Namespace ASE_Assignment

Classes

AppArray

Represents a custom implementation of the Array command in the application.

AppCanvas

Represents a canvas for drawing shapes and managing drawing properties. Implements the BOOSE.ICanvas interface.

AppClear

Represents a command to clear the canvas in the application.

AppCommandFactory

Factory class for creating specific command objects based on the command type. Inherits from CommandFactory and overrides the MakeCommand method.

AppDecrement

Command that decrements the value of a specified integer variable. Inherits from the BOOSE.Command class.

AppElse

Represents a custom implementation of the "else" command in the application.

AppEnd

Represents a custom implementation of the "end" command in the application.

AppFor

Represents a custom implementation of the "for" loop command in the application.

AppIf

Represents a custom implementation of the "if" command in the application.

AppInt

Represents the singleton instance of the AppInt class that extends BOOSE.Int.

AppMethod

Represents a custom implementation of the "method" command in the application.

AppReal

Represents a custom implementation of the "real" command in the application.

AppReset

Represents a command to reset the canvas.

AppTri

Represents the triangle drawing command in the application. Inherits from BOOSE.CommandTwo Parameters and handles drawing triangles on the canvas.

AppWhile

Represents a custom implementation of a "while" loop command in the application.

AppWrite

Represents a command to write processed text on a canvas.

Form1

Represents the main form of the application, which handles user interactions.

Class AppArray

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a custom implementation of the Array command in the application.

```
public class AppArray : Array, ICommand
```

Inheritance

object ← Command ← Evaluation ← Array ← AppArray

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.CheckParameters(string[]) , 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)

Remarks

This class extends the BOOSE.Array class to provide additional functionality or customization for handling arrays within the application. It reduces the restriction counter and utilizes the base class's compile and execute logic.

Constructors

AppArray()

Initializes a new instance of the AppArray class.

```
public AppArray()
```

Remarks

The constructor reduces the restriction counter, allowing the array-related commands to operate with fewer constraints compared to the base implementation.

Methods

Compile()

Compiles the Array command.

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

Remarks

This method invokes the base class's BOOSE.Array.Compile() method to handle the compilation of array-related operations.

Execute()

Executes the Array command.

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

Remarks

This method invokes the base class's BOOSE.Array.Execute() method to execute the logic associated with array operations within the application.

Class AppCanvas

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a canvas for drawing shapes and managing drawing properties. Implements the BOOSE.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()

Initializes a new instance of the AppCanvas class.

```
public AppCanvas()
```

Properties

PenColour

Gets or sets the pen color used for drawing.

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

Property Value

object

Exceptions

CanvasException

Thrown when the provided color is invalid.

Xpos

Gets or sets the current x-coordinate position on the canvas.

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

Property Value

int

Ypos

Gets or sets the current y-coordinate position on the canvas.

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

Property Value

int

Methods

Circle(int, bool)

Draws a circle on the canvas.

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

Parameters

`radius` int

The radius of the circle to draw.

`filled` bool

Indicates whether the circle should be filled or just outlined.

Exceptions

CanvasException

Thrown if the radius is invalid or drawing fails.

Clear()

Clears the canvas by filling it with white color.

```
public void Clear()
```

Exceptions

CanvasException

Thrown if the canvas cannot be cleared.

DrawTo(int, int)

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

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

Parameters

`x` int

The x-coordinate of the endpoint.

y int

The y-coordinate of the endpoint.

Exceptions

CanvasException

Thrown if the position is invalid or drawing fails.

MoveTo(int, int)

Moves the drawing position to the specified coordinates.

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

Parameters

x int

The x-coordinate of the new position.

y int

The y-coordinate of the new position.

Exceptions

CanvasException

Thrown if the position is invalid.

Rect(int, int, bool)

Draws a rectangle at the current position with specified dimensions.

```
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

Indicates whether the rectangle should be filled or just outlined.

Exceptions

CanvasException

Thrown if dimensions are invalid or drawing fails.

Reset()

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

```
public void Reset()
```

Set(int, int)

Sets the size of the canvas and initializes the graphics object.

```
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 color values are invalid.

Tri(int, int)

Draws a triangle at the current position with specified dimensions.

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

Parameters

`width` int

The width of the triangle's base.

`height` int

The height of the triangle.

Exceptions

CanvasException

Thrown if dimensions are invalid or drawing fails.

WriteText(string)

Writes text at the current position.

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

Parameters

`text` string

The text to write.

Exceptions

ArgumentException

Thrown if the text is null or empty.

getBitmap()

Retrieves the bitmap representing the current state of the canvas.

```
public object getBitmap()
```

Returns

object

The bitmap object of the canvas.

Class AppClear

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a command to clear the canvas in the application.

```
public class AppClear : CanvasCommand, ICommand
```

Inheritance

object ← Command ← CanvasCommand ← AppClear

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)

Remarks

This class extends the BOOSE.CanvasCommand class and provides functionality to clear the canvas. It overrides methods for executing the command and checking parameters.

Methods

CheckParameters(string[])

Validates the parameters for the "clear canvas" command.

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

Parameters

`parameter` string[]

An array of parameters passed to the command.

Remarks

This method ensures that the command is invoked with the correct parameters. For this command, at least one parameter is expected to avoid throwing a `BOOSE.CommandException`.

Exceptions

CommandException

Thrown if the parameter array is empty or invalid for the "clear canvas" command.

Execute()

Executes the "clear canvas" command.

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

Remarks

This method calls the base class's `Execute` method to ensure any pre-execution logic is processed, and then clears the canvas using the `Canvas.Clear` method.

Class AppCommandFactory

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Factory class for creating specific command objects based on the command type. Inherits from CommandFactory and overrides the MakeCommand method.

```
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 a command object based on the given command type.

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

Parameters

`commandType` string

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

Returns

ICommand

An ICommand object representing the specified command type.

Remarks

This method overrides the MakeCommand method from the CommandFactory class to return specific command implementations based on the commandType argument. If the commandType is not recognized, the base class implementation is used.

Class AppDecrement

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Command that decrements the value of a specified integer variable. Inherits from the BOOSE.Command class.

```
public class AppDecrement : Command, ICommand
```

Inheritance

object ← Command ← AppDecrement

Implements

ICommand

Inherited Members

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

Methods

CheckParameters(string[])

Checks if the provided parameters are valid for the decrement operation. Validates the existence and type of the specified variable.

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

Parameters

`parameters` string[]

An array of parameters where the first element is the variable name to decrement.

Execute()

Executes the decrement command on the specified variable. Decreases the value of the variable by one.

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

Class AppElse

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a custom implementation of the "else" command in the application.

```
public class AppElse : Else, ICommand
```

Inheritance

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

Implements

ICommand

Inherited Members

Else.CheckParameters(string[]) , 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)

Remarks

This class extends the BOOSE.Else class, enabling the customization of behavior and restrictions for the "else" command. It is primarily used in conditional structures to handle alternative logic when the associated condition is not met.

Constructors

AppElse()

Initializes a new instance of the AppElse class.

```
public AppElse()
```

Remarks

This constructor invokes the `ReduceRestrictions` method to relax the default restrictions defined in the base BOOSE.Else class.

Methods

Compile()

Compiles the logic for the "else" command.

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

Remarks

This method calls the base class's BOOSE.Else.Compile() method to process the compilation of the "else" command logic, ensuring that it is properly integrated into the conditional structure of the application.

Execute()

Executes the "else" command.

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

Remarks

This method invokes the base class's BOOSE.Else.Execute() method to perform the execution of the alternative logic defined for the "else" branch.

Restrictions()

Defines restrictions for the "else" command.

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

Remarks

This method overrides the base class's restriction logic but does not implement any additional restrictions, leaving the "else" command unrestricted.

Class AppEnd

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a custom implementation of the "end" command in the application.

```
public class AppEnd : End, ICommand
```

Inheritance

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

Implements

ICommand

Inherited Members

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)

Remarks

This class extends the BOOSE.End class, allowing for customized behavior and relaxation of restrictions for the "end" command.

Constructors

AppEnd()

Initializes a new instance of the AppEnd class.

```
public AppEnd()
```

Remarks

The constructor automatically invokes the `ReduceRestrictions` method to handle the compilation of the method logic as defined in the application.

Methods

Compile()

Compiles the logic for the "end" command.

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

Remarks

This method calls the base class's BOOSE.End.Compile() method to process the compilation of the "end" command logic, ensuring proper functionality in the application.

Execute()

Executes the "end" command.

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

Remarks

This method calls the base class's BOOSE.End.Execute() method to carry out the execution of the "end" command, terminating the current logic or flow.

Restrictions()

Defines restrictions for the "end" command.

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

Remarks

This method overrides the base class's restriction logic but does not implement any additional restrictions, leaving the "end" command unrestricted.

Class AppFor

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a custom implementation of the "for" loop command in the application.

```
public class AppFor : For, ICommand
```

Inheritance

object ← Command ← Evaluation ← Boolean ← ConditionalCommand ← For ← AppFor

Implements

ICommand

Inherited Members

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)

Remarks

This class extends the BOOSE.For class, enabling loop functionality with customizable behavior for compilation, execution, and restrictions.

Constructors

AppFor()

Initializes a new instance of the AppFor class.

```
public AppFor()
```

Remarks

This constructor initializes the loop command without any specific customization. Further custom logic can be added as needed.

Methods

Compile()

Compiles the logic for the "for" loop command.

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

Remarks

This method invokes the base class's BOOSE.For.Compile() method to handle the compilation of the "for" loop logic, ensuring that the loop operates correctly in the application.

Execute()

Executes the "for" loop command.

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

Remarks

This method invokes the base class's BOOSE.For.Execute() method to perform the execution of the "for" loop logic based on the provided parameters.

Restrictions()

Defines restrictions for the "for" loop command.

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

Remarks

This method overrides the base class's restriction logic but does not implement any additional restrictions, leaving the "for" loop command unrestricted.

Class AppIf

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a custom implementation of the "if" command in the application.

```
public class AppIf : If, ICommand
```

Inheritance

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

Implements

ICommand

Inherited Members

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)

Remarks

This class extends the BOOSE.If class, enabling conditional execution with reduced restrictions and customized behavior for compilation and execution.

Constructors

AppIf()

Initializes a new instance of the AppIf class.

```
public AppIf()
```

Remarks

This constructor automatically invokes the `ReduceRestrictions` method to handle the default restrictions defined in the base BOOSE.If class.

Methods

Compile()

Compiles the logic for the "if" command.

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

Remarks

This method invokes the base class's `Compile` method to handle the compilation of the "if" logic as per the application's requirements.

Execute()

Executes the logic for the "if" command.

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

Remarks

This method invokes the base class's `Execute` method to perform the execution of the "if" logic as per the application's functionality.

Restrictions()

Defines restrictions for the "if" command.

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

Remarks

This method overrides the base class's restriction logic but does not implement any additional restrictions, leaving the "if" command unrestricted.

Class AppInt

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents the singleton instance of the AppInt class that extends BOOSE.Int.

```
public sealed class AppInt : Int, ICommand
```

Inheritance

object ← Command ← Evaluation ← Int ← AppInt

Implements

ICommand

Inherited Members

Evaluation.CheckParameters(string[]) , Evaluation.ProcessExpression(string) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value , Evaluation.Local , 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.ReferenceEquals(object, object)

Constructors

AppInt()

Initializes a new instance of the AppInt class.

```
public AppInt()
```

Properties

Instance

Gets the singleton instance of the AppInt class.

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

Property Value

AppInt

The singleton instance of AppInt.

Methods

Compile()

Compiles the current instance by calling the base Compile() method.

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

Execute()

Executes the current instance by calling the base Execute() method.

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

Restrictions()

Defines restrictions for the "else" command.

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

Remarks

This method overrides the base class's restriction logic but does not implement any additional restrictions, leaving the "else" command unrestricted.

Class AppMethod

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a custom implementation of the "method" command in the application.

```
public class AppMethod : Method, ICommand
```

Inheritance

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

Implements

ICommand

Inherited Members

Method.CheckParameters(string[]) , 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)

Remarks

This class extends the BOOSE.Method class, allowing for custom behavior related to compilation, execution, and restrictions of a method command as required by the application. Restrictions are reduced during initialization.

Constructors

AppMethod()

Initializes a new instance of the AppMethod class.

```
public AppMethod()
```

Remarks

The constructor automatically invokes the `ReduceRestrictions` method to relax any restrictions defined by the base BOOSE.Method class.

Methods

Compile()

Compiles the "method" command logic.

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

Remarks

This method invokes the base class's BOOSE.Method.Compile() method to handle the compilation of the method logic as defined in the application.

Execute()

Executes the "method" command.

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

Remarks

This method invokes the base class's BOOSE.Method.Execute() method to perform the execution of the method command.

Restrictions()

Defines restrictions for the "method" command.

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

Remarks

This method overrides the base class's restriction logic but does not implement any additional restrictions, leaving the "method" command unrestricted.

Class AppReal

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a custom implementation of the "real" command in the application.

```
public class AppReal : Real, ICommand
```

Inheritance

object ← Command ← Evaluation ← Real ← AppReal

Implements

ICommand

Inherited Members

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)

Remarks

This class extends the BOOSE.Real class, allowing for custom behavior related to compilation, execution, and restrictions of the "real" command as required by the application.

Constructors

AppReal()

Initializes a new instance of the AppReal class.

```
public AppReal()
```

Remarks

The constructor is provided for any specific initialization needs of the "real" command. Currently, no additional setup is required.

Methods

Compile()

Compiles the "real" command logic.

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

Remarks

This method invokes the base class's BOOSE.Real.Compile() method to handle the compilation process for the "real" command.

Execute()

Executes the "real" command.

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

Remarks

This method invokes the base class's BOOSE.Real.Execute() method to perform the execution of the "real" command as defined in the application's logic.

Restrictions()

Defines restrictions for the "real" command.

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

Remarks

This method overrides the base class's restriction logic but does not implement any additional restrictions, leaving the "real" command unrestricted.

Class AppReset

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a command to reset the canvas.

```
public class AppReset : CanvasCommand, ICommand
```

Inheritance

object ← Command ← CanvasCommand ← AppReset

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)

Methods

CheckParameters(string[])

Validates the parameters for the reset command.

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

Parameters

`parameter` string[]

The array of parameters to validate.

Exceptions

CommandException

Thrown when the parameter array is empty or invalid for this command.

Execute()

Executes the reset command by invoking the canvas's reset functionality.

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

Exceptions

CanvasException

Thrown when the base canvas reset operation fails.

Class AppTri

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents the triangle drawing command in the application. Inherits from BOOSE.CommandTwo Parameters and handles drawing triangles on the canvas.

```
public class AppTri : CommandTwoParameters, ICommand
```

Inheritance

object ← Command ← CanvasCommand ← CommandOneParameter ← CommandTwoParameters ← AppTri

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

AppTri()

Initializes a new instance of the AppTri class. This constructor is the default constructor with no parameters.

```
public AppTri()
```

AppTri(Canvas, int, int)

Initializes a new instance of the AppTri class with the specified canvas, width, and height. This constructor is used to initialize the triangle drawing command with the given canvas and dimensions.

```
public AppTri(Canvas c, int width, int height)
```

Parameters

`c` Canvas

The canvas on which to draw the triangle.

`width` int

The width of the triangle.

`height` int

The height of the triangle.

Methods

CheckParameters(string[])

Validates the parameters for the triangle drawing command. Ensures that exactly two parameters are passed to the command.

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

Parameters

`parameterList` string[]

An array of strings representing the parameters for the triangle drawing command.

Remarks

This method checks that the number of parameters is exactly two, and validates that both width and height are positive integers.

Exceptions

CommandException

Thrown if the number of parameters is incorrect or invalid.

Execute()

Executes the triangle drawing command. Parses the parameters for width and height, and then calls the Tri(int, int) method to draw the triangle on the canvas.

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

Remarks

This method uses the parsed width and height parameters to draw a triangle on the canvas.

Class AppWhile

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a custom implementation of a "while" loop command in the application.

```
public class AppWhile : While, ICommand
```

Inheritance

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

Implements

ICommand

Inherited Members

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)

Remarks

This class extends the BOOSE.While class, providing functionality for compilation, execution, and restriction handling specific to the application requirements.

Constructors

AppWhile()

Initializes a new instance of the AppWhile class.

```
public AppWhile()
```

Remarks

This constructor reduces restrictions to allow for more flexibility in the while-loop logic. The `ReduceRestrictions` method is invoked to relax the default constraints.

Methods

Compile()

Compiles the "while" loop command.

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

Remarks

This method invokes the base class's `Compile` method to handle the compilation of the "while" loop logic as required by the application.

Execute()

Executes the "while" loop command.

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

Remarks

This method invokes the base class's `Execute` method to execute the "while" loop logic defined in the application.

Restrictions()

Defines restrictions for the "while" command.

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

Remarks

This method overrides the base class's restriction logic but does not implement any additional restrictions, leaving the "while" command unrestricted.

Class AppWrite

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a command to write processed 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)

Remarks

This class processes input text, validates it, and writes it onto a canvas.

Constructors

AppWrite()

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

```
public AppWrite()
```

Remarks

This constructor is used when the canvas and text will be set later.

AppWrite(Canvas, string)

Initializes a new instance of the AppWrite class with the specified canvas and text.

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

Parameters

`c` Canvas

The BOOSE.Canvas object where the text will be written.

`text` string

The text to process and write onto the canvas.

Remarks

Use this constructor when the canvas and text are known during initialization.

Methods

CheckParameters(string[])

Validates the parameters passed to the command.

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

Parameters

`parameterList` string[]

An array of parameters to validate.

Remarks

This method ensures that exactly one valid parameter (non-null, non-empty, and non-whitespace text) is provided for processing.

Exceptions

CommandException

Thrown when the parameter list is null, has an incorrect number of parameters, or contains invalid text.

Execute()

Executes the command to process and write text on the canvas.

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

Remarks

This method validates the text, processes it to handle expressions or concatenations, and writes the result to the canvas. Any exceptions are logged for debugging purposes.

Exceptions

RestrictionException

Thrown when the text is null, empty, contains only whitespace, or exceeds 100 characters.

Class Form1

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents the main form of the application, which handles user interactions.

```
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, and parser for the application.

```
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.

Namespace TestProject

Classes

Test1

Tests for the functionality of drawing and setting properties on the canvas.

Test1.AppArrayTest

Test1.AppElseTest

Test1.AppEndTest

Test1.AppForTest

Test1.AppIfTest

Test1.AppIntTest

Test1.AppMethodTest

Test1.AppRealTest

Test1.AppwhileTest

Class Test1

Namespace: TestProject

Assembly: TestProject1.dll

Tests for the functionality of drawing and setting properties on the canvas.

```
[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

Clear_Canvas()

Tests the behavior of clearing the canvas. Verifies that the canvas is cleared correctly.

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

Setup()

Initializes the canvas before each test.

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

Test_Circle_InvalidParameters()

Tests drawing a circle with invalid parameters. Verifies that a CanvasException is thrown.

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

Test_Circle_ValidParameters()

Tests drawing a circle with valid parameters. Verifies that the position remains unchanged after drawing the circle.

```
public void Test_Circle_ValidParameters()
```

Test_DrawTo_InvalidCoordinates()

Tests the DrawTo method with invalid coordinates. Verifies that a CanvasException is thrown.

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

Test_DrawTo_ValidCoordinates()

Tests the DrawTo method with valid coordinates. Verifies that the canvas position is updated correctly.

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

Test_Invalid_Text()

```
[TestMethod]
[ExpectedException(typeof(CommandException))]
public void Test_Invalid_Text()
```

Test_MoveTo_InvalidCoordinates()

Tests the MoveTo method with invalid coordinates. Verifies that a CanvasException is thrown.

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

Test_MoveTo_ValidCoordinates()

Tests the MoveTo method with valid coordinates. Verifies that the canvas position is updated correctly.

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

Test_MultilineDrawingProgram()

Tests a multiline drawing program. Verifies the sequence of drawing operations in the canvas, ensuring that shapes are drawn correctly and colors are applied as expected across multiple moves and drawings.

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

Test_Rect_InvalidParameters()

Tests the CheckParameters method of AppTri with invalid parameters. Verifies that a CommandException is thrown.

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

Test_Rect_ValidParameters()

Tests drawing a Rectangle with valid parameters. Verifies that the position remains unchanged after drawing the circle.

```
public void Test_Rect_ValidParameters()
```

Test_Reset()

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

Test_SetColour_InvalidColourRange()

Tests setting an invalid color range (out of bounds). Verifies that a CanvasException is thrown when any of the RGB values are outside the valid range (0-255).

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

Test_SetColour_ValidColour()

Tests setting a valid color value for the pen on the canvas. Verifies that the color is set correctly.

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

Test_TriInvalidParameter()

Tests the CheckParameters method of AppTri with invalid parameters. Verifies that a CommandException is thrown.

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

Test_Tri_ValidParameters()

Tests the CheckParameters method of AppTri with valid parameters. Verifies that no exceptions are thrown.

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

Test_ValidText()

Tests the CheckParameters method of AppWrite with valid text. Verifies that no exceptions are thrown.

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

Class Test1.AppArrayTest

Namespace: TestProject

Assembly: TestProject1.dll

```
[TestClass]
public class Test1.AppArrayTest
```

Inheritance

object ← Test1.AppArrayTest

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()

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

Array_Restrictions_Test()

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

Class Test1.AppElseTest

Namespace: TestProject

Assembly: TestProject1.dll

```
[TestClass]
public class Test1.AppElseTest
```

Inheritance

object ← Test1.AppElseTest

Inherited Members

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

Methods

Else_Conditional_Test()

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

Class Test1.AppEndTest

Namespace: TestProject

Assembly: TestProject1.dll

```
[TestClass]
public class Test1.AppEndTest
```

Inheritance

object ← Test1.AppEndTest

Inherited Members

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

Methods

End_Conditional_Test()

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

Class Test1.AppForTest

Namespace: TestProject

Assembly: TestProject1.dll

```
[TestClass]
public class Test1.AppForTest
```

Inheritance

object ← Test1.AppForTest

Inherited Members

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

Methods

For_Null_Test()

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

For_Restrictions_Test()

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

Class Test1.AppIfTest

Namespace: TestProject

Assembly: TestProject1.dll

```
[TestClass]
public class Test1.AppIfTest
```

Inheritance

object ← Test1.AppIfTest

Inherited Members

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

Methods

If_Conditional_Test()

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

If_Null_Test()

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

Class Test1.AppIntTest

Namespace: TestProject

Assembly: TestProject1.dll

```
[TestClass]
public class Test1.AppIntTest
```

Inheritance

object ← Test1.AppIntTest

Inherited Members

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

Methods

AppInt_Singleton_Test()

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

Int_Restrictions_Test()

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

Int_Test_Constructor()

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

Class Test1.AppMethodTest

Namespace: TestProject

Assembly: TestProject1.dll

```
[TestClass]
public class Test1.AppMethodTest
```

Inheritance

object ← Test1.AppMethodTest

Inherited Members

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

Methods

Method_Null_Test()

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

Method_Restrictions_Test()

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

Class Test1.AppRealTest

Namespace: TestProject

Assembly: TestProject1.dll

```
[TestClass]
public class Test1.AppRealTest
```

Inheritance

object ← Test1.AppRealTest

Inherited Members

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

Methods

RealRestrictionsTest()

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

Real_Null_Test()

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

Class Test1.AppwhileTest

Namespace: TestProject

Assembly: TestProject1.dll

```
[TestClass]
public class Test1.AppwhileTest
```

Inheritance

object ← Test1.AppwhileTest

Inherited Members

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

Methods

While_Null_Test()

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

While_Restrictions_Test()

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