Namespace ASE_Assignment

Classes

DrawingSurface

DrawingSurface is an implemntation of the ICanvas interface.

ExtendedCommandFactory

Inherits CommandFactory to add further commands to BOOSE
Used to overwrite datatypes to enable full functionality

FreeArray

FreeArray Inherits BOOSE.Array, bypassing the restrictions by calling ReduceRestrictionCounter when the program checks the counter.

FreeBool

A re-write of BOOSE.bool, removing the maximum boolean restriction.

FreeCompoundCommand

A re-write of CompoundCommand, removing the restrictions.

FreeElse

FreeElse has the restrictions removed, courtesy of FreeCompoundCommand.

FreeEnd

Rewritten FreeEnd to remove restrictions, courtesy of FreeCompoundCommand

FreeFor

FreeFor inherits BOOSE.For

FreeIf

Rewritten FreeIf to remove restrictions, courtesy of FreeCompoundCommand
If doesn't require a custom compile/execute. Therefore, the restrictions removed from FreeCompound command are sufficient.

FreeInt

Integer class made from the inheritance of evaluation.

FreeReal

FreeReal inherits BOOSE.Real, removing restrictions by overriding the command.

FreeWhile

Rewritten FreeWhile to remove restrictions, courtesy of FreeCompoundCommand
While doesn't require a custom compile/execute. Therefore, the restrictions removed from

FreeCompound command are sufficient.

PaintForm

Paintform MSWindows form for interacting with BOOSE

UnrestrictedParser

Extended of the IParser interface, A parser made for the BOOSE Language.

UnrestrictedProgram

Extension of StoredProgram, overriding the run remove the restrictions.

Class DrawingSurface

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

DrawingSurface is an implemntation of the ICanvas interface.

```
public class DrawingSurface : ICanvas
```

Inheritance

object ← DrawingSurface

Implements

ICanvas

Inherited Members

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

Constructors

DrawingSurface(int, int)

Constructor method for DrawingSurface.

```
public DrawingSurface(int Width, int Height)
```

Parameters

`Width` int

Pixel width of DrawingSurface

`Height` int

Pixel height of DrawingSurface

Properties

Canvas_Height

Pixel height of Canvas

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

Property Value

int

Canvas_Width

Pixel width of Canvas

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

Property Value

int

PenColour

Current Colour of the Pen

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

Property Value

object

PenSize

Pixel radius of pen.

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

Property Value

int

Xpos

Horizontal positon of the Pen

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

Property Value

int

Ypos

Vertical position of the Pen

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

Property Value

int

Methods

Circle(int, bool)

Draws a circle from the pen's current position, with the specified radius.

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

Parameters

`radius` int

The radius of the circle, in pixels.

`filled` bool

True to draw a filled circle; false to draw only the outline.

Examples

Example 1:

```
circle 50, true
```

Creates a filled circle with a diameter of 100 pixels.

Example 2:

```
circle 100, false
```

Creates an outlined circle with a diameter of 200 pixels and an edge width matching the pen's width.

Clear()

Returns the DrawingSurface to its original color, removing all drawings.

```
public void Clear()
```

DrawTo(int, int)

Draws a line from current pen position to desired pen position.
Updates pen position when finished.

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

Parameters

`x` int

Desired final X position of pen

`y` int

Desired final Y position of pen

Examples

Example 1

```
drawto 100,100
```

Will draw a line to position 100,100 and update Xpos and Ypos
All new drawing will be done with the pen from this position unless moved.

Example 2

```
drawto 100000,100000
```

Anything beyond bounds will result in an error message.

Exceptions

CanvasException

Drawing beyond DrawingCanvas boundaries
Check Canvas_Width and Canvas_Height.

MoveTo(int, int)

Moves the pen from current position to specified one.

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

Parameters

x int

Desired X position of pen

y int

Desired Y position of pen

Examples

```
moveto 100,100
```

Updates the pen's position to Xpos and Ypos to 100 respectively.

Exceptions

CanvasException

Attempt to move pen out of bounds
Check Canvas_Width and Canvas_Height.

Rect(int, int, bool)

Draws a rectangle.

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

Parameters

`width` int

Pixel width of rectangle

`height` int

Pixel height of rectangle

`filled` bool

Determines if the rectangle is filled or not.

Reset()

Reset the pen's attributes.
Position: (0,0)
Width: 4 pixels
Color: Red

```
public void Reset()
```

Set(int, int)

Establishes the Canvas and Bitmap size.
Width/Height must match drawing area size.

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

Parameters

`width` int

Pixel width of Canvas + Bitmap

`height` int

Pixel height of Canvas + Bitmap

SetColour(int, int, int)

Set the pen's colour using RGB values

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

Parameters

`red` int

RGB Red Value

`green` int

RGB Blue Value

`blue` int

RGB Green Value

Exceptions

CanvasException

RGB Value out of range
Exceeds 255 or is Below 0

Tri(int, int)

Draws a triangle from pen postion in the order:
`East -> North West -> South West`
Only capable of Equilateral or Isosceles triangles.

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

Parameters

`width` int

Pixel width of the base of the triangle.

`height` int

Pixel height from base to highest point.

WriteText(string)

Draws text at pen's position. Doesn't update pen position.
Font: Arial
Size: 10

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

Parameters

`text` string

Desired text to be output.

getBitmap()

Returns the Canvas' bitmap in its current state.

```
public object getBitmap()
```

Returns

object

```
Bitmap CanvasBitmap
```

Class ExtendedCommandFactory

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

Inherits CommandFactory to add further commands to BOOSE
Used to overwrite datatypes to enable full functionality

```
public class ExtendedCommandFactory : CommandFactory, ICommandFactory
```

Inheritance

object ← CommandFactory ← ExtendedCommandFactory

Implements

ICommandFactory

Inherited Members

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

Methods

MakeCommand(string)

Creates a new object of a specified BOOSE command

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

Parameters

`commandType` string

BOOSE command as a string

Returns

ICommand

An object of the specified command.

Class FreeArray

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

FreeArray Inherits BOOSE.Array, bypassing the restrictions by calling ReduceRestrictionCounter when the program checks the counter.

```
public class FreeArray : Array, ICommand
```

Inheritance

object ← Command ← Evaluation ← Array ← FreeArray

Implements

ICommand

Inherited Members

Array.PEEK , Array.POKE , Array.type , Array.rows , Array.columns , Array.valueInt , Array.valueReal , Array.intArray , Array.realArray , Array.pokeValue , Array.peekVar , Array.rowS , Array.columnS , Array.row , Array.column , Array.ArrayRestrictions() , Array.ReduceRestrictionCounter() , Array.Compile() , Array.Execute() , Array.ProcessArrayParametersExecute(bool) , Array.SetIntArray(int, int, int) , Array.SetRealArray(double, int, int) , Array.GetIntArray(int, int) , Array.GetRealArray(int, int) , Array.Rows , Array.Columns , Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , Evaluation.ProcessExpression(string) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , 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

FreeArray()

```
public FreeArray()
```

Methods

CheckParameters(string[])

Ensures that there are enough parameters to create an array.

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

Parameters

`parameterList` string[]

The parameters passed to make the array

ProcessArrayParametersCompile(bool)

Compiles the ArrayParameters, string parameters which are elements of the array.
The array can only hold integers and reals.

```
protected override void ProcessArrayParametersCompile(bool peekOrPoke)
```

Parameters

`peekOrPoke` bool

Ensures parameters passed are Integers or Reals

Class FreeBool

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

A re-write of BOOSE.bool, removing the maximum boolean restriction.

```
public class FreeBool : Evaluation, ICommand
```

Inheritance

object ← Command ← Evaluation ← FreeBool

Implements

ICommand

Inherited Members

Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , Evaluation.Compile() , Evaluation.CheckParameters(string[]) , Evaluation.ProcessExpression(string) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , 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

FreeBool()

```
public FreeBool()
```

Properties

BoolValue

Current Value of the boolean

```
public bool BoolValue { get; set; }
```

Property Value

bool

Methods

Execute()

Updates the polarity of the boolean

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

Exceptions

StoredProgramException

Throws an exception if an invalid boolean expression is passed.

Class FreeCompoundCommand

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

A re-write of CompoundCommand, removing the restrictions.

```
public class FreeCompoundCommand : ConditionalCommand, ICommand
```

Inheritance

object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← FreeCompoundCommand

Implements

ICommand

Derived

FreeElse, FreeEnd, FreeIf, FreeWhile

Inherited Members

ConditionalCommand.endLineNumber , ConditionalCommand.Execute() , ConditionalCommand.EndLineNumber , ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType , ConditionalCommand.ReturnLineNumber , Boolean.Restrictions() , Boolean.BoolValue , Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , Evaluation.ProcessExpression(string) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , 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

FreeCompoundCommand()

```
public FreeCompoundCommand()
```

Properties

Command

Stores the command that will be run if a condition is met.

```
public ConditionalCommand Command { get; set; }
```

Property Value

ConditionalCommand

Methods

CheckParameters(string[])

Ensures that there are parameters passed for the condition.
Also ensurs that there are end if/while/for

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

Parameters

`parameter` string[]

Exceptions

CommandException

Compile()

```
public override void Compile()
```

Class FreeElse

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

FreeElse has the restrictions removed, courtesy of FreeCompoundCommand.

```
public class FreeElse : FreeCompoundCommand, ICommand
```

Inheritance

object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← FreeCompoundCommand ← FreeElse

Implements

ICommand

Inherited Members

FreeCompoundCommand.Command , ConditionalCommand.endLineNumber , ConditionalCommand.EndLineNumber , ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType , ConditionalCommand.ReturnLineNumber , Boolean.Restrictions() , Boolean.BoolValue , Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , Evaluation.ProcessExpression(string) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , 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

FreeElse()

```
public FreeElse()
```

Methods

CheckParameters(string[])

Ensures that else contains no other parameters.
Ensures 'else' isn't used when there is no end present.

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

Parameters

`parameter` string[]

Singular else coommand

Exceptions

CommandException

Compile()

Compiles the command, removing it of the program stack when finished.

```
public override void Compile()
```

Execute()

Repositions the program counter to the end location if an else clause is triggered.

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

Class FreeEnd

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

Rewritten FreeEnd to remove restrictions, courtesy of FreeCompoundCommand

```
public class FreeEnd : FreeCompoundCommand, ICommand
```

Inheritance

object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← FreeCompoundCommand ← FreeEnd

Implements

ICommand

Inherited Members

FreeCompoundCommand.Command , FreeCompoundCommand.CheckParameters(string[]) , ConditionalCommand.endLineNumber , ConditionalCommand.EndLineNumber , ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType , ConditionalCommand.ReturnLineNumber , Boolean.Restrictions() , Boolean.BoolValue , Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , Evaluation.ProcessExpression(string) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , 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

FreeEnd()

```
public FreeEnd()
```

Methods

Compile()

Compiles the end command, ensuring it specifies a target operation.

```
for count = 1 to 20 step 2
circle count * 10
end for -- Example of declaration used correctly
```

```
public override void Compile()
```

Exceptions

CommandException

Throws an exception if end doesn't declare method

Execute()

Determines the iteration/selection type and alters the PC accordingly.
Based on the outcome of the selection/loop status

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

Exceptions

CommandException

Class FreeFor

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

FreeFor inherits BOOSE.For

```
public class FreeFor : For, ICommand
```

Inheritance

object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← For ← FreeFor

Implements

ICommand

Inherited Members

For.Compile() , For.Execute() , For.LoopControlV , For.From , For.To , For.Step , ConditionalCommand.endLineNumber , ConditionalCommand.EndLineNumber , ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType , ConditionalCommand.ReturnLineNumber , Boolean.Restrictions() , Boolean.BoolValue , Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , Evaluation.CheckParameters(string[]) , Evaluation.ProcessExpression(string) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , 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

FreeFor()

```
public FreeFor()
```

Class FreeIf

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

Rewritten FreeIf to remove restrictions, courtesy of FreeCompoundCommand
If doesn't require a custom compile/execute. Therefore, the restrictions removed from FreeCompound command are sufficient.

```
public class FreeIf : FreeCompoundCommand, ICommand
```

Inheritance
object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← FreeCompoundCommand ← FreeIf

Implements
ICommand

Inherited Members
FreeCompoundCommand.Command , FreeCompoundCommand.CheckParameters(string[]) , FreeCompoundCommand.Compile() , ConditionalCommand.endLineNumber , ConditionalCommand.Execute() , ConditionalCommand.EndLineNumber , ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType , ConditionalCommand.ReturnLineNumber , Boolean.Restrictions() , Boolean.BoolValue , Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , Evaluation.ProcessExpression(string) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , 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

FreeIf()

```
public FreeIf()
```

Class FreeInt

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

Integer class made from the inheritance of evaluation.

```
public class FreeInt : Evaluation, ICommand
```

Inheritance

object ← Command ← Evaluation ← FreeInt

Implements

ICommand

Inherited Members

Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , Evaluation.CheckParameters(string[]) , Evaluation.ProcessExpression(string) , Evaluation.Expression , Evaluation.VarName , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , 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

FreeInt()

```
public FreeInt()
```

Properties

Value

BOOSE Integer's Value

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

Property Value

int

Methods

Compile()

```
public override void Compile()
```

Execute()

Ensures that value is an integer before execution.

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

Exceptions

StoredProgramException

Throws an exception if value is not an integer, or is a double.

Class FreeReal

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

FreeReal inherits BOOSE.Real, removing restrictions by overriding the command.

```
public class FreeReal : Real, ICommand
```

Inheritance

object ← Command ← Evaluation ← Real ← FreeReal

Implements

ICommand

Inherited Members

Real.Compile() , Real.Execute() , Real.Value , Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , Evaluation.CheckParameters(string[]) , Evaluation.ProcessExpression(string) , Evaluation.Expression , Evaluation.VarName , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , 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

FreeReal()

```
public FreeReal()
```

Methods

Restrictions()

Command overriden to null, removing the restrictions.

```
public override void Restrictions()
```

Class FreeWhile

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

Rewritten FreeWhile to remove restrictions, courtesy of FreeCompoundCommand
While doesn't require a custom compile/execute. Therefore, the restrictions removed from FreeCompound command are sufficient.

```
public class FreeWhile : FreeCompoundCommand, ICommand
```

Inheritance

object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← FreeCompoundCommand ← FreeWhile

Implements

ICommand

Inherited Members

FreeCompoundCommand.Command , FreeCompoundCommand.CheckParameters(string[]) , FreeCompoundCommand.Compile() , ConditionalCommand.endLineNumber , ConditionalCommand.Execute() , ConditionalCommand.EndLineNumber , ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType , ConditionalCommand.ReturnLineNumber , Boolean.Restrictions() , Boolean.BoolValue , Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , Evaluation.ProcessExpression(string) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , 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

FreeWhile()

```
public FreeWhile()
```

Class PaintForm

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

Paintform MSWindows form for interacting with BOOSE

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

Inheritance

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

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

PaintForm()

Constructor for PaintForm

```
public PaintForm()
```

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 UnrestrictedParser

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

Extended of the IParser interface, A parser made for the BOOSE Language.

```
public class UnrestrictedParser : IParser
```

Inheritance

object ← UnrestrictedParser

Implements

IParser

Inherited Members

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

Constructors

UnrestrictedParser(ExtendedCommandFactory, StoredProgram)

The parser consists of:
CommandFactory: Used to create the compile the commands in ParseCommand(string)

StoredProgram: Used to store the commands that have been compiled for later execution.

```
public UnrestrictedParser(ExtendedCommandFactory Factory, StoredProgram Program)
```

Parameters

`Factory` ExtendedCommandFactory

Command factory creates objects, such as parameters/commands, requried to run boose.

`Program` StoredProgram

Stores comamnds and variables. Is responsible for program execution.

Methods

ParseCommand(string)

Converts the string context of BOOSED in functioning code

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

Parameters

`Line` string

One line of a BOOSE program

Returns

ICommand

A fully built command specified by the ICommand interface

ParseProgram(string)

Seperates the BOOSE code into invidiual lines using SplitProgram(string), parsing them with ParseCommand(string)

```
public void ParseProgram(string program)
```

Parameters

`program` string

Entire BOOSE program passed as a string.

Class UnrestrictedProgram

Namespace: ASE_Assignment

Assembly: ASE Assignment.dll

Extension of StoredProgram, overriding the run remove the restrictions.

```
public class UnrestrictedProgram : StoredProgram, IList, ICollection, IEnumerable,
ICloneable, IStoredProgram
```

Inheritance

object ← ArrayList ← StoredProgram ← UnrestrictedProgram

Implements

IList, ICollection, IEnumerable, ICloneable, IStoredProgram

Inherited Members

StoredProgram.SyntaxOk , StoredProgram.AddMethod(Method) , StoredProgram.GetMethod(string) , StoredProgram.AddVariable(Evaluation) , StoredProgram.GetVariable(string) , StoredProgram.GetVariable(int) , StoredProgram.FindVariable(Evaluation) , StoredProgram.FindVariable(string) , StoredProgram.VariableExists(string) , StoredProgram.GetVarValue(string) , StoredProgram.UpdateVariable(string, int) , StoredProgram.UpdateVariable(string, double) , StoredProgram.UpdateVariable(string, bool) , StoredProgram.DeleteVariable(string) , StoredProgram.IsExpression(string) , StoredProgram.EvaluateExpressionWithString(string) , StoredProgram.EvaluateExpression(string) , StoredProgram.Push(ConditionalCommand) , StoredProgram.Pop() , StoredProgram.Add(Command) , StoredProgram.NextCommand() , StoredProgram.ResetProgram() , StoredProgram.Commandsleft() , StoredProgram.PC , ArrayList.Adapter(IList) , ArrayList.Add(object) , ArrayList.AddRange(ICollection) , ArrayList.BinarySearch(int, int, object, IComparer) , ArrayList.BinarySearch(object) , ArrayList.BinarySearch(object, IComparer) , ArrayList.Clear() , ArrayList.Clone() , ArrayList.Contains(object) , ArrayList.CopyTo(Array) , ArrayList.CopyTo(Array, int) , ArrayList.CopyTo(int, Array, int, int) , ArrayList.FixedSize(ArrayList) , ArrayList.FixedSize(IList) , ArrayList.GetEnumerator() , ArrayList.GetEnumerator(int, int) , ArrayList.GetRange(int, int) , ArrayList.IndexOf(object) , ArrayList.IndexOf(object, int) , ArrayList.IndexOf(object, int, int) , ArrayList.Insert(int, object) , ArrayList.InsertRange(int, ICollection) , ArrayList.LastIndexOf(object) , ArrayList.LastIndexOf(object, int) , ArrayList.LastIndexOf(object, int, int) , ArrayList.ReadOnly(ArrayList) , ArrayList.ReadOnly(IList) , ArrayList.Remove(object) , ArrayList.RemoveAt(int) , ArrayList.RemoveRange(int, int) , ArrayList.Repeat(object, int) , ArrayList.Reverse() , ArrayList.Reverse(int, int) , ArrayList.SetRange(int, ICollection) ,

ArrayList.Sort() , ArrayList.Sort(IComparer) , ArrayList.Sort(int, int, IComparer) , ArrayList.Synchronized(ArrayList) , ArrayList.Synchronized(IList) , ArrayList.ToArray() , ArrayList.ToArray(Type) , ArrayList.TrimToSize() , ArrayList.Capacity , ArrayList.Count , ArrayList.IsFixedSize , ArrayList.IsReadOnly , ArrayList.IsSynchronized , ArrayList.this[int] , ArrayList.SyncRoot , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

UnrestrictedProgram(ICanvas)

```
public UnrestrictedProgram(ICanvas canvas)
```

Parameters

`canvas` ICanvas

Methods

Run()

Compiles the commands and is responsible for executing them sequentially

```
public override void Run()
```

Exceptions

StoredProgramException

If an error occurs, it throws an exception displaying the line where it happend.