing of triangles onto the canvas

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

Parameters

`width` int

The desired width of the triangle

`height` int

The desired height of the triangle

Exceptions

CanvasException

Thrown when user attempts to make a triangle with 0 values or one that exceeds the canvas size

WriteText(string)

Handles the writing of text onto the canvas

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

Parameters

`text` string

The user input which will be written onto the canvas

getBitmap()

Returns the bitmap used for drawing

```
public object getBitmap()
```

Returns

object

The bitmap used as a canvas

Class myWriteText

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

```
public class myWriteText : Command, ICommand
```

Inheritance

object ← Command ← myWriteText

Implements

ICommand

Inherited Members

Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , Command.Compile() , Command.ProcessParameters(string) , Command.ToString() , Command.Program , Command.Name , Command.ParameterList , Command.Parameters , Command.Paramsint , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

myWriteText(myCanvas)

References the myCanvas class to allow access to its methods.

```
public myWriteText(myCanvas canvas)
```

Parameters

`canvas` myCanvas

Uses 'canvas' to reference the myCanvas class

Methods

CheckParameters(string[])

Checks the entered parameters and throws exception when they are out of permitted bounds.

```
public override void CheckParameters(string[] parameter)
```

Parameters

`parameter` string[]

This is the string of user entered values

Exceptions

CommandException

Thrown when the user enters too few or too many parameters

Execute()

Executes the command by ensuring there is text to be drawn and if so, it calls the WriteText method from the myCanvas class if not, an exception is thrown

```
public override void Execute()
```

Exceptions

CommandException

Thrown if there is no text to be drawn by the program