Namespace ASE_SaruAcharya

Classes

AppCanvas

Represents a canvas for drawing shapes, lines, and text.

AppCommandFactory

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

AppTriangle

Represents a command to draw a triangle on a canvas.

AppWrite

Represents a command to write text on a canvas.

ArrayApp

Represents an array manipulation command in the application, derived from BOOSE.Evaluation. This class handles the creation, access, and modification of both integer and real arrays.

CompoundCommandApp

ConditionalCommandApp

ElseApp

Represents the Else command in the application, derived from CompoundCommandApp. This class handles the "else" functionality in a conditional structure.

EndApp

Represents an application-specific implementation of the CompoundCommandApp class that provides custom behavior for the "End" command.

ForApp

Represents an application-specific implementation of the BOOSE.For class, providing functionality to reset or decrease a private static field.

Form1

Represents the main form of the application, which serves as the user interface for drawing and running commands on a canvas.

IfApp

Represents an application that handles If commands, inheriting functionality from CompoundCommandApp.

IntApp

Represents an application that extends the functionality of the Int class.

MethodApp

Represents an application-specific method class that overrides restrictions on method counts and provides functionality to reset or decrease private static fields.

PeekApp

Represents an application that extends the functionality of the ArrayApp class to handle Peek operations.

PokeApp

Represents an application that extends the functionality of the ArrayApp class to handle Poke operations.

RealApp

Represents an application that extends the functionality of the Real class.

WhileApp

Represents an application for handling While commands, inheriting functionality from CompoundCommandApp.

Class AppCanvas

Namespace: ASE_SaruAcharya

Assembly: ASE_SaruAcharya.dll

Represents a canvas for drawing shapes, lines, and text.

```
public class AppCanvas : ICanvas
```

Inheritance

object ← AppCanvas

Implements

ICanvas

Inherited Members

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

Constructors

AppCanvas()

Initializes a new instance of the AppCanvas class with default settings.

```
public AppCanvas()
```

Properties

PenColour

Gets or sets the current pen color.

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

Property Value

object

Xpos

Gets or sets the current X-coordinate of the pen.

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

Property Value

int

Ypos

Gets or sets the current Y-coordinate of the pen.

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

Property Value

int

Methods

Circle(int, bool)

Draws a circle with the specified radius and fill option.

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

Parameters

`radius` int

The radius of the circle.

`filled` bool

If true, the circle is filled; otherwise, it is outlined.

Exceptions

CanvasException

Thrown if the radius is invalid.

Clear()

Clears the canvas with a gray background.

```
public void Clear()
```

DrawTo(int, int)

Draws a line from the current pen position to the specified position.

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

Parameters

`toX` int

The X-coordinate of the end point.

`toY` int

The Y-coordinate of the end point.

Exceptions

CanvasException

Thrown if the position is out of bounds.

MoveTo(int, int)

Moves the pen to the specified position without drawing.

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

Parameters

`x` int

The X-coordinate of the position.

`y` int

The Y-coordinate of the position.

Exceptions

CanvasException

Thrown if the position is out of bounds.

Rect(int, int, bool)

Draws a rectangle with the specified dimensions and fill option.

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

Parameters

`width` int

The width of the rectangle.

`height` int

The height of the rectangle.

`filled` bool

If true, the rectangle is filled; otherwise, it is outlined.

Exceptions

CanvasException

Thrown if the dimensions are invalid.

Reset()

Resets the pen position to the origin (0, 0).

```
public void Reset()
```

Set(int, int)

Resizes the canvas to the specified dimensions.

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

Parameters

`xsize` int

The width of the canvas.

`ysize` int

The 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

The red component (0-255).

`green` int

The green component (0-255).

`blue` int

The blue component (0-255).

Exceptions

CanvasException

Thrown if the RGB values are invalid.

Tri(int, int)

Draws a triangle with the specified base width and height.

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

Parameters

`width` int

The base width of the triangle.

`height` int

The height of the triangle.

WriteText(string)

Writes text at the current pen position.

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

Parameters

`text` string

The text to write.

Exceptions

CanvasException

Thrown if the text is null or empty.

getBitmap()

Gets the current bitmap of the canvas.

```
public object getBitmap()
```

Returns

object

The current bitmap.

Class AppCommandFactory

Namespace: ASE_SaruAcharya

Assembly: ASE_SaruAcharya.dll

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

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

Inheritance

object ← CommandFactory ← AppCommandFactory

Implements

ICommandFactory

Inherited Members

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

Constructors

AppCommandFactory()

Initializes a new instance of the AppCommandFactory class.

```
public AppCommandFactory()
```

Methods

MakeCommand(string)

Creates an instance of a command based on the specified command type.

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

Parameters

`commandType` string

The type of command to create (e.g., "tri", "write", "circle", "rect").

Returns

ICommand

An instance of the corresponding command if the command type matches a known command; otherwise, the base factory's command creation method is used.

Exceptions

CommandException

Thrown if the base factory fails to create a valid command for an unknown command type.

Class AppTriangle

Namespace: ASE_SaruAcharya

Assembly: ASE_SaruAcharya.dll

Represents a command to draw a triangle on a canvas.

```
public class AppTriangle : CommandTwoParameters, ICommand
```

Inheritance

object ← Command ← CanvasCommand ← CommandOneParameter ← CommandTwoParameters ← AppTriangle

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

AppTriangle()

Initializes a new instance of the AppTriangle class.

```
public AppTriangle()
```

AppTriangle(Canvas, int, int)

Initializes a new instance of the AppTriangle class with the specified canvas, width, and height.

```
public AppTriangle(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 the validity of the parameters provided to the triangle command.

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

Parameters

`parameterList` string[]

An array of parameters to validate.

Exceptions

CommandException

Thrown if the number of parameters is incorrect or if width/height are invalid.

Execute()

Executes the triangle drawing command by parsing parameters and drawing the triangle on the canvas.

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

Exceptions

CommandException

Thrown if the parameters are invalid.

Class AppWrite

Namespace: ASE_SaruAcharya

Assembly: ASE_SaruAcharya.dll

Represents a command to write text on a canvas.

```
public class AppWrite : CommandOneParameter, ICommand
```

Inheritance

object ← Command ← CanvasCommand ← CommandOneParameter ← AppWrite

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

AppWrite()

```
public AppWrite()
```

AppWrite(Canvas, string)

```
public AppWrite(Canvas c, string text)
```

Parameters

`c` Canvas

`text` string

Methods

CheckParameters(string[])

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

Parameters

`parameterList` string[]

Execute()

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

Class ArrayApp

Namespace: ASE_SaruAcharya

Assembly: ASE_SaruAcharya.dll

Represents an array manipulation command in the application, derived from BOOSE.Evaluation. This class handles the creation, access, and modification of both integer and real arrays.

```
public class ArrayApp : Evaluation, ICommand
```

Inheritance

object ← Command ← Evaluation ← ArrayApp

Implements

ICommand

Derived

PeekApp, PokeApp

Inherited Members

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()

Initializes a new instance of the ArrayApp class.

```
public ArrayApp()
```

Fields

IntValue

```
protected int IntValue
```

Field Value

int

PEEK

```
protected const bool PEEK = false
```

Field Value

bool

POKE

```
public const bool POKE = true
```

Field Value

bool

RealValue

```
protected double RealValue
```

Field Value

double

columnCurrent

```
protected int columnCurrent
```

Field Value

int

columnExpression

```
protected string columnExpression
```

Field Value

string

columnsCount

```
protected int columnsCount
```

Field Value

int

intArray

```
protected int[,] intArray
```

Field Value

int[,]

peekValue

```
protected string peekValue
```

Field Value

string

pokeValue

```
protected string pokeValue
```

Field Value

string

realArray

```
protected double[,] realArray
```

Field Value

double[,]

rowCurrent

```
protected int rowCurrent
```

Field Value

int

rowExpression

```
protected string rowExpression
```

Field Value

string

rowsCount

```
protected int rowsCount
```

Field Value

int

type

```
protected string type
```

Field Value

string

Properties

Columns

Gets the number of columns in the array.

```
protected int Columns { get; }
```

Property Value

int

Rows

Gets the number of rows in the array.

```
protected int Rows { get; }
```

Property Value

int

Methods

ArrayRestrictions()

Handles the restrictions related to array creation, though it's currently not implemented.

```
public void ArrayRestrictions()
```

CheckParameters(string[])

Checks the parameters provided for the array definition.

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

Parameters

`parameterList` string[]

The list of parameters to validate.

Exceptions

CommandException

Thrown if the parameters are invalid.

Compile()

Compiles the array definition, setting the type and dimensions of the array, and adding the variable to the program.

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

Execute()

Executes the array creation and initialization process based on the specified type.

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

GetIntArray(int, int)

Gets the integer value at the specified row and column in the integer array.

```
public virtual int GetIntArray(int row, int col)
```

Parameters

`row` int

The row index.

`col` int

The column index.

Returns

int

The value at the specified location.

GetRealArray(int, int)

Gets the real value at the specified row and column in the real array.

```
public virtual double GetRealArray(int row, int col)
```

Parameters

`row` int

The row index.

`col` int

The column index.

Returns

double

The value at the specified location.

ProcessArrayParametersCompile(bool)

Processes the array parameters during the compilation phase for a poke or peek operation.

```
protected virtual void ProcessArrayParametersCompile(bool isPokeOperation)
```

Parameters

`isPokeOperation` bool

Indicates whether the operation is a poke (set value) or peek (get value).

ProcessArrayParametersExecute(bool)

Processes the array parameters during the execution phase for a poke or peek operation.

```
protected virtual void ProcessArrayParametersExecute(bool isPokeOperation)
```

Parameters

`isPokeOperation` bool

Indicates whether the operation is a poke (set value) or peek (get value).

ReduceRestrictionCounter()

Reduces the restriction counter (not implemented).

```
protected void ReduceRestrictionCounter()
```

SetIntArray(int, int, int)

Sets a value in the integer array at the specified row and column.

```
public virtual void SetIntArray(int value, int row, int col)
```

Parameters

`value` int

The value to set.

`row` int

The row index.

`col` int

The column index.

SetRealArray(double, int, int)

Sets a value in the real array at the specified row and column.

```
public virtual void SetRealArray(double value, int row, int col)
```

Parameters

`value` double

The value to set.

`row` int

The row index.

`col` int

The column index.

Class CompoundCommandApp

Namespace: ASE_SaruAcharya

Assembly: ASE_SaruAcharya.dll

```
public class CompoundCommandApp : ConditionalCommandApp, ICommand
```

Inheritance

object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← ConditionalCommandApp ← CompoundCommandApp

Implements

ICommand

Derived

ElseApp, EndApp, IfApp, WhileApp

Inherited Members

ConditionalCommandApp.Execute() , 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

CompoundCommandApp()

Initializes a new instance of the CompoundCommandApp class without instance restrictions.

```
public CompoundCommandApp()
```

Remarks

This constructor removes any restrictions on the number of instances that can be created, enabling flexible use of compound commands.

Properties

CorrespondingCommand

Gets or sets the corresponding conditional command associated with this compound command.

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

Property Value

ConditionalCommand

Remarks

This property is used to establish a link between the compound command and its parent control flow command. For example, an "if-end" command will link to its "if" block.

Methods

CheckParameters(string[])

Validates the parameters passed to the compound command.

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

Parameters

`parameters` string[]

An array of parameters to validate.

Remarks

This method ensures that the compound command is associated with a valid control flow structure. It validates the following:

- Only one parameter is passed.
- The parameter contains an expected "end" marker, such as "ifEnd", "whileEnd", "forEnd", or "methodEnd".

If the validation fails, a BOOSE.CommandException is thrown.

Exceptions

CommandException

Thrown when the parameter count is invalid or the parameter does not contain a valid "end" marker.

Compile()

Compiles the compound command to prepare it for execution.

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

Remarks

This method calls the base BOOSE.ConditionalCommand.Compile() method to handle any setup or preparation required before the command is executed.

ResetOrDecreaseCount(int)

Resets or decreases the value of a private static field in the BOOSE.Boolean class.

```
public void ResetOrDecreaseCount(int newValue)
```

Parameters

`newValue` int

The new value to set for the private field.

Examples

```
var appMethod = new AppMethod();
appMethod.ResetOrDecreaseCount(5);
```

Remarks

This method accesses the private static field in the BOOSE.Boolean class using reflection and updates its value.

Exceptions

BOOSEException

Thrown when the private field cannot be accessed using reflection.

Class ConditionalCommandApp

Namespace: ASE_SaruAcharya

Assembly: ASE_SaruAcharya.dll

```
public class ConditionalCommandApp : ConditionalCommand, ICommand
```

Inheritance

object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← ConditionalCommandApp

Implements

ICommand

Derived

CompoundCommandApp

Inherited Members

ConditionalCommand.endLineNumber , ConditionalCommand.Compile() , 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)

Methods

Execute()

Executes the conditional command and resets the private static field to 0.

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

Examples

```
var command = new AppConditionalCommand();
command.Execute();
```

Remarks

This method overrides the BOOSE.ConditionalCommand.Execute() method to include resetting the value of the private static field to 0 after executing the base command.

ResetOrDecreaseCount(int)

Resets or decreases the value of a private static field in the BOOSE.Boolean class.

```
public void ResetOrDecreaseCount(int newValue)
```

Parameters

`newValue` int

The new value to set for the private field.

Examples

```
var command = new AppConditionalCommand();
command.ResetOrDecreaseCount(5);
```

Remarks

This method accesses the private static field in the BOOSE.Boolean class using reflection and updates its value.

Exceptions

BOOSEException

Thrown when the private field cannot be accessed using reflection.

Class ElseApp

Namespace: ASE_SaruAcharya

Assembly: ASE_SaruAcharya.dll

Represents the Else command in the application, derived from CompoundCommandApp. This class handles the "else" functionality in a conditional structure.

```
public class ElseApp : CompoundCommandApp, ICommand
```

Inheritance

object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← ConditionalCommandApp ← CompoundCommandApp ← ElseApp

Implements

ICommand

Inherited Members

CompoundCommandApp.CorrespondingCommand , CompoundCommandApp.ResetOrDecreaseCount(int) , 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

ElseApp()

Initializes a new instance of the ElseApp class.

```
public ElseApp()
```

Properties

CorrespondingEnd

Gets or sets the corresponding BOOSE.End object for this Else command.

```
public End CorrespondingEnd { get; set; }
```

Property Value

End

Methods

CheckParameters(string[])

Checks if the parameters provided are valid for the Else command. Expected parameter is "else".

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

Parameters

`parameters` string[]

The parameters to validate.

Exceptions

CommandException

Thrown if the parameters are invalid.

Compile()

Compiles the Else command by setting the corresponding command, line number, and end line number. It also pushes this command onto the program stack.

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

Execute()

Executes the Else command. If the corresponding condition is true, it sets the program counter to the end line number.

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

Class EndApp

Namespace: ASE_SaruAcharya

Assembly: ASE_SaruAcharya.dll

Represents an application-specific implementation of the CompoundCommandApp class that provides custom behavior for the "End" command.

```
public class EndApp : CompoundCommandApp, ICommand
```

Inheritance

object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← ConditionalCommandApp ← CompoundCommandApp ← EndApp

Implements

ICommand

Inherited Members

CompoundCommandApp.CorrespondingCommand , CompoundCommandApp.CheckParameters(string[]) , CompoundCommandApp.ResetOrDecreaseCount(int) , 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)

Remarks

This class handles the compilation and execution logic for the "End" command, including its interaction with corresponding commands such as "If", "While", and "For".

Constructors

EndApp()

Initializes a new instance of the EndApp class.

```
public EndApp()
```

Methods

Compile()

Compiles the "End" command by associating it with its corresponding command and validating the syntax based on the command type.

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

Exceptions

CommandException

Thrown if the syntax for the corresponding command type is invalid.

Execute()

Executes the "End" command, managing control flow based on the type of the corresponding command.

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

Exceptions

CommandException

Thrown if loop control variables are invalid or if loop steps prevent termination.

Class ForApp

Namespace: ASE_SaruAcharya

Assembly: ASE_SaruAcharya.dll

Represents an application-specific implementation of the BOOSE.For class, providing functionality to reset or decrease a private static field.

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

Inheritance

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

Implements

ICommand

Inherited Members

For.Compile() , 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)

Remarks

This class overrides the execution logic of the BOOSE.For class and includes methods to manipulate a private static field in the BOOSE.Boolean class using reflection.

Methods

Execute()

Executes the overridden logic and resets the private static field to 0.

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

Examples

```
var appFor = new ForApp();
appFor.Execute();
```

Remarks

This method overrides the BOOSE.For.Execute() method to include resetting the private static field to 0 after executing the base logic.

ResetOrDecreaseCount(int)

Resets or decreases the value of a private static field in the BOOSE.Boolean class.

```
public void ResetOrDecreaseCount(int newValue)
```

Parameters

`newValue` int

The new value to set for the private field.

Examples

```
var appFor = new ForApp();
appFor.ResetOrDecreaseCount(5);
```

Remarks

This method accesses the private static field in the BOOSE.Boolean class using reflection and updates its value.

Exceptions

BOOSEException

Thrown when the private field cannot be accessed using reflection.

Class Form1

Namespace: ASE_SaruAcharya

Assembly: ASE_SaruAcharya.dll

Represents the main form of the application, which serves as the user interface for drawing and running commands on a canvas.

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

Inheritance

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

Implements

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

Inherited Members

Form.SetVisibleCore(bool) , Form.Activate() , Form.ActivateMdiChild(Form) , Form.AddOwnedForm(Form) , Form.AdjustFormScrollbars(bool) , Form.Close() , Form.CreateAccessibilityInstance() , Form.CreateControlsInstance() , Form.CreateHandle() , Form.DefWndProc(ref Message) , Form.ProcessMnemonic(char) , Form.CenterToParent() , Form.CenterToScreen() , Form.LayoutMdi(MdiLayout) , Form.OnActivated(EventArgs) , Form.OnBackgroundImageChanged(EventArgs) , Form.OnBackgroundImageLayoutChanged(EventArgs) , Form.OnClosing(CancelEventArgs) , Form.OnClosed(EventArgs) , Form.OnFormClosing(FormClosingEventArgs) , Form.OnFormClosed(FormClosedEventArgs) , Form.OnCreateControl() , Form.OnDeactivate(EventArgs) , Form.OnEnabledChanged(EventArgs) , Form.OnEnter(EventArgs) , Form.OnFontChanged(EventArgs) , Form.OnGotFocus(EventArgs) , Form.OnHandleCreated(EventArgs) , Form.OnHandleDestroyed(EventArgs) , Form.OnHelpButtonClicked(CancelEventArgs) , Form.OnLayout(LayoutEventArgs) , Form.OnLoad(EventArgs) , Form.OnMaximizedBoundsChanged(EventArgs) , Form.OnMaximumSizeChanged(EventArgs) , Form.OnMinimumSizeChanged(EventArgs) , Form.OnInputLanguageChanged(InputLanguageChangedEventArgs) , Form.OnInputLanguageChanging(InputLanguageChangingEventArgs) , Form.OnVisibleChanged(EventArgs) , Form.OnMdiChildActivate(EventArgs) , Form.OnMenuStart(EventArgs) , Form.OnMenuComplete(EventArgs) , Form.OnPaint(PaintEventArgs) , Form.OnResize(EventArgs) , Form.OnDpiChanged(DpiChangedEventArgs) , Form.OnGetDpiScaledSize(int, int, ref Size) , Form.OnRightToLeftLayoutChanged(EventArgs) , Form.OnShown(EventArgs) , Form.OnTextChanged(EventArgs) , Form.ProcessCmdKey(ref Message, Keys) , Form.ProcessDialogKey(Keys) , Form.ProcessDialogChar(char) , Form.ProcessKeyPreview(ref Message) , Form.ProcessTabKey(bool) , Form.RemoveOwnedForm(Form) , Form.Select(bool, bool) , Form.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()

Initializes a new instance of the Form1 class. Sets up the canvas, command factory, program storage, and parser.

```
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_SaruAcharya

Assembly: ASE_SaruAcharya.dll

Represents an application that handles If commands, inheriting functionality from CompoundCommandApp.

```
public class IfApp : CompoundCommandApp, ICommand
```

Inheritance

object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← ConditionalCommandApp ← CompoundCommandApp ← IfApp

Implements

ICommand

Inherited Members

CompoundCommandApp.CorrespondingCommand , CompoundCommandApp.CheckParameters(string[]) , CompoundCommandApp.Compile() , CompoundCommandApp.ResetOrDecreaseCount(int) , ConditionalCommandApp.Execute() , 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

IfApp()

Initializes a new instance of the IfApp class.

```
public IfApp()
```

Methods

ReduceRestrictions()

Reduces restrictions associated with If commands.

```
protected void ReduceRestrictions()
```

Class IntApp

Namespace: ASE_SaruAcharya

Assembly: ASE_SaruAcharya.dll

Represents an application that extends the functionality of the Int 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)

Methods

Restrictions()

Applies specific restrictions for the IntApp.

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

Class MethodApp

Namespace: ASE_SaruAcharya

Assembly: ASE_SaruAcharya.dll

Represents an application-specific method class that overrides restrictions on method counts and provides functionality to reset or decrease private static fields.

```
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.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)

Remarks

This class demonstrates the use of reflection to manipulate private static fields in the BOOSE.Boolean and BOOSE.Method classes. It also overrides base class restrictions by invoking ReduceRestrictions.

Constructors

MethodApp()

Initializes a new instance of the MethodApp class.

```
public MethodApp()
```

Examples

```
var appMethod = new MethodApp();
```

Remarks

The constructor overrides restrictions on method count by invoking ReduceRestrictions twice. It also resets the counts for the fields 꾶 and 꾶 to 0.

Methods

ResetOrDecreaseCount(int)

Resets or decreases the value of a private static field in the BOOSE.Boolean class.

```
public void ResetOrDecreaseCount(int newValue)
```

Parameters

`newValue` int

The new value to set for the private field.

Examples

```
var appMethod = new MethodApp();
appMethod.ResetOrDecreaseCount(5);
```

Remarks

This method accesses the private static field 븀 in the BOOSE.Boolean class using reflection and updates its value.

Exceptions

BOOSEException

Thrown when the private field 뱸 cannot be accessed using reflection.

ResetOrDecreaseMethodCount(int)

Resets or decreases the value of a private static field in the BOOSE.Method class.

```
public void ResetOrDecreaseMethodCount(int newValue)
```

Parameters

`newValue` int

The new value to set for the private field.

Examples

```
var appMethod = new MethodApp();
appMethod.ResetOrDecreaseMethodCount(10);
```

Remarks

This method accesses the private static field 뵐 in the BOOSE.Method class using reflection and updates its value.

Exceptions

BOOSEException

Thrown when the private field 뵐 cannot be accessed using reflection.

Class PeekApp

Namespace: ASE_SaruAcharya

Assembly: ASE_SaruAcharya.dll

Represents an application that extends the functionality of the ArrayApp class to handle Peek operations.

```
public class PeekApp : ArrayApp, ICommand
```

Inheritance

object ← Command ← Evaluation ← ArrayApp ← PeekApp

Implements

ICommand

Inherited Members

ArrayApp.PEEK , ArrayApp.POKE , ArrayApp.type , ArrayApp.IntValue , ArrayApp.RealValue , ArrayApp.rowsCount , ArrayApp.columnsCount , ArrayApp.intArray , ArrayApp.realArray , ArrayApp.pokeValue , ArrayApp.peekValue , ArrayApp.rowExpression , ArrayApp.columnExpression , ArrayApp.rowCurrent , ArrayApp.columnCurrent , ArrayApp.Rows , ArrayApp.Columns , ArrayApp.ArrayRestrictions() , ArrayApp.ReduceRestrictionCounter() , ArrayApp.ProcessArrayParametersCompile(bool) , ArrayApp.ProcessArrayParametersExecute(bool) , ArrayApp.SetIntArray(int, int, int) , ArrayApp.SetRealArray(double, int, int) , ArrayApp.GetIntArray(int, int) , ArrayApp.GetRealArray(int, int) , 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

PeekApp()

Initializes a new instance of the PeekApp class.

```
public PeekApp()
```

Methods

CheckParameters(string[])

Validates the parameters for the Peek operation.

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

Parameters

`parameters` string[]

The array of parameters to validate.

Exceptions

NotImplementedException

Thrown when parameter validation is not implemented.

Compile()

Compiles the Peek operation by processing array parameters.

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

Execute()

Executes the Peek operation by processing array parameters and updating variables.

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

Exceptions

CommandException

Thrown when the array type is unsupported.

Class PokeApp

Namespace: ASE_SaruAcharya

Assembly: ASE_SaruAcharya.dll

Represents an application that extends the functionality of the ArrayApp class to handle Poke operations.

```
public class PokeApp : ArrayApp, ICommand
```

Inheritance

object ← Command ← Evaluation ← ArrayApp ← PokeApp

Implements

ICommand

Inherited Members

ArrayApp.PEEK , ArrayApp.POKE , ArrayApp.type , ArrayApp.IntValue , ArrayApp.RealValue , ArrayApp.rowsCount , ArrayApp.columnsCount , ArrayApp.intArray , ArrayApp.realArray , ArrayApp.pokeValue , ArrayApp.peekValue , ArrayApp.rowExpression , ArrayApp.columnExpression , ArrayApp.rowCurrent , ArrayApp.columnCurrent , ArrayApp.Rows , ArrayApp.Columns , ArrayApp.ArrayRestrictions() , ArrayApp.ReduceRestrictionCounter() , ArrayApp.ProcessArrayParametersCompile(bool) , ArrayApp.ProcessArrayParametersExecute(bool) , ArrayApp.SetIntArray(int, int, int) , ArrayApp.SetRealArray(double, int, int) , ArrayApp.GetIntArray(int, int) , ArrayApp.GetRealArray(int, int) , 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.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

PokeApp()

Initializes a new instance of the PokeApp class.

```
public PokeApp()
```

Methods

CheckParameters(string[])

Validates the parameters for the Poke operation.

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

Parameters

`parameter` string[]

The array of parameters to validate.

Exceptions

CommandException

Thrown when the number of parameters is invalid.

Compile()

Compiles the Poke operation by processing array parameters.

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

Execute()

Executes the Poke operation by processing array parameters.

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

Set(StoredProgram, string)

Sets up the PokeApp with the specified program and parameters list.

```
public override void Set(StoredProgram program, string paramsList)
```

Parameters

`program` StoredProgram

The program to associate with this operation.

`paramsList` string

The list of parameters to set.

Class RealApp

Namespace: ASE_SaruAcharya

Assembly: ASE_SaruAcharya.dll

Represents an application that extends the functionality of the Real class.

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

Inheritance

object ← Command ← Evaluation ← Real ← RealApp

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)

Methods

Restrictions()

Overrides the Restrictions method to define specific restrictions for RealApp.

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

Class WhileApp

Namespace: ASE_SaruAcharya

Assembly: ASE_SaruAcharya.dll

Represents an application for handling While commands, inheriting functionality from CompoundCommandApp.

```
public class WhileApp : CompoundCommandApp, ICommand
```

Inheritance

object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← ConditionalCommandApp ← CompoundCommandApp ← WhileApp

Implements

ICommand

Inherited Members

CompoundCommandApp.CorrespondingCommand , CompoundCommandApp.CheckParameters(string[]) , CompoundCommandApp.Compile() , CompoundCommandApp.ResetOrDecreaseCount(int) , ConditionalCommandApp.Execute() , 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

WhileApp()

Initializes a new instance of the WhileApp class.

```
public WhileApp()
```

Methods

ReduceRestrictions()

Reduces restrictions associated with While commands.

```
protected void ReduceRestrictions()
```

Namespace ASE_SaruAcharya.Tests

Classes

AppCanvasTests

Unit tests for the AppCanvas class to verify its behavior and ensure robustness.

Class AppCanvasTests

Namespace: ASE_SaruAcharya.Tests

Assembly: UnitTesting.dll

Unit tests for the AppCanvas class to verify its behavior and ensure robustness.

```
[TestClass]
public class AppCanvasTests
```

Inheritance

object ← AppCanvasTests

Inherited Members

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

Methods

Array_Null_Test()

Tests to ensure that Array parameters are not null.

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

Circle_InvalidRadius_ThrowsException()

Tests the Circle(int, bool) method with an invalid radius and expects an exception.

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

Circle_ValidRadius_DrawsCircle()

Tests the Circle(int, bool) method with a valid radius.

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

Clear_CanvasCleared_Successfully()

Tests the Clear() method to ensure the canvas is cleared successfully.

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

Else_Conditional_Test()

Tests that AppElse is of the correct type ConditionalCommand.

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

End_Conditional_Test()

Tests that AppEnd is of the correct type ConditionalCommand.

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

For_Null_Test()

Tests to ensure that For parameters are not null.

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

If_Conditional_Test()

Tests that AppIf is of the correct type ConditionalCommand.

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

If_Null_Test()

Tests to ensure that If parameters are not null.

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

Int_Restrictions_Test()

Tests that the Restrictions method for Int does not throw exceptions.

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

Int_Test_Constructor()

Tests the successful creation of an Int instance.

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

Method_Null_Test()

Tests to ensure that Method parameters are not null.

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

MoveTo_InvalidPosition_ThrowsException()

Tests the MoveTo(int, int) method with invalid coordinates and expects an exception.

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

MoveTo_ValidPosition_UpdatesPosition()

Tests the MoveTo(int, int) method with valid coordinates.

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

Real_Null_Test()

Tests to ensure that Real parameters are not null.

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

Rect_InvalidDimensions_ThrowsException()

Tests the Rect(int, int, bool) method with invalid dimensions and expects an exception.

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

Rect_ValidDimensions_DrawsRectangle()

Tests the Rect(int, int, bool) method with valid dimensions.

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

SetColour_InvalidValues_ThrowsException()

Tests the SetColour(int, int, int) method with invalid color values and expects an exception.

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

SetColour_ValidValues_UpdatesPenColour()

Tests the SetColour(int, int, int) method with valid color values.

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

Setup()

Initializes resources before each test is executed.

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

Tri_ValidDimensions_DrawsTriangle()

Tests the Tri(int, int) method with valid dimensions.

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

While_Null_Test()

Tests to ensure that While parameters are not null.

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

WriteText_NullOrEmptyText_ThrowsException()

Tests the WriteText(string) method with null or empty text input and expects an exception.

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

WriteText_ValidText_DrawsText()

Tests the WriteText(string) method with valid text input.

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