Namespace ASEProject

Classes

Test1

Class Test1

Namespace: ASEProject

Assembly: ASEProject.dll

```
[TestClass]
public sealed class Test1
```

Inheritance

object ← Test1

Inherited Members

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

Methods

TestMethod1()

```
[TestMethod]
public void TestMethod1()
```

Namespace ASE_Assignment_Demo

Classes

AppCanvas

A class representing a drawable canvas for graphics operations. Implements the ICanvas interface.

ArrayApp

The ArrayApp class inherits from the BOOSE.Array class. This class is designed to demonstrate overriding the constructor of the parent class to modify its behavior.

CallApp

CanvasLoader

Represents a class that provides functionality to load a canvas image from the file system using a graphical file dialog.

CanvasSaver

Represents a class that provides functionality to save a canvas image to the file system using a graphical save file dialog.

ClearApp

Represents a command to clear the canvas, extending the `CanvasCommand` class.

CommandFactoryApp

A custom implementation of the CommandFactory that creates commands specific to the application.

CommandFileReader

Provides functionality to read command strings from a file using a graphical file dialog.

CommandFileWriter

Provides functionality to write commands to a file using a graphical save file dialog.

ElseApp

The ElseApp class inherits from the BOOSE.Else class. This class is intended to demonstrate overriding the Restrictions method from the parent class.

EndApp

The EndApp class inherits from the BOOSE.End class. This class demonstrates overriding the Restrictions method of the parent class.

ForApp

The ForApp class inherits from the BOOSE.For class. This class demonstrates overriding the Restrictions method of the parent class.

Form1

IfApp

The IfApp class inherits from the BOOSE.If class. This class demonstrates modifying the behavior of the parent class by overriding the constructor and the Restrictions method.

IntApp

The IntApp class inherits from the BOOSE.Int class. This class demonstrates overriding both the constructor and the Restrictions method of the parent class.

MethodApp

The MethodApp class inherits from the BOOSE.Method class. This class demonstrates modifying the behavior of the parent class by overriding the constructor and the Restrictions method.

RealApp

ResetApp

The `AppReset` class is a command that resets the canvas. It inherits from the `CanvasCommand` class.

TriangleApp

A command to draw a triangle on the canvas. Inherits from CommandTwoParameters.

WhileApp

The WhileApp class inherits from the BOOSE.While class. This class demonstrates modifying the behavior of the parent class by overriding the constructor and the Restrictions method.

WriteApp

A command to write text on the canvas. Inherits from CommandOneParameter.

Class AppCanvas

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

A class representing a drawable canvas for graphics operations. Implements the ICanvas interface.

```
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()

Constructor to initialize the canvas with default size and pen properties.

```
public AppCanvas()
```

Properties

Pen

Gets and sets the Pen object used for drawing.

```
public Pen Pen { get; set; }
```

Property Value

Pen

PenColour

Gets and sets the pen color.

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

Property Value

object

Xpos

Gets and sets the X position of the pen.

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

Property Value

int

Ypos

Gets and sets the Y position of the pen.

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

Property Value

int

Methods

Circle(int, bool)

Draws a circle of specified radius at the given pen position.

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

Parameters

`radius` int

Radius of the circle.

`filled` bool

True to fill the circle, false to draw only the outline.

Exceptions

CanvasException

Is thrown when graphics is not intialized or when the radius is 0.

Clear()

Clears the canvas and resets the pen position.

```
public void Clear()
```

DrawTo(int, int)

Draws a line to the provided coordinates from the pen's current position.

```
public void DrawTo(int toX, int toY)
```

Parameters

`toX` int

X-coordinate of the endpoint.

`toY` int

Y-coordinate of the endpoint.

Exceptions

CanvasException

Is thrown when graphics is not initizalied or the coordinates are out of the canvas bounds.

InitializeGraphics(Bitmap)

Initializes the graphics object with the given bitmap.

```
public void InitializeGraphics(Bitmap bitmap)
```

Parameters

`bitmap` Bitmap

The bitmap to use for drawing.

Exceptions

CanvasException

Is thrown when graphic object initialization fails.

MoveTo(int, int)

Moves the pen to a new position.

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

Parameters

`x` int

X-coordinate of the new position.

`y` int

Y-coordinate of the new position.

Exceptions

CanvasException

Is thrown when the corrdinates are outside the canvas bounds.

Rect(int, int, bool)

Draws a rectangle at the current pen position.

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

Parameters

`width` int

Width of the rectangle.

`height` int

Height of the rectangle.

`filled` bool

True to fill the rectangle, false for outline only.

Exceptions

CanvasException

Is thrown is when graphics is not initizalied or widht or height is less than 0.

Reset()

Resets the pen position to the origin.

```
public void Reset()
```

Set(int, int)

Sets the canvas dimensions and initializes the graphics context.

```
public void Set(int xsize, int ysize)
```

Parameters

`xsize` int

Width of the canvas.

`ysize` int

Height of the canvas.

SetColour(int, int, int)

Sets the pen color using RGB values.

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

Parameters

`red` int

Red component (0-255).

`green` int

Green component (0-255).

`blue` int

Blue component (0-255).

Exceptions

CanvasException

Is thrown when the RGB values are outside the valid range (0-255).

Tri(int, int)

Draws a triangle at the current pen position.

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

Parameters

`width` int

Base width of the triangle.

`height` int

Height of the triangle.

Exceptions

CanvasException

Is thrown when the width or height is less than 0 or graphics context is not initialized.

WriteText(string)

Writes text at the current pen position.

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

Parameters

`text` string

Text to write on the canvas.

Exceptions

CanvasException

Is thrown when the text is null or empty or the graphics context is not initialized.

getBitmap()

Returns the bitmap used for drawing.

```
public object getBitmap()
```

Returns

object

The bitmap object.

Class ArrayApp

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

The ArrayApp class inherits from the BOOSE.Array class. This class is designed to demonstrate overriding the constructor of the parent class to modify its behavior.

```
public class ArrayApp : Array, ICommand
```

Inheritance

object ← Command ← Evaluation ← Array ← ArrayApp

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.CheckParameters(string[]) , Array.Execute() , Array.ProcessArrayParametersCompile(bool) , 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

ArrayApp()

Default constructor for the ArrayApp class. This constructor overrides the parent class constructor to remove a certain restriction by calling the ReduceRestrictionCounter method.

```
public ArrayApp()
```

Class CallApp

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

```
public class CallApp : Call, ICommand
```

Inheritance

object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← CompoundCommand ← Call ← CallApp

Implements

ICommand

Inherited Members

Call.methodName , Call.Compile() , Call.Execute() , CompoundCommand.ReduceRestrictions() , CompoundCommand.CheckParameters(string[]) , CompoundCommand.CorrespondingCommand , ConditionalCommand.endLineNumber , ConditionalCommand.EndLineNumber , ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType , ConditionalCommand.ReturnLineNumber , 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

CallApp()

```
public CallApp()
```

Methods

Restrictions()

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

Class CanvasLoader

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

Represents a class that provides functionality to load a canvas image from the file system using a graphical file dialog.

```
public class CanvasLoader
```

Inheritance

object ← CanvasLoader

Inherited Members

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

Remarks

The CanvasLoader class uses a Windows Forms OpenFileDialog to allow the user to select an image file (PNG, JPEG, BMP) to load into the application. The selected file is loaded as a Bitmap object. If the user cancels the dialog or an error occurs during loading, the method returns `null`.

Constructors

CanvasLoader()

Initializes a new instance of the CanvasLoader class.

```
public CanvasLoader()
```

Remarks

This constructor initializes the class without requiring any parameters. The class provides the capability to load an image through a user-interactive dialog, which can be invoked using the LoadCanvas() method.

Methods

LoadCanvas()

Opens a file dialog to allow the user to select an image file to load as a canvas.

```
public Bitmap LoadCanvas()
```

Returns

Bitmap

A Bitmap object representing the loaded image if successful; otherwise, `null`.

Remarks

The method displays an OpenFileDialog configured to filter for common image file formats (PNG, JPEG, BMP). If the user selects a file, the method attempts to load it as a Bitmap.

- Returns a Bitmap if the image loads successfully.
- Returns `null` if the user cancels the dialog or an error occurs.

If the image fails to load due to an invalid file or other issues, an error message is shown using Show(string, string, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, bool) with detailed exception information.

The following example demonstrates how to use the LoadCanvas() method:

```
CanvasLoader loader = new CanvasLoader();
Bitmap canvas = loader.LoadCanvas();
if (canvas != null)
{
    // Successfully loaded the canvas
    pictureBox.Image = canvas;
}
else
{
    // The user canceled or an error occurred
    MessageBox.Show("No canvas was loaded.");
}
```

Exceptions

ArgumentException

Thrown internally if the file selected by the user is invalid or cannot be loaded as a Bitmap.

OutOfMemoryException

Thrown internally if the file is too large or the system is out of memory.

Class CanvasSaver

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

Represents a class that provides functionality to save a canvas image to the file system using a graphical save file dialog.

```
public class CanvasSaver
```

Inheritance

object ← CanvasSaver

Inherited Members

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

Remarks

The CanvasSaver class uses a Windows Forms SaveFileDialog to allow the user to specify a file name and format for saving a Bitmap image. It supports saving images in PNG, JPEG, and BMP formats, as well as other formats if specified. If the user cancels the dialog or an error occurs during saving, the method handles the error gracefully and informs the user.

Constructors

CanvasSaver()

Initializes a new instance of the CanvasSaver class.

```
public CanvasSaver()
```

Remarks

This constructor initializes the class without requiring any parameters. The class provides the capability to save an image through a user-interactive dialog, which can be invoked using the SaveCanvas(Bitmap) method.

Methods

SaveCanvas(Bitmap)

Opens a save file dialog to allow the user to specify the file name and format for saving the canvas image.

```
public void SaveCanvas(Bitmap canvasBitmap)
```

Parameters

`canvasBitmap` Bitmap

The Bitmap image to be saved.

Remarks

The method displays a SaveFileDialog configured to filter for common image file formats (PNG, JPEG, BMP). Based on the user's selected file extension, the appropriate image format is applied.

- If the user selects a file and the image is successfully saved, a success message is displayed.
- If the user cancels the dialog, the method performs no action.
- If an error occurs during the save process, an error message is displayed with exception details.

The method ensures compatibility with multiple image formats and provides a simple interface for saving images from a canvas to the file system.

The following example demonstrates how to use the SaveCanvas(Bitmap) method:

```
CanvasSaver saver = new CanvasSaver();
Bitmap canvas = new Bitmap(100, 100); // Example canvas
saver.SaveCanvas(canvas);
```

Exceptions

ArgumentNullException

Thrown if `canvasBitmap` is `null`.

Class ClearApp

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

Represents a command to clear the canvas, extending the `CanvasCommand` class.

```
public class ClearApp : CanvasCommand, ICommand
```

Inheritance

object ← Command ← CanvasCommand ← ClearApp

Implements

ICommand

Inherited Members

CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas , 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

ClearApp()

Default constructor that initializes a new instance of the `ClearApp` class. Invokes the base class constructor without parameters

```
public ClearApp()
```

ClearApp(ICanvas)

Initializes a new instance of the `ClearApp` class with a specified canvas. Passes the canvas object to the base class constructor.

```
public ClearApp(ICanvas c)
```

Parameters

`c` ICanvas

The canvas object to be cleared.

Methods

CheckParameters(string[])

Checks the parameters passed to the `ClearApp` command. An exception is thrown indicating that the command does not accept any parameters.

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

Parameters

`parameter` string[]

An array of strings representing the parameters passed to the command.

Exceptions

ArgumentException

Thrown when more than one parameter is provided.

Execute()

Executes the `Clear` command on the canvas. If the canvas is not null, it will call the `Clear` method on the canvas object. If the canvas is null, it will output a message indicating that the canvas is not set.

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

Class CommandFactoryApp

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

A custom implementation of the CommandFactory that creates commands specific to the application.

```
public class CommandFactoryApp : CommandFactory, ICommandFactory
```

Inheritance

object ← CommandFactory ← CommandFactoryApp

Implements

ICommandFactory

Inherited Members

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

Constructors

CommandFactoryApp()

Constructor that initializes the application canvas.

```
public CommandFactoryApp()
```

Methods

MakeCommand(string)

Overrides the MakeCommand method to create custom commands for the application.

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

Parameters

`commandType` string

The type of command to create, as a string.

Returns

ICommand

An ICommand instance corresponding to the specified command type.

Remarks

The factory supports the following custom commands:

- "tri": Creates an instance of AppTriangle.
- "write": Creates an instance of AppWrite.
  For other command types, the base factory's implementation is used.

Exceptions

ArgumentNullException

Thrown if the `commandType` is null or empty.

Class CommandFileReader

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

Provides functionality to read command strings from a file using a graphical file dialog.

```
public class CommandFileReader
```

Inheritance

object ← CommandFileReader

Inherited Members

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

Remarks

The CommandFileReader class utilizes a Windows Forms OpenFileDialog to let the user select a text file containing commands. The contents of the file are read and returned as a string. If the operation is canceled or an error occurs, the method returns an empty string.

Constructors

CommandFileReader()

Initializes a new instance of the CommandFileReader class.

```
public CommandFileReader()
```

Remarks

This constructor initializes the class without requiring any parameters. The class provides the capability to load commands from a file using the CommandReaders method.

Methods

CommandReader()

Opens a file dialog to let the user select a text file and reads its content.

```
public string CommandReader()
```

Returns

string

A string containing the content of the selected file, or an empty string if the operation fails or is canceled.

Remarks

This method displays an OpenFileDialog configured to filter for text files (*.txt). If a file is selected, its content is read and returned. In case of an error, an error message is shown to the user using Message Box, and an empty string is returned.

- If the user selects a valid file, the content of the file is read and returned.
- If the user cancels the dialog, an empty string is returned.
- If an error occurs during file reading, an error message is shown, and an empty string is returned.

This method supports reading plain text files and ensures that file reading errors are handled gracefully.

The following example demonstrates how to use the LoadCommands method:

```
CommandFileReader loader = new CommandFileReader();
string commands = loader.CommandReader();
if (!string.IsNullOrEmpty(commands))
{
    // Successfully loaded commands
    Console.WriteLine(commands);
}
else
{
    // No commands loaded
    Console.WriteLine("No commands were loaded.");
}
```

Exceptions

UnauthorizedAccessException

Thrown internally if the file cannot be accessed due to permission issues.

IOException

Thrown internally if there is an issue with the file system during reading.

Exception

Thrown internally for other unexpected errors during file reading.

Class CommandFileWriter

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

Provides functionality to write commands to a file using a graphical save file dialog.

```
public class CommandFileWriter
```

Inheritance

object ← CommandFileWriter

Inherited Members

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

Remarks

The CommandFileWriter class uses a Windows Forms SaveFileDialog to allow the user to specify a file name and location for saving commands as text. The commands are saved as plain text in the specified file.

Constructors

CommandFileWriter()

Initializes a new instance of the CommandFileWriter class.

```
public CommandFileWriter()
```

Remarks

This constructor initializes the class without requiring any parameters. The class provides the capability to save commands to a file using the WriteCommandsToFile(string) method.

Methods

WriteCommandsToFile(string)

Opens a save file dialog to allow the user to specify a file name and location, and writes the given commands to the selected file.

```
public void WriteCommandsToFile(string commands)
```

Parameters

`commands` string

A string containing the commands to be saved to the file.

Remarks

This method displays a SaveFileDialog configured to save files with a .txt extension. If a file is selected, the commands are written to the file using a StreamWriter. If the user cancels the dialog or an error occurs during writing, an appropriate message is displayed using MessageBox.

- If the user selects a valid file, the commands are successfully saved to the file.
- If the user cancels the dialog, no action is performed.
- If an error occurs during file writing, an error message is displayed to the user.

The following example demonstrates how to use the WriteCommandsToFile(string) method:

```
CommandFileWriter writer = new CommandFileWriter();
string commands = "DRAW RECTANGLE 100 200";
writer.WriteCommandsToFile(commands);
```

Class ElseApp

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

The ElseApp class inherits from the BOOSE.Else class. This class is intended to demonstrate overriding the Restrictions method from the parent class.

```
public class ElseApp : Else, ICommand
```

Inheritance

object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← CompoundCommand ← Else ← ElseApp

Implements

ICommand

Inherited Members

Else.CheckParameters(string[]) , Else.Compile() , Else.Execute() , Else.CorrespondingEnd , CompoundCommand.ReduceRestrictions() , CompoundCommand.CorrespondingCommand , ConditionalCommand.endLineNumber , ConditionalCommand.EndLineNumber , ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType , ConditionalCommand.ReturnLineNumber , 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)

Methods

Restrictions()

Overrides the Restrictions method from the BOOSE.Else class. This method is intended to modify or implement custom restrictions. Currently, it is an empty implementation.

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

Class EndApp

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

The EndApp class inherits from the BOOSE.End class. This class demonstrates overriding the Restrictions method of the parent class.

```
public class EndApp : End, ICommand
```

Inheritance

object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← CompoundCommand ← End ← EndApp

Implements

ICommand

Inherited Members

End.Compile() , End.Execute() , CompoundCommand.ReduceRestrictions() , CompoundCommand.CheckParameters(string[]) , CompoundCommand.CorrespondingCommand , ConditionalCommand.endLineNumber , ConditionalCommand.EndLineNumber , ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType , ConditionalCommand.ReturnLineNumber , 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)

Methods

Restrictions()

Overrides the Restrictions method from the BOOSE.End class. This method is intended to modify or implement custom restrictions. Currently, it is an empty implementation.

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

Class ForApp

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

The ForApp class inherits from the BOOSE.For class. This class demonstrates overriding the Restrictions method of the parent class.

```
public class ForApp : For, ICommand
```

Inheritance

object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← For ← ForApp

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.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)

Methods

Restrictions()

Overrides the Restrictions method from the BOOSE.For class. This method is intended to modify or implement custom restrictions.

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

Class Form1

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.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.

Class IfApp

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

The IfApp class inherits from the BOOSE.If class. This class demonstrates modifying the behavior of the parent class by overriding the constructor and the Restrictions method.

```
public class IfApp : If, ICommand
```

Inheritance

object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← CompoundCommand ← If ← IfApp

Implements

ICommand

Inherited Members

CompoundCommand.ReduceRestrictions() , CompoundCommand.CheckParameters(string[]) , CompoundCommand.Compile() , CompoundCommand.CorrespondingCommand , ConditionalCommand.endLineNumber , ConditionalCommand.Execute() , ConditionalCommand.EndLineNumber , ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType , ConditionalCommand.ReturnLineNumber , 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

IfApp()

Initializes a new instance of the IfApp class. Calls the ReduceRestrictions method to reduce any imposed restrictions.

```
public IfApp()
```

Methods

Restrictions()

Overrides the Restrictions method from the BOOSE.If class. This method is intended to modify or implement custom restrictions. Currently, it is an empty implementation.

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

Class IntApp

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

The IntApp class inherits from the BOOSE.Int class. This class demonstrates overriding both the constructor and the Restrictions method of the parent class.

```
public class IntApp : Int, ICommand
```

Inheritance

object ← Command ← Evaluation ← Int ← IntApp

Implements

ICommand

Inherited Members

Int.Compile() , Int.Execute() , 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

IntApp()

Default constructor for the IntApp class. This constructor overrides the parent class constructor to potentially remove any restrictions. Currently, it is an empty implementation.

```
public IntApp()
```

Methods

Restrictions()

Overrides the Restrictions method from the BOOSE.Int class. This method is intended to implement or modify custom restrictions. Currently, it is an empty implementation.

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

Class MethodApp

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

The MethodApp class inherits from the BOOSE.Method class. This class demonstrates modifying the behavior of the parent class by overriding the constructor and the Restrictions method.

```
public class MethodApp : Method, ICommand
```

Inheritance

object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← CompoundCommand ← Method ← MethodApp

Implements

ICommand

Inherited Members

Method.CheckParameters(string[]) , Method.Compile() , Method.Execute() , Method.LocalVariables , Method.MethodName , Method.Type , CompoundCommand.ReduceRestrictions() , CompoundCommand.CorrespondingCommand , ConditionalCommand.endLineNumber , ConditionalCommand.EndLineNumber , ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType , ConditionalCommand.ReturnLineNumber , 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

MethodApp()

Initializes a new instance of the MethodApp class. Calls the ReduceRestrictions method to reduce any imposed restrictions.

```
public MethodApp()
```

Methods

Restrictions()

Overrides the Restrictions method from the BOOSE.Method class. This method is intended to modify or implement custom restrictions. Currently, it is an empty implementation.

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

Class RealApp

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

```
public class RealApp : Real, ICommand
```

Inheritance

object ← Command ← Evaluation ← Real ← RealApp

Implements

ICommand

Inherited Members

Real.Compile() , Real.Execute() , 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

RealApp()

Initializes a new instance of the RealApp class. Calls the Restrictions() method to enforce restriction logic.

```
public RealApp()
```

Properties

Value

Gets or sets the value of the real number. This property overrides the base class property and uses the realValue field to store the value.

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

Property Value

[double](#)

Methods

Restrictions()

Enforces restriction logic by checking if the restriction count has been exceeded. Throws a BOOSE.RestrictionException if the count exceeds the limit.

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

Exceptions

RestrictionException

Thrown when the restriction count surpasses 50.

Class ResetApp

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

The `AppReset` class is a command that resets the canvas. It inherits from the `CanvasCommand` class.

```
public class ResetApp : CanvasCommand, ICommand
```

Inheritance

object ← Command ← CanvasCommand ← ResetApp

Implements

ICommand

Inherited Members

CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas , 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

ResetApp()

Initializes a new instance of the AppReset class. This constructor calls the base class constructor without parameters.

```
public ResetApp()
```

ResetApp(ICanvas)

Initializes a new instance of the AppReset class. This constructor allows you to provide a canvas object, which will be passed to the base class constructor.

```
public ResetApp(ICanvas c)
```

Parameters

`c` ICanvas

The ICanvas object that represents the canvas to be reset.

Methods

CheckParameters(string[])

Checks the parameters passed to the `AppReset` command. An exception is thrown indicating that the command does not accept any parameters.

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

Parameters

`parameter` string[]

An array of strings representing the parameters passed to the command.

Exceptions

ArgumentException

Thrown when more than one parameter is provided.

Execute()

Executes the reset command on the canvas. If the canvas is not null, it will call the `Reset` method on the canvas object. If the canvas is null, it will output a message indicating that the canvas is not set.

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

Class TriangleApp

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

A command to draw a triangle on the canvas. Inherits from CommandTwoParameters.

```
public class TriangleApp : CommandTwoParameters, ICommand
```

Inheritance

object ← Command ← CanvasCommand ← CommandOneParameter ← CommandTwoParameters ← TriangleApp

Implements

ICommand

Inherited Members

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

TriangleApp()

Default constructor.

```
public TriangleApp()
```

TriangleApp(Canvas, int, int)

Initializes a new instance of the TriangleApp class with specified parameters.

```
public TriangleApp(Canvas canvas, int width, int height)
```

Parameters

`canvas` Canvas

The canvas on which the triangle will be drawn.

`width` int

The width of the triangle.

`height` int

The height of the triangle.

Methods

CheckParameters(string[])

Checks if the provided parameters are valid for this command.

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

Parameters

`parameters` string[]

An array of strings representing the command parameters.

Exceptions

CommandException

Thrown if the parameters are null, do not contain exactly two elements, or are not positive integers.

Execute()

Executes the command to draw a triangle on the canvas. Parses the parameters and uses them to define the triangle's dimensions.

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

Class WhileApp

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

The WhileApp class inherits from the BOOSE.While class. This class demonstrates modifying the behavior of the parent class by overriding the constructor and the Restrictions method.

```
public class WhileApp : While, ICommand
```

Inheritance

object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← CompoundCommand ← While ← WhileApp

Implements

ICommand

Inherited Members

CompoundCommand.ReduceRestrictions() , CompoundCommand.CheckParameters(string[]) , CompoundCommand.Compile() , CompoundCommand.CorrespondingCommand , ConditionalCommand.endLineNumber , ConditionalCommand.Execute() , ConditionalCommand.EndLineNumber , ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType , ConditionalCommand.ReturnLineNumber , 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

WhileApp()

Initializes a new instance of the WhileApp class. Calls the ReduceRestrictions method to reduce any imposed restrictions.

```
public WhileApp()
```

Methods

Restrictions()

Overrides the Restrictions method from the BOOSE.While class. This method is intended to modify or implement custom restrictions. Currently, it is an empty implementation.

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

Class WriteApp

Namespace: ASE_Assignment_Demo

Assembly: ASE Assignment Demo.dll

A command to write text on the canvas. Inherits from CommandOneParameter.

```
public class WriteApp : CommandOneParameter, ICommand
```

Inheritance

object ← Command ← CanvasCommand ← CommandOneParameter ← WriteApp

Implements

ICommand

Inherited Members

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

WriteApp()

Default constructor.

```
public WriteApp()
```

WriteApp(AppCanvas)

Initializes a new instance of the WriteApp class with a specified canvas.

```
public WriteApp(AppCanvas canvas)
```

Parameters

`canvas` AppCanvas

The canvas on which the text will be written.

Methods

CheckParameters(string[])

Checks and validates the parameters provided to the command. Ensures the text parameter is not null, empty, or whitespace.

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

Parameters

`parameterList` string[]

An array of strings containing the command parameters.

Exceptions

CommandException

Thrown if the text is null, empty, or contains only whitespace.

Execute()

Executes the command to write text on the canvas.

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

Exceptions

CommandException

Thrown if the text is empty or the canvas is not an instance of.

Namespace ASE_Assignment_Test

Classes

AppCanvasTest

Unit tests for the AppCanvas class, covering its core functionality.

ArrayAppTests

Test class for ArrayApp. Contains unit tests to verify the behavior of the ArrayApp class.

IfAppTests

Unit test class for testing the functionality of the AppIf class. It includes tests for the Restrictions method and conditional execution in an if block.

IntAppTests

Test class for IntApp. Contains unit tests to verify the behavior and functionality of the IntApp class.

MethodAppTests

Unit test class for verifying the functionality of the MethodApp class. Includes tests for the Restrictions() method and constructor behavior related to reducing restrictions.

RealAppTests

Unit test class for verifying the functionality of the RealApp class. Includes tests for the Restrictions() method and the getter/setter behavior of the Value property.

Class AppCanvasTest

Namespace: ASE_Assignment_Test

Assembly: ASE_Assignment_Test.dll

Unit tests for the AppCanvas class, covering its core functionality.

```
[TestClass]
public class AppCanvasTest
```

Inheritance

object ← AppCanvasTest

Inherited Members

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

Methods

Circle_InvalidParameters()

```
[TestMethod]
[ExpectedException(typeof(CanvasException))]
public void Circle_InvalidParameters()
```

Circle_ValidParameters()

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

DrawTo_InvalidCoordinates_ThrowsCanvasException()

Tests that the DrawTo(int, int) method throws a BOOSE.CanvasException for invalid coordinates.

```
[TestMethod]
[ExpectedException(typeof(CanvasException))]
public void DrawTo_InvalidCoordinates_ThrowsCanvasException()
```

DrawTo_ValidCoordinates_UpdatesPenPosition()

Tests that the DrawTo(int, int) method updates the pen position correctly for valid coordinates.

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

MoveTo_BoundaryCoordinates_SetsCorrectPenPosition()

Tests that the MoveTo(int, int) method correctly sets the pen position at boundary values.

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

MoveTo_InvalidCoordinates_ThrowsCanvasException()

Tests that the MoveTo(int, int) method throws a BOOSE.CanvasException for invalid coordinates.

```
[TestMethod]
[ExpectedException(typeof(CanvasException))]
public void MoveTo_InvalidCoordinates_ThrowsCanvasException()
```

MoveTo_ValidCoordinates_SetsCorrectPenPosition()

Tests that the MoveTo(int, int) method correctly updates the pen position for valid coordinates.

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

Rect_InvalidDimensions_ThrowsCanvasException()

```
[TestMethod]
[ExpectedException(typeof(CanvasException))]
public void Rect_InvalidDimensions_ThrowsCanvasException()
```

Rect_ValidParameters()

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

SetColour_InvalidRGB_ThrowsException()

```
[TestMethod]
[ExpectedException(typeof(CanvasException))]
public void SetColour_InvalidRGB_ThrowsException()
```

SetColour_ValidRGB_SetsCorrectColor()

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

TestMultipleCommands_DrawShapesAndText()

Tests multiple commands such as drawing shapes and writing text sequentially on the canvas. Validates the correct pen position after the commands are executed.

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

Tri_InvalidDimensions_ThrowsCanvasException()

```
[TestMethod]
[ExpectedException(typeof(CanvasException))]
public void Tri_InvalidDimensions_ThrowsCanvasException()
```

Tri_ValidDimensions_DrawsCorrectTriangle()

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

WriteText_EmptyText_ThrowsCanvasException()

```
[TestMethod]
[ExpectedException(typeof(CanvasException))]
public void WriteText_EmptyText_ThrowsCanvasException()
```

WriteText_ValidText_WritesTextCorrectly()

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

Class ArrayAppTests

Namespace: ASE_Assignment_Test

Assembly: ASE_Assignment_Test.dll

Test class for ArrayApp. Contains unit tests to verify the behavior of the ArrayApp class.

```
[TestClass]
public class ArrayAppTests
```

Inheritance

object ← ArrayAppTests

Inherited Members

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

Methods

ArrayApp_Creation_ShouldNotThrowException()

Tests that an instance of the ArrayApp class can be created without throwing exceptions.

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

ArrayApp_MultipleInstances_ShouldNotThrowRestrictionException()

Tests that creating multiple instances of ArrayApp does not result in restriction exceptions.

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

Class IfAppTests

Namespace: ASE_Assignment_Test

Assembly: ASE_Assignment_Test.dll

Unit test class for testing the functionality of the AppIf class. It includes tests for the Restrictions method and conditional execution in an if block.

```
[TestClass]
public class IfAppTests
```

Inheritance

object ← IfAppTests

Inherited Members

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

Methods

AppIf_ConditionalExecution_Failure_ShouldFail()

Tests the conditional execution failure inside an if block in the AppIf class. Ensures the block does not execute when the condition is false.

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

AppIf_ConditionalExecution_ShouldPass()

Tests the conditional execution inside an if block in the AppIf class. Ensures the block is executed when the condition is true.

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

AppIf_Restrictions_ShouldNotThrowException()

Tests the Restrictions method in the AppIf class. Ensures that no exceptions are thrown when calling the Restrictions method.

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

Class IntAppTests

Namespace: ASE_Assignment_Test

Assembly: ASE_Assignment_Test.dll

Test class for IntApp. Contains unit tests to verify the behavior and functionality of the IntApp class.

```
[TestClass]
public class IntAppTests
```

Inheritance

object ← IntAppTests

Inherited Members

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

Methods

IntApp_InheritsBaseIntFunctionality_ShouldSetAndRetrieveValueCorrectly()

Tests that the IntApp class correctly inherits and implements functionality from the base BOOSE.Int class.

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

IntApp_LargeNumberOfInstancesCreation_ShouldNotThrowExceptions()

Tests that creating a large number of IntApp instances does not cause any errors or exceed restrictions.

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

IntApp_MultipleInstancesCreation_ShouldNotThrowExceptions()

Tests that multiple instances of IntApp can be created without throwing exceptions.

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

IntApp_RestrictionsMethodInvocation_ShouldNotThrowExceptions()

Tests that the Restrictions() method can be called without throwing exceptions.

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

IntApp_SetAndGetValue_ShouldStoreAndRetrieveIntegerValues()

Tests setting and getting the value of an IntApp instance.

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

Class MethodAppTests

Namespace: ASE_Assignment_Test

Assembly: ASE_Assignment_Test.dll

Unit test class for verifying the functionality of the MethodApp class. Includes tests for the Restrictions() method and constructor behavior related to reducing restrictions.

```
[TestClass]
public class MethodAppTests
```

Inheritance

object ← MethodAppTests

Inherited Members

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

Methods

MethodApp_Constructor_ShouldInvokeReduceRestrictions()

Tests that the constructor of MethodApp successfully invokes the restriction reduction logic without throwing exceptions.

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

MethodApp_RestrictionsMethod_ShouldNotThrowException()

Tests that the Restrictions() method can be called without throwing exceptions.

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

Class RealAppTests

Namespace: ASE_Assignment_Test

Assembly: ASE_Assignment_Test.dll

Unit test class for verifying the functionality of the RealApp class. Includes tests for the Restrictions() method and the getter/setter behavior of the Value property.

```
[TestClass]
public class RealAppTests
```

Inheritance

object ← RealAppTests

Inherited Members

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

Methods

RealApp_RestrictionsMethod_ShouldNotThrowException()

Ensures that calling the Restrictions() method does not throw any exceptions.

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

RealApp_ShouldCorrectlyInheritBaseClassFunctionality()

Verifies that the RealApp class inherits functionality from the BOOSE.Real base class and correctly implements it.

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

RealApp_ValueProperty_ShouldHandleRealValuesCorrectly()

Verifies that the Value property correctly handles setting and retrieving real (double) values.

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