Namespace ASE_Assignment

Classes

AppArray

Represents a specific implementation of an array command in the system. Inherits from the base Array class and performs additional operations.

AppCall

Represents an extended version of the BOOSE.Call class with relaxed restrictions.

AppCanvas

Represents a drawing canvas where users can draw shapes and text. This class allows drawing basic shapes such as circles, rectangles, and triangles, as well as moving the drawing pen and changing properties like color and size.

AppClear

Represents a command that clears the canvas. Inherits from the BOOSE.CanvasCommand class.

AppCommandFactory

A singleton implementation of the command factory pattern. This class creates specific command objects based on the provided command type.

AppElse

Represents an application-specific implementation of the BOOSE.Else class. This class overrides certain behaviors, such as restrictions.

AppFor

Represents an application-specific implementation of the BOOSE.For class. This class overrides certain behaviors, such as restrictions.

AppIf

Represents a specific implementation of the BOOSE.If class. Provides custom behavior by reducing restrictions upon initialization.

AppInt

Represents a specific implementation of the BOOSE.Int class. Inherits the behavior and properties of the BOOSE.Int class. Overrides the BOOSE.Int.Restrictions() method.

AppMethod

Represents an application method that extends the BOOSE.Method class. This class provides functionality to reset or decrease certain counts and reduce restrictions.

AppReals

Represents a specialized implementation of the BOOSE.Real class with restrictions on value usage.

AppReset

Represents a command that resets the canvas. Inherits from the BOOSE.CanvasCommand class.

AppStoredProgram

Represents a specific implementation of the BOOSE.StoredProgram class. Inherits the behavior and properties of the BOOSE.StoredProgram class.

AppTriangle

Command to create a triangle with specified width and height on a canvas.

AppWhile

Represents an application-specific implementation of the BOOSE.While class. This class provides custom functionality by reducing restrictions during initialization.

AppWrite

Represents a command to write text on the canvas. Inherits from BOOSE.CommandOneParameter and provides functionality to display text messages with length restrictions.

Form1

Main form class that handles canvas operations and command execution.

Class AppArray

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a specific implementation of an array command in the system. Inherits from the base Array class and performs additional operations.

```
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.Compile() , Array.CheckParameters(string[]) , Array.Execute() , Array.ProcessArrayParametersCompile(bool) , Array.ProcessArrayParametersExecute(bool) , Array.SetIntArray(int, int, int) , Array.SetRealArray(double, int, int) , Array.GetIntArray(int, int) , Array.GetRealArray(int, int) , Array.Rows , Array.Columns , Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , Evaluation.ProcessExpression(string) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , Command.ProcessParameters(string) , Command.ToString() , Command.Program , Command.Name , Command.ParameterList , Command.Parameters , Command.Paramsint , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

AppArray()

Initializes a new instance of the AppArray class. This constructor reduces the restriction counter to enforce system rules regarding array operations.

```
public AppArray()
```

Class AppCall

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents an extended version of the BOOSE.Call class with relaxed restrictions.

```
public class AppCall : Call, ICommand
```

Inheritance

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

Implements

ICommand

Inherited Members

Call.methodName , Call.Compile() , Call.Execute() , CompoundCommand.ReduceRestrictions() , CompoundCommand.CheckParameters(string[]) , CompoundCommand.CorrespondingCommand , ConditionalCommand.endLineNumber , ConditionalCommand.EndLineNumber , ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType , ConditionalCommand.ReturnLineNumber , Boolean.BoolValue , Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , Evaluation.ProcessExpression(string) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , Command.ProcessParameters(string) , Command.ToString() , Command.Program , Command.Name , Command.ParameterList , Command.Parameters , Command.Paramsint , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Remarks

The AppCall class overrides the restrictions imposed by the base BOOSE.Call class, allowing more flexibility in its use. It is specifically designed to remove predefined limitations by reducing restriction counters and overriding restriction-related methods.

Constructors

AppCall()

Initializes a new instance of the AppCall class.

```
public AppCall()
```

Remarks

This constructor reduces the restriction counters inherited from the base class by calling the BOOSE. CompoundCommand.ReduceRestrictions() method.

Methods

Restrictions()

Overrides the restrictions method to effectively neutralize any restrictions.

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

Remarks

The overridden BOOSE.Boolean.Restrictions() method in this class does nothing, effectively disabling any restriction logic inherited from the base class.

See Also

Call

Class AppCanvas

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a drawing canvas where users can draw shapes and text. This class allows drawing basic shapes such as circles, rectangles, and triangles, as well as moving the drawing pen and changing properties like color and size.

```
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 the canvas and sets up a red pen. Sets default canvas size and pen properties.

```
public AppCanvas()
```

Properties

PenColour

Gets or sets the pen color. Allows the user to change the drawing pen's color.

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

Property Value

object

Xpos

Gets or sets the X position on the canvas. Controls where the drawing pen is located horizontally.

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

Property Value

int

Ypos

Gets or sets the Y position on the canvas. Controls where the drawing pen is located vertically.

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

Property Value

int

Methods

Circle(int, bool)

Draws a circle with the specified radius and fills it if specified.

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

Parameters

`radius` int

The radius of the circle.

filled bool

Specifies whether the circle should be filled with the current pen color.

Clear()

```
public void Clear()
```

DrawTo(int, int)

Draws a line from the current position to the specified position. Moves the pen to the end point after drawing.

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

Parameters

toX int

The X-coordinate to draw the line to.

toY int

The Y-coordinate to draw the line to.

MoveTo(int, int)

Moves the current position to the specified X and Y.

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

Parameters

x int

y int

Rect(int, int, bool)

Draws a rectangle with the specified width, height, 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

Specifies whether the rectangle should be filled with the current pen color.

Reset()

Resets the position to (0, 0). Moves the pen back to the top-left corner of the canvas.

```
public void Reset()
```

Set(int, int)

Sets the canvas size to the specified width and height. Initializes the graphics object to start drawing on the canvas.

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

Parameters

`xsize` int

The desired width of the canvas.

ysize int↗

The desired height of the canvas.

SetColour(int, int, int)

Sets the pen color using RGB values. This method allows for precise color selection using red, green, and blue 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).

SetPenSize(int)

Sets the pen size. Allows for adjusting the width of the drawing pen.

```
public void SetPenSize(int size)
```

Parameters

size int↗

The size of the pen (1 to 10).

Tri(int, int)

Draws a triangle with the specified 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 the specified text on the canvas.

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

Parameters

`text` string

getBitmap()

Returns the current bitmap of the canvas. This bitmap represents the visual content of the canvas.

```
public object getBitmap()
```

Returns

object

Class AppClear

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a command that clears the canvas. Inherits from the BOOSE.CanvasCommand class.

```
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)

Constructors

AppClear()

Initializes a new instance of the AppClear class. This constructor is parameterless.

```
public AppClear()
```

AppClear(Canvas)

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

```
public AppClear(Canvas c)
```

Parameters

`c` Canvas

The BOOSE.Canvas object to clear.

Methods

CheckParameters(string[])

Validates the parameters for the clear command. Throws an exception if there is more than one parameter.

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

Parameters

`parameter` string[]

An array of strings representing the command parameters.

Exceptions

Exception

Thrown when the number of parameters exceeds one.

Execute()

Executes the clear command, clearing the canvas.

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

Class AppCommandFactory

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

A singleton implementation of the command factory pattern. This class creates specific command objects based on the provided command type.

```
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()

Properties

Instance

Provides access to the singleton instance of the AppCommandFactory. Ensures that only one instance of the factory exists throughout the application's lifecycle.

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

Property Value

AppCommandFactory

Methods

MakeCommand(string)

Creates and returns a command object 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", "int", "real", "array")

Returns

ICommand

A command object that implements the ICommand interface.

Class AppElse

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents an application-specific implementation of the BOOSE.Else class. This class overrides certain behaviors, such as restrictions.

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

Inheritance

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

Implements

ICommand

Inherited Members

Else.CheckParameters(string[]) , Else.Compile() , Else.Execute() , Else.CorrespondingEnd , CompoundCommand.ReduceRestrictions() , CompoundCommand.CorrespondingCommand , ConditionalCommand.endLineNumber , ConditionalCommand.EndLineNumber , ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType , ConditionalCommand.ReturnLineNumber , Boolean.BoolValue , Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , Evaluation.ProcessExpression(string) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , Command.ProcessParameters(string) , Command.ToString() , Command.Program , Command.Name , Command.ParameterList , Command.Parameters , Command.Paramsint , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Methods

Restrictions()

Overrides the Else.Restrictions method to provide custom restriction logic.

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

Class AppFor

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents an application-specific implementation of the BOOSE.For class. This class overrides certain behaviors, such as restrictions.

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

Inheritance

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

Implements

ICommand

Inherited Members

For.Compile() , For.Execute() , For.LoopControlV , For.From , For.To , For.Step , ConditionalCommand.endLineNumber , ConditionalCommand.EndLineNumber , ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType , ConditionalCommand.ReturnLineNumber , Boolean.BoolValue , Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , Evaluation.CheckParameters(string[]) , Evaluation.ProcessExpression(string) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , Command.ProcessParameters(string) , Command.ToString() , Command.Program , Command.Name , Command.ParameterList , Command.Parameters , Command.Paramsint , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Methods

Restrictions()

Overrides the For.Restrictions method to provide custom restriction logic.

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

Class AppIf

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a specific implementation of the BOOSE.If class. Provides custom behavior by reducing restrictions upon initialization.

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

Inheritance

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

Implements

ICommand

Inherited Members

CompoundCommand.ReduceRestrictions() , CompoundCommand.CheckParameters(string[]) , CompoundCommand.Compile() , CompoundCommand.CorrespondingCommand , ConditionalCommand.endLineNumber , ConditionalCommand.Execute() , ConditionalCommand.EndLineNumber , ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType , ConditionalCommand.ReturnLineNumber , Boolean.BoolValue , Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , Evaluation.ProcessExpression(string) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , Command.ProcessParameters(string) , Command.ToString() , Command.Program , Command.Name , Command.ParameterList , Command.Parameters , Command.Paramsint , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

AppIf()

Initializes a new instance of the AppIf class. Calls the If.ReduceRestrictions method to relax the restrictions applied to the instance.

```
public AppIf()
```

Methods

Restrictions()

Overrides the If.Restrictions method to provide custom restriction logic.

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

Class AppInt

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a specific implementation of the BOOSE.Int class. Inherits the behavior and properties of the BOOSE.Int class. Overrides the BOOSE.Int.Restrictions() method.

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

Inheritance

object ← Command ← Evaluation ← Int ← AppInt

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()

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

Class AppMethod

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents an application method that extends the BOOSE.Method class. This class provides functionality to reset or decrease certain counts and reduce restrictions.

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

Inheritance

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

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)

Constructors

AppMethod()

Initializes a new instance of the AppMethod class. This constructor calls the base Method.ReduceRestrictions method and resets both the count and method count to zero.

```
public AppMethod()
```

Methods

ResetOrDecreaseCount(int)

Resets or decreases the count in the BOOSE.Boolean class to the specified new value. The method uses reflection to access and modify the non-public field "뇀" in the BOOSE.Boolean class.

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

Parameters

`newValue` int

The new value to set for the count.

Exceptions

BOOSEException

Thrown when the non-public field "뇀" in the BOOSE.Boolean class cannot be accessed.

ResetOrDecreaseMethodCount(int)

Resets or decreases the method count in the BOOSE.Method class to the specified new value. The method uses reflection to access and modify the non-public field "뇔" in the BOOSE.Method class.

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

Parameters

`newValue` int

The new value to set for the method count.

Exceptions

BOOSEException

Thrown when the non-public field "뇔" in the BOOSE.Method class cannot be accessed.

Class AppReals

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a specialized implementation of the BOOSE.Real class with restrictions on value usage.

```
public class AppReals : Real, ICommand
```

Inheritance

object ← Command ← Evaluation ← Real ← AppReals

Implements

ICommand

Inherited Members

Real.Compile() , Real.Execute() , Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , Evaluation.CheckParameters(string[]) , Evaluation.ProcessExpression(string) , Evaluation.Expression , Evaluation.VarName , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , Command.ProcessParameters(string) , Command.ToString() , Command.Program , Command.Name , Command.ParameterList , Command.Parameters , Command.Paramsint , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

AppReals()

Initializes a new instance of the AppReals class and applies restrictions.

```
public AppReals()
```

Properties

Value

Gets or sets the real value with restrictions applied during assignment.

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

Property Value

double

Methods

Restrictions()

Applies restrictions to the use of the class by limiting the number of instances or value assignments.

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

Exceptions

RestrictionException

Thrown when restrictions are exceeded.

Class AppReset

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a command that resets the canvas. Inherits from the BOOSE.CanvasCommand class.

```
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)

Constructors

AppReset()

Initializes a new instance of the AppReset class. This constructor is parameterless.

```
public AppReset()
```

AppReset(Canvas)

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

```
public AppReset(Canvas c)
```

Parameters

c Canvas

The BOOSE.Canvas object to reset.

Methods

CheckParameters(string[])

Validates the parameters for the reset command. Throws an exception if there is more than one parameter.

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

Parameters

parameter string[]

An array of strings representing the command parameters.

Exceptions

Exception

Thrown when the number of parameters exceeds one.

Execute()

Executes the reset command, resetting the canvas to its initial state.

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

Class AppStoredProgram

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a specific implementation of the BOOSE.StoredProgram class. Inherits the behavior and properties of the BOOSE.StoredProgram class.

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

Inheritance

object ← ArrayList ← StoredProgram ← AppStoredProgram

Implements

IList, ICollection, IEnumerable, ICloneable, IStoredProgram

Inherited Members

StoredProgram.SyntaxOk , StoredProgram.AddMethod(Method) , StoredProgram.GetMethod(string) , StoredProgram.AddVariable(Evaluation) , StoredProgram.GetVariable(string) , StoredProgram.GetVariable(int) , StoredProgram.FindVariable(Evaluation) , StoredProgram.FindVariable(string) , StoredProgram.VariableExists(string) , StoredProgram.GetVarValue(string) , StoredProgram.UpdateVariable(string, int) , StoredProgram.UpdateVariable(string, double) , StoredProgram.UpdateVariable(string, bool) , StoredProgram.DeleteVariable(string) , StoredProgram.IsExpression(string) , StoredProgram.EvaluateExpressionWithString(string) , StoredProgram.EvaluateExpression(string) , StoredProgram.Push(ConditionalCommand) , StoredProgram.Pop() , StoredProgram.Add(Command) , StoredProgram.NextCommand() , StoredProgram.ResetProgram() , StoredProgram.Commandsleft() , StoredProgram.PC , ArrayList.Adapter(IList) , ArrayList.Add(object) , ArrayList.AddRange(ICollection) , ArrayList.BinarySearch(int, int, object, IComparer) , ArrayList.BinarySearch(object) , ArrayList.BinarySearch(object, IComparer) , ArrayList.Clear() , ArrayList.Clone() , ArrayList.Contains(object) , ArrayList.CopyTo(Array) , ArrayList.CopyTo(Array, int) , ArrayList.CopyTo(int, Array, int, int) , ArrayList.FixedSize(ArrayList) , ArrayList.FixedSize(IList) , ArrayList.GetEnumerator() , ArrayList.GetEnumerator(int, int) , ArrayList.GetRange(int, int) , ArrayList.IndexOf(object) , ArrayList.IndexOf(object, int) , ArrayList.IndexOf(object, int, int) , ArrayList.Insert(int, object) , ArrayList.InsertRange(int, ICollection) , ArrayList.LastIndexOf(object) , ArrayList.LastIndexOf(object, int) , ArrayList.LastIndexOf(object, int, int) , ArrayList.ReadOnly(ArrayList) , ArrayList.ReadOnly(IList) , ArrayList.Remove(object) , ArrayList.RemoveAt(int) , ArrayList.RemoveRange(int, int) , ArrayList.Repeat(object, int) , ArrayList.Reverse() , ArrayList.Reverse(int, int) , ArrayList.SetRange(int, ICollection) , ArrayList.Sort() , ArrayList.Sort(IComparer) , ArrayList.Sort(int, int, IComparer) , ArrayList.Synchronized(ArrayList) , ArrayList.Synchronized(IList) , ArrayList.ToArray() , ArrayList.ToArray(Type) , ArrayList.TrimToSize() , ArrayList.Capacity , ArrayList.Count , ArrayList.IsFixedSize , ArrayList.IsReadOnly , ArrayList.IsSynchronized , ArrayList.this[int] , ArrayList.SyncRoot , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

AppStoredProgram(ICanvas)

Initializes a new instance of the AppStoredProgram class.

```
public AppStoredProgram(ICanvas canvas)
```

Parameters

canvas ICanvas

The canvas used to interact with the stored program.

Methods

Run()

Runs the stored program, executing commands in sequence. If too many iterations occur or the program counter (PC) is too low, an error is reported. If there are unpopped items on the stack after running, the syntax is considered incorrect.

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

Class AppTriangle

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Command to create a triangle with specified width and height 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 with default values.

```
public AppTriangle()
```

AppTriangle(Canvas, int, int)

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

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

Parameters

`c` Canvas

Canvas where the triangle will be drawn.

`width` int

Width of the triangle.

`height` int

Height of the triangle.

Methods

CheckParameters(string[])

Checks the parameters for valid width and height before execution.

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

Parameters

`parameterList` string[]

Array of parameters to check.

Execute()

Executes the command to draw the triangle on the canvas.

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

Class AppWhile

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents an application-specific implementation of the BOOSE.While class. This class provides custom functionality by reducing restrictions during initialization.

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

Inheritance

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

Implements

ICommand

Inherited Members

CompoundCommand.ReduceRestrictions() , CompoundCommand.CheckParameters(string[]) , CompoundCommand.Compile() , CompoundCommand.CorrespondingCommand , ConditionalCommand.endLineNumber , ConditionalCommand.Execute() , ConditionalCommand.EndLineNumber , ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType , ConditionalCommand.ReturnLineNumber , Boolean.BoolValue , Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , Evaluation.ProcessExpression(string) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , Command.ProcessParameters(string) , Command.ToString() , Command.Program , Command.Name , Command.ParameterList , Command.Parameters , Command.Paramsint , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

AppWhile()

Initializes a new instance of the AppWhile class. The constructor invokes the base method to reduce restrictions.

```
public AppWhile()
```

Methods

Restrictions()

Overrides the While.Restrictions method to allow custom restriction logic.

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

Class AppWrite

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Represents a command to write text on the canvas. Inherits from BOOSE.CommandOneParameter and provides functionality to display text messages with length restrictions.

```
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 command requires exactly one parameter: the text message to be displayed. The text length is restricted to a maximum of 1000 characters.

Constructors

AppWrite()

Initializes a new instance of the AppWrite class. This constructor is used by the command factory pattern.

```
public AppWrite()
```

AppWrite(Canvas, string)

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

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

Parameters

`c` Canvas

The canvas instance on which to write the text.

`message` string

The text message to be written on the canvas.

Methods

CheckParameters(string[])

Validates the parameters passed to the write command and stores the unprocessed text.

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

Parameters

`parameterList` string[]

An array of string parameters to validate.

Exceptions

CommandException

Thrown when the number of parameters is not exactly 1, as the write command requires exactly one text parameter.

Execute()

Executes the write command with the specified text message. Uses the unprocessed parameter directly as the text to be written.

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

Exceptions

RestrictionException

Thrown when the text length exceeds the maximum allowed length of 1000 characters.

See Also

CommandOneParameter
Canvas

Class Form1

Namespace: ASE_Assignment

Assembly: ASE_Assignment.dll

Main form class that handles canvas operations and command execution.

```
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 components and sets up the canvas, command factory, 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.

Namespace BOOSEConsoleApp

Classes

Program

Represents the entry point for the BOOSE Console Application.

Class Program

Namespace: BOOSEConsoleApp

Assembly: BOOSEConsole.dll

Represents the entry point for the BOOSE Console Application.

```
public class Program
```

Inheritance

object ← Program

Inherited Members

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

Remarks

This application allows users to interact with a canvas through commands. Users can draw shapes such as circles, rectangles, and triangles, move the pen position, and clear the canvas.

Methods

Main(string[])

The main method where the program starts execution.

```
public static void Main(string[] args)
```

Parameters

`args` string[]

Command-line arguments passed to the application.

Remarks

The main method initializes the canvas and sets up the required objects to parse and execute text-based user commands. Users can enter commands to simulate drawing shapes, moving the pen, and interacting with the canvas. The program runs and outputs text messages describing the actions performed until the "exit" command is entered.

Namespace UnitTesting

Classes

Test1

This class contains unit tests for verifying drawing operations on the AppCanvas class. It includes tests for circle, rectangle, and line drawing, movement and position boundary checks, variable replaced, method execution, design patterns, and more.

Test1.ResetTest

Tests the Reset method of the Canvas class to ensure the position is reset to the origin (0, 0).

Class Test1

Namespace: UnitTesting

Assembly: UnitTesting.dll

This class contains unit tests for verifying drawing operations on the AppCanvas class. It includes tests for circle, rectangle, and line drawing, movement and position boundary checks, variable replaced, method execution, design patterns, and more.

```
[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

AppArrayTest()

Tests array operations involving integers and real values. Verifies that the expected values are correctly stored and retrieved from the arrays.

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

AppForTests()

Tests the drawing functionality in the AppFor class by simulating multiple drawing steps with different pens and circle radii.

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

AppIfTest()

Tests conditional operations on AppIf objects. Verifies that the expected values are correctly determined based on the control variable.

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

AppIntTests()

Tests a series of drawing operations involving circles and rectangles on the canvas. Verifies that the bitmap object is created after all operations.

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

AppMethodExecutionTest()

Tests the method execution of AppMethod to verify multiplication and drawing actions are handled correctly.

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

AppRealsTests()

Tests basic operations on AppReals objects, including area and circumference calculations. Verifies that the calculations are correct based on the input values.

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

AppTriangleTest()

Tests the triangle drawing functionality in AppCanvas, ensuring a triangle is drawn at the specified position.

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

AppWriteTest()

Tests the writing functionality of the canvas in AppCanvas, ensuring the bitmap is created after writing text.

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

Multiline_Test()

Tests the drawing of a circle with multiline commands on the canvas. Verifies that the bitmap object is created after drawing the circle.

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

TestColorBoundaryValues()

Tests the behavior of AppCanvas when setting the color with values that are exactly at the valid boundary (e.g., 0 and 255). Verifies that the canvas correctly handles color values at their valid boundary limits.

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

TestDrawAfterClear()

Tests the behavior of AppCanvas when drawing after clearing the canvas. Verifies that the canvas is cleared and drawing can continue after the canvas is reset.

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

TestDrawExcessivelyLargeShape()

Tests the behavior of AppCanvas when drawing with an excessively large shape (e.g., radius exceeds canvas size). Verifies that the canvas correctly handles drawing outside the visible bounds.

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

TestDrawMultipleShapesAtOnce()

Tests the behavior of AppCanvas when attempting to draw multiple shapes at once. Verifies that all shapes are drawn sequentially and the canvas properly handles multiple drawing operations.

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

TestDrawTo()

Tests the drawing of a line using the DrawTo method. Verifies that the bitmap object is created and the pen position is updated.

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

TestDrawZeroDimensions()

Tests the behavior of AppCanvas when attempting to draw a shape with zero dimensions (e.g., radius = 0 for circle). Verifies that no shape is drawn and the canvas remains empty.

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

TestMoveTo_Boundary()

Tests the MoveTo method to move the pen to the canvas boundary. Verifies that the X and Y positions are correctly updated at the boundary.

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

TestMoveTo_ValidPosition()

Tests the MoveTo method to move the pen to a specific position. Verifies that the X and Y positions are correctly updated.

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

TestMultipleDrawings()

Tests the behavior of AppCanvas when drawing multiple shapes sequentially. Verifies that the canvas correctly displays each shape drawn on it.

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

TestRectangleOutline()

Tests the drawing of an outlined rectangle on the canvas. Verifies that the bitmap object is created after drawing the rectangle.

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

TestSingletonInstance_CreatesOnlyOneInstance()

Tests that the singleton pattern is correctly implemented in the AppCommandFactory class.

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

Class Test1.ResetTest

Namespace: UnitTesting

Assembly: UnitTesting.dll

Tests the Reset method of the Canvas class to ensure the position is reset to the origin (0, 0).

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

Inheritance

object ← Test1.ResetTest

Inherited Members

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

Methods

Reset_ResetsPositionToOrigin()

Verifies that calling the Reset method resets the position to the origin (0, 0).

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

SetUp()

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