Namespace Medium_Scale_Software_Engineering_Project

Classes

AppCanvas
: Represents a drawable canvas that supports basic drawing operations such as lines, shapes, and text.

AppCommandFactory
: Implement CommandFactory with support for all BOOSE commands

Form1

MethodDefinition
: Represents a method definition

MyCircle
: MyCircle command - draws a circle with given radius, optionally filled.

MyClear
: MyClear command - clears the canvas.

MyDrawTo
: DrawTo command - draws a line from current position to specified (x,y) coordinates.

MyForCommand
: MyForCommand class - represents a custom 'for' command.

MyIfCommand
: Implements the IF command with support for else and nested if statements

MyIntCommand
: Implements the INT variable declaration and assignment command

MyMoveTo
: MyMoveTo command - moves the current position to specified (x,y) coordinates without drawing.

MyPen
: Pen command - sets the drawing pen color using RGB values.

MyRect
: MyRect command - draws a rectangle with specified width and height, optionally filled.

MyReset

MyReset command - resets the canvas and clears stored commands.

MyTri

MyTri command - draws a triangle with specified width and height.

MyWhileCommand

Implements the WHILE loop command

MyWrite

myWrite command - writes text on the canvas at the current pen position.

Program

UnknownCommand

Command that does nothing for unknown commands Prevents exceptions during parsing

VariableStore

Singleton class that replaces BOOSE internal variable handling for Int, Real, Array, and Method types with unlimited capacity.

Interfaces

IVariableStore

Abstraction for variable storage and evaluation. Implemented by VariableStore.

Class AppCanvas

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

Represents a drawable canvas that supports basic drawing operations such as lines, shapes, and text.

```
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)

AppCanvas constructor initializes the canvas with specified width and height.

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

Parameters

`width` int

`height` int

Properties

PenColour

Property to get or set the pen colour used for drawing on the canvas.

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

Property Value

object

Xpos

Gets or sets the current X position of the pen on the canvas.

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

Property Value

int

Ypos

Gets or sets the current Y position of the pen on the canvas.

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

Property Value

int

Methods

Circle(int, bool)

Creates a circle at the current pen position with the specified radius.

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

Parameters

`radius` int

`filled` bool

Clear()

Clears the canvas by filling it with black color.

```
public void Clear()
```

DrawCursorDot()

Draws a white dot with black outline at the current pen position. This should be called separately to draw the cursor dot on top of everything.

```
public void DrawCursorDot()
```

DrawTo(int, int)

Draws a line from the current pen position to the specified (x, y) coordinates.

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

Parameters

`x` int

`y` int

MoveTo(int, int)

Moves the pen to the specified (x, y) coordinates without drawing.

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

Parameters

`x` int

`y` int

Rect(int, int, bool)

Renders a rectangle at the current pen position with the specified width and height.

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

Parameters

`width` int

`height` int

`filled` bool

Reset()

Resets the canvas to its initial state.

```
public void Reset()
```

Set(int, int)

Sets the size of the canvas to the specified width and height.

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

Parameters

`width` int

`height` int

SetColour(int, int, int)

Sets the current drawing colour using RGB values.

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

Parameters

`red` int

`green` int

`blue` int

Tri(int, int)

Draws a triangle at the current pen position with the specified width and height.

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

Parameters

`width` int

`height` int

WriteText(string)

Writes the specified text at the current pen position. This is called by the "write" command in BOOSE.

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

Parameters

`text` string

getBitmap()

Gets the underlying bitmap of the canvas for rendering.

```
public object getBitmap()
```

Returns

object

Class AppCommandFactory

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

Implement CommandFactory with support for all BOOSE commands

```
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()

Constructor for AppCommandFactory.

```
public AppCommandFactory()
```

AppCommandFactory(AppCanvas)

Constructor for AppCommandFactory with AppCanvas parameter.

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

Parameters

`canvas` AppCanvas

Methods

MakeCommand(string)

Make a command, showing message boxes on errors

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

Parameters

`name` string

Returns

ICommand

Class Form1

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

```
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.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

Form1()

```
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.

Interface IVariableStore

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

Abstraction for variable storage and evaluation. Implemented by VariableStore.

```
public interface IVariableStore
```

Properties

CurrentMethod

Current method name being executed.

```
string CurrentMethod { get; }
```

Property Value

string

Methods

Clear()

Clears all variables and methods from the store.

```
void Clear()
```

DefineMethod(string, MethodDefinition)

Defines a new method.

```
void DefineMethod(string name, MethodDefinition definition)
```

Parameters

`name` string

`definition` MethodDefinition

EnterMethod(string)

Enters a method context for tracking current method.

```
void EnterMethod(string methodName)
```

Parameters

`methodName` string

EvaluateExpression(string)

Evaluates an expression string and returns the result.

```
object EvaluateExpression(string expression)
```

Parameters

`expression` string

Returns

object

ExitMethod()

Exits the current method context.

```
void ExitMethod()
```

GetMethod(string)

Gets a method definition by name.

```
MethodDefinition GetMethod(string name)
```

Parameters

`name` string

Returns

MethodDefinition

GetVariable(string)

Gets a variable by name, searching from the current scope outward.

```
object GetVariable(string name)
```

Parameters

`name` string

Returns

object

MethodExists(string)

Method existence check.

```
bool MethodExists(string name)
```

Parameters

`name` string

Returns

bool

PopScope()

Pops the current variable scope from the stack.

```
void PopScope()
```

PushScope()

Pushes a new variable scope onto the stack.

```
void PushScope()
```

SetVariable(string, object, string)

Sets a variable in the current scope.

```
void SetVariable(string name, object value, string type = null)
```

Parameters

`name` string

`value` object

`type` string

VariableExists(string)

Verifies if a variable exists in any scope.

```
bool VariableExists(string name)
```

Parameters

`name` string

Returns

bool

Class MethodDefinition

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

Represents a method definition

```
public class MethodDefinition
```

Inheritance

object ← MethodDefinition

Inherited Members

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

Constructors

MethodDefinition()

```
public MethodDefinition()
```

Properties

Body

```
public List<string> Body { get; set; }
```

Property Value

List <string>

Name

```
public string Name { get; set; }
```

Property Value

string

Parameters

```
public List<string> Parameters { get; set; }
```

Property Value

List <string>

ReturnType

```
public string ReturnType { get; set; }
```

Property Value

string

Methods

ParseSignature()

```
public (string ShortName, List<(string Type, string Name)> Parameters) ParseSignature()
```

Returns

(string ShortName, List <(string Type, string Name)> Parameters)

Class MyCircle

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

MyCircle command - draws a circle with given radius, optionally filled.

```
public class MyCircle : CommandOneParameter, ICommand
```

Inheritance

object ← Command ← CanvasCommand ← CommandOneParameter ← MyCircle

Implements

ICommand

Inherited Members

CommandOneParameter.param1 , CommandOneParameter.param1unprocessed , CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas , Command.IsDouble , 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

MyCircle()

MyCircle constructor

```
public MyCircle()
```

MyCircle(Canvas, int)

MyCircle constructor with canvas and radius

```
public MyCircle(Canvas c, int radius)
```

Parameters

`c` Canvas

`radius` int

Methods

CheckParameters(string[])

Check parameters for MyCircle command

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

Parameters

`parameterList` string[]

Exceptions

CommandException

Execute()

Execute the MyCircle command

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

Exceptions

CanvasException

Class MyClear

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

MyClear command - clears the canvas.

```
public class MyClear : Command, ICommand
```

Inheritance

object ← Command ← MyClear

Implements

ICommand

Inherited Members

Command.IsDouble , 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

MyClear(AppCanvas)

MyClear constructor

```
public MyClear(AppCanvas canvas)
```

Parameters

`canvas` AppCanvas

Methods

CheckParameters(string[])

CheckParameters for MyClear command

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

Parameters

`parameter` string[]

Exceptions

NotImplementedException

Execute()

Execute the MyClear command

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

Class MyDrawTo

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

DrawTo command - draws a line from current position to specified (x,y) coordinates.

```
public class MyDrawTo : CommandTwoParameters, ICommand
```

Inheritance

object ← Command ← CanvasCommand ← CommandOneParameter ← CommandTwoParameters ← MyDrawTo

Implements

ICommand

Inherited Members

CommandTwoParameters.param2 , CommandTwoParameters.param2unprocessed , CommandOneParameter.param1 , CommandOneParameter.param1unprocessed , CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas , Command.IsDouble , 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

MyDrawTo()

MyDrawTo constructor

```
public MyDrawTo()
```

MyDrawTo(Canvas, int, int)

MyDrawTo constructor with canvas and coordinates

```
public MyDrawTo(Canvas c, int x, int y)
```

Parameters

`c` Canvas

`x` int

`y` int

Methods

CheckParameters(string[])

CheckParameters for MyDrawTo command

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

Parameters

`parameterList` string[]

Execute()

Execute the MyDrawTo command

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

Exceptions

CanvasException

Class MyForCommand

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

MyForCommand class - represents a custom 'for' command.

```
public class MyForCommand : Command, ICommand
```

Inheritance

object ← Command ← MyForCommand

Implements

ICommand

Inherited Members

Command.IsDouble , 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)

Methods

CheckParameters(string[])

CheckParameters for MyForCommand

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

Parameters

`parameters` string[]

Execute()

Execute the MyForCommand

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

Class MyIfCommand

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

Implements the IF command with support for else and nested if statements

```
public class MyIfCommand : Command, ICommand
```

Inheritance

object ← Command ← MyIfCommand

Implements

ICommand

Inherited Members

Command.IsDouble , 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

MyIfCommand()

MyIfCommand constructor

```
public MyIfCommand()
```

Methods

AddToElseBlock(string)

Add command to the ELSE block

```
public void AddToElseBlock(string command)
```

Parameters

`command` string

AddToIfBlock(string)

Add command to the IF block

```
public void AddToIfBlock(string command)
```

Parameters

`command` string

CheckParameters(string[])

CheckParameters for MyIfCommand

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

Parameters

`parameters` string[]

Exceptions

CommandException

Execute()

Execute the MyIfCommand

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

SetHasElse(bool)

Set whether the IF command has an ELSE block

```
public void SetHasElse(bool hasElse)
```

Parameters

`hasElse` bool

Class MyIntCommand

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

Implements the INT variable declaration and assignment command

```
public class MyIntCommand : Command, ICommand
```

Inheritance

object ← Command ← MyIntCommand

Implements

ICommand

Inherited Members

Command.IsDouble , 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

MyIntCommand()

MyIntCommand constructor

```
public MyIntCommand()
```

Methods

CheckParameters(string[])

CheckParameters for INT command

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

Parameters

`parameters` string[]

Exceptions

CommandException

Execute()

Execute the INT command

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

Exceptions

Exception

Class MyMoveTo

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

MyMoveTo command - moves the current position to specified (x,y) coordinates without drawing.

```
public class MyMoveTo : CommandTwoParameters, ICommand
```

Inheritance

object ← Command ← CanvasCommand ← CommandOneParameter ← CommandTwoParameters ← MyMoveTo

Implements

ICommand

Inherited Members

CommandTwoParameters.param2 , CommandTwoParameters.param2unprocessed , CommandOneParameter.param1 , CommandOneParameter.param1unprocessed , CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas , Command.IsDouble , 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

MyMoveTo()

MyMoveTo constructor

```
public MyMoveTo()
```

MyMoveTo(Canvas, int, int)

MyMoveTo constructor with canvas and coordinates

```
public MyMoveTo(Canvas c, int x, int y)
```

Parameters

`c` Canvas

`x` int

`y` int

Methods

CheckParameters(string[])

CheckParameters for MyMoveTo command

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

Parameters

`parameterList` string[]

Execute()

Execute the MyMoveTo command

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

Exceptions

CanvasException

Class MyPen

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

Pen command - sets the drawing pen color using RGB values.

```
public class MyPen : CommandThreeParameters, ICommand
```

Inheritance

object ← Command ← CanvasCommand ← CommandOneParameter ← CommandTwoParameters ← CommandThreeParameters ← MyPen

Implements

ICommand

Inherited Members

CommandThreeParameters.param3 , CommandThreeParameters.param3unprocessed , CommandTwoParameters.param2 , CommandTwoParameters.param2unprocessed , CommandOneParameter.param1 , CommandOneParameter.param1unprocessed , CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas , Command.IsDouble , 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

MyPen()

MyPen constructor

```
public MyPen()
```

MyPen(Canvas, int, int, int)

MyPen constructor with canvas and RGB values

```
public MyPen(Canvas c, int r, int g, int b)
```

Parameters

`c` Canvas

`r` int

`g` int

`b` int

Methods

CheckParameters(string[])

CheckParameters for MyPen command

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

Parameters

`parameterList` string[]

Exceptions

CommandException

Execute()

Execute the MyPen command

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

Exceptions

CanvasException

Class MyRect

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

MyRect command - draws a rectangle with specified width and height, optionally filled.

```
public class MyRect : CommandTwoParameters, ICommand
```

Inheritance

object ← Command ← CanvasCommand ← CommandOneParameter ← CommandTwoParameters ← MyRect

Implements

ICommand

Inherited Members

CommandTwoParameters.param2 , CommandTwoParameters.param2unprocessed , CommandOneParameter.param1 , CommandOneParameter.param1unprocessed , CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas , Command.IsDouble , 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

MyRect()

MyRect constructor

```
public MyRect()
```

MyRect(Canvas, int, int, bool)

MyRect constructor with canvas, width, height, and filled option

```
public MyRect(Canvas c, int width, int height, bool filled)
```

Parameters

`c` Canvas

`width` int

`height` int

`filled` bool

Methods

CheckParameters(string[])

CheckParameters for MyRect command

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

Parameters

`parameterList` string[]

Exceptions

CommandException

Execute()

Execute the MyRect command

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

Exceptions

CanvasException

Class MyReset

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

MyReset command - resets the canvas and clears stored commands.

```
public class MyReset : Command, ICommand
```

Inheritance

object ← Command ← MyReset

Implements

ICommand

Inherited Members

Command.IsDouble , 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

MyReset(AppCanvas, StoredProgram)

MyReset constructor

```
public MyReset(AppCanvas canvas, StoredProgram prog)
```

Parameters

`canvas` AppCanvas

`prog` StoredProgram

Methods

CheckParameters(string[])

CheckParameters for MyReset command

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

Parameters

`parameter` string[]

Exceptions

CommandException

Execute()

Execute the MyReset command

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

Class MyTri

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

MyTri command - draws a triangle with specified width and height.

```
public class MyTri : CommandTwoParameters, ICommand
```

Inheritance

object ← Command ← CanvasCommand ← CommandOneParameter ← CommandTwoParameters ← MyTri

Implements

ICommand

Inherited Members

CommandTwoParameters.param2 , CommandTwoParameters.param2unprocessed , CommandOneParameter.param1 , CommandOneParameter.param1unprocessed , CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas , Command.IsDouble , 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

MyTri()

MyTri constructor

```
public MyTri()
```

Methods

CheckParameters(string[])

CheckParameters for MyTri command

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

Parameters

`parameterList` string[]

Execute()

Execute the MyTri command

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

Exceptions

CanvasException

Class MyWhileCommand

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

Implements the WHILE loop command

```
public class MyWhileCommand : Command, ICommand
```

Inheritance

object ← Command ← MyWhileCommand

Implements

ICommand

Inherited Members

Command.IsDouble , 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

MyWhileCommand()

MyWhileCommand constructor

```
public MyWhileCommand()
```

Methods

AddToLoopBlock(string)

Add a command to the loop block

```
public void AddToLoopBlock(string command)
```

Parameters

`command` string

CheckParameters(string[])

CheckParameters for WHILE command

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

Parameters

`parameters` string[]

Exceptions

CommandException

Execute()

Execute the WHILE command

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

Class MyWrite

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

myWrite command - writes text on the canvas at the current pen position.

```
public class MyWrite : Command, ICommand
```

Inheritance

object ← Command ← MyWrite

Implements

ICommand

Inherited Members

Command.IsDouble , 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

MyWrite(ICanvas)

MyWrite constructor

```
public MyWrite(ICanvas canvas)
```

Parameters

`canvas` ICanvas

Methods

CheckParameters(string[])

CheckParameters for Write command

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

Parameters

`parameters` string[]

Exceptions

CommandException

Execute()

Execute the Write command

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

Exceptions

Exception

Class Program

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

```
public static class Program
```

Inheritance

object ← Program

Inherited Members

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

Class UnknownCommand

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

Command that does nothing for unknown commands Prevents exceptions during parsing

```
public class UnknownCommand : ICommand
```

Inheritance

object ← UnknownCommand

Implements

ICommand

Inherited Members

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

Properties

Canvas

Canvas property required by ICommand interface.

```
public Canvas Canvas { get; set; }
```

Property Value

Canvas

Methods

CheckParameters(string[])

CheckParameters method required by ICommand interface.

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

Parameters

`parameters` string[]

Compile()

Compile method required by ICommand interface.

```
public void Compile()
```

Exceptions

NotImplementedException

Execute()

Execute method that does nothing.

```
public void Execute()
```

Set(StoredProgram, string)

Set method required by ICommand interface.

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

Parameters

`Program` StoredProgram

`Params` string

Exceptions

NotImplementedException

Class VariableStore

Namespace: Medium_Scale_Software_Engineering_Project

Assembly: Medium Scale Software Engineering Project.dll

Singleton class that replaces BOOSE internal variable handling for Int, Real, Array, and Method types with unlimited capacity.

```
public sealed class VariableStore : IVariableStore
```

Inheritance

object ← VariableStore

Implements

IVariableStore

Inherited Members

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

Properties

CurrentMethod

Current method name being executed.

```
public string CurrentMethod { get; }
```

Property Value

string

Instance

Get the singleton instance.

```
public static VariableStore Instance { get; }
```

Property Value

VariableStore

Methods

Clear()

Clears all variables and methods from the store.

```
public void Clear()
```

DefineMethod(string, MethodDefinition)

Defines a new method.

```
public void DefineMethod(string name, MethodDefinition definition)
```

Parameters

`name` string

`definition` MethodDefinition

EnterMethod(string)

Enters a method context for tracking current method.

```
public void EnterMethod(string methodName)
```

Parameters

`methodName` string

EvaluateExpression(string)

Evaluates an expression string and returns the result.

```
public object EvaluateExpression(string expression)
```

Parameters

`expression` string

Returns

object

ExitMethod()

Exits the current method context.

```
public void ExitMethod()
```

GetMethod(string)

Gets a method definition by name.

```
public MethodDefinition GetMethod(string name)
```

Parameters

`name` string

Returns

MethodDefinition

GetVariable(string)

Gets a variable by name, searching from the current scope outward.

```
public object GetVariable(string name)
```

Parameters

`name` string

Returns

object

MethodExists(string)

Method existence check.

```
public bool MethodExists(string name)
```

Parameters

`name` string

Returns

bool

PopScope()

Pops the current variable scope from the stack.

```
public void PopScope()
```

PushScope()

Pushes a new variable scope onto the stack.

```
public void PushScope()
```

SetVariable(string, object, string)

Sets a variable in the current scope.

```
public void SetVariable(string name, object value, string type = null)
```

Parameters

`name` string

`value` object

`type` string

VariableExists(string)

Verifies if a variable exists in any scope.

```
public bool VariableExists(string name)
```

Parameters

`name` string

Returns

bool