Namespace BooseApp

Classes

AppArray

A new Array Class that inherits BOOSE Array to remove restrictions placed.

AppCall

A new Call Class that inherits BOOSE Call to remove restrictions placed.

AppCanvas

Represents a custom canvas for graphical operations, implementing the ICanvas interface from the BOOSE library.

AppCircle

Represents a command to draw a circle on a canvas with customizable radius and fill options.

AppCommandFactory

A factory class responsible for creating command objects for the BooseApp application.

AppElse

A new Else Class that inherits BOOSE Else to remove restrictions placed.

AppEnd

A new End Class that inherits BOOSE End to remove restrictions placed.

AppExceptionHandler

Handles exceptions by categorizing them into syntax, runtime, and unexpected errors. Provides methods for recording, retrieving, and clearing error messages.

AppFor

A new For Class that inherits BOOSE For to remove restrictions placed.

AppGame

Represents the main form of the BooseApp application. Handles user interaction, command parsing, and error handling.

AppIf

A new If Class that inherits BOOSE If to remove restrictions placed.

AppInt

A new Int Class that inherits BOOSE Int to remove restrictions placed.

AppMethod

A new Method Class that inherits BOOSE Method to remove restrictions placed.

AppParser

A custom implementation of the BOOSE BOOSE.Parser class. The AppParser is responsible for parsing a program string, processing commands, and updating the associated BOOSE.StoredProgram.

AppReal

A new Real Class that inherits BOOSE Real to remove restrictions placed.

AppRect

Represents a command to draw a rectangle on a canvas. The rectangle can be filled or unfilled, based on the parameters provided.

AppStoredProgram

Represents a specialized version of the BOOSE.StoredProgram class, designed to execute a sequence of commands on a canvas while monitoring for potential issues like infinite loops.

AppTriangle

Represents a command to draw a triangle on a canvas. Inherits functionality from BOOSE.Command TwoParameters.

AppWhile

A new Write Class that inherits BOOSE Write to remove restrictions placed.

AppWrite

Represents a command to write text on the canvas in the BooseApp application.

Form1

Represents the main form of the BooseApp application. Handles user interaction, command parsing, and error handling.

Class AppArray

Namespace: BooseApp

Assembly: BooseApp.dll

A new Array Class that inherits BOOSE Array to remove restrictions placed.

```
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. Resets the restriction counter to enable the creation of arrays without limitations.

```
public AppArray()
```

Class AppCall

Namespace: BooseApp

Assembly: BooseApp.dll

A new Call Class that inherits BOOSE Call to remove restrictions placed.

```
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)

Constructors

AppCall()

Initializes a new instance of the AppCall class.

```
public AppCall()
```

Methods

Restrictions()

Overrides the Restrictions function of Boolean Class to enable the use of call without variable limits.

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

Class AppCanvas

Namespace: BooseApp

Assembly: BooseApp.dll

Represents a custom canvas for graphical operations, implementing the ICanvas interface from the BOOSE library.

```
public class AppCanvas : ICanvas
```

Inheritance

object ← AppCanvas

Implements

ICanvas

Inherited Members

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

Constructors

AppCanvas()

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

```
public AppCanvas()
```

Properties

PenColour

Gets or sets the current pen color.

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

Property Value

object

Xpos

Gets the single instance of the AppCanvas class. Ensures thread safety by using a lock mechanism.

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

Property Value

int

Ypos

Gets or sets the current Y-coordinate on the canvas.

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

Property Value

int

Methods

Circle(int, bool)

Draws a circle on the canvas at the current position with the specified radius.

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

Parameters

`radius` int

Radius of the circle. Must be non-negative.

`filled` bool

Indicates whether the circle should be filled.

Exceptions

CanvasException

Thrown if the radius is invalid.

Clear()

Clears the canvas by resetting it to a gray background.

```
public void Clear()
```

DrawTo(int, int)

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

```
public void DrawTo(int tox, int toy)
```

Parameters

`tox` int

Target X-coordinate.

`toy` int

Target Y-coordinate.

Exceptions

CanvasException

Thrown if the coordinates are outside the canvas bounds.

MoveTo(int, int)

Moves the pen to the specified coordinates without drawing.

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

Parameters

`x` int

New X-coordinate.

`y` int

New Y-coordinate.

Exceptions

CanvasException

Thrown if the coordinates are outside the canvas bounds.

Rect(int, int, bool)

Draws a rectangle centered at the current position with the specified dimensions.

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

Parameters

`width` int

Width of the rectangle.

`height` int

Height of the rectangle.

`filled` bool

Indicates whether the rectangle should be filled.

Exceptions

CanvasException

Thrown if the dimensions are invalid.

Reset()

Resets the pen position to (0, 0).

```
public void Reset()
```

Set(int, int)

Initializes the canvas with the default variable values and positions.

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

Parameters

`xsize` int

Width of the canvas.

`ysize` int

Height of the canvas.

SetColour(int, int, int)

Sets the pen color using RGB values.

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

Parameters

`red` int

Red component

`green` int

Green component

`blue` int

Blue component

Exceptions

CanvasException

Thrown if any component exceeds the valid range.

Tri(int, int)

Placeholder for drawing a triangle. Not yet implemented.

Width of the triangle.Height of the triangle.Thrown if Width and Height are Invalid.

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

Parameters

`width` int

`height` int

WriteText(string)

Displays text on the canvas at the current position.

The text or variable to be displayed.Thrown if no parameters are supplied to write.

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

Parameters

`text` string

getBitmap()

Saves the current canvas bitmap as an image file and returns the bitmap object.

```
public object getBitmap()
```

Returns

object

The current bitmap of the canvas.

Class AppCircle

Namespace: BooseApp

Assembly: BooseApp.dll

Represents a command to draw a circle on a canvas with customizable radius and fill options.

```
public class AppCircle : CommandTwoParameters, ICommand
```

Inheritance

object ← Command ← CanvasCommand ← CommandOneParameter ← CommandTwoParameters ← AppCircle

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

AppCircle()

Initializes a new instance of the AppCircle class with default values.

```
public AppCircle()
```

AppCircle(Canvas, int, bool)

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

```
public AppCircle(Canvas c, int radius, bool fill)
```

Parameters

`c` Canvas

The canvas on which the circle will be drawn.

`radius` int

The radius of the circle.

`fill` bool

A boolean indicating whether the circle should be filled.

Methods

CheckParameters(string[])

Checks the validity of the parameters provided for the command. Ensures the number of parameters is between 1 and 2.

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

Parameters

`parameterList` string[]

An array of strings representing the parameters.

Exceptions

CommandException

Thrown when the number of parameters is invalid.

Execute()

Validates the radius and fill parameters before drawing. Then Executes the Command to draw the Circle.

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

Exceptions

RestrictionException

Thrown when the radius exceeds the allowed boundary of 9999.

Class AppCommandFactory

Namespace: BooseApp

Assembly: BooseApp.dll

A factory class responsible for creating command objects for the BooseApp application.

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

Inheritance

object ← CommandFactory ← AppCommandFactory

Implements

ICommandFactory

Inherited Members

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

Constructors

AppCommandFactory()

Initializes a new instance of the AppCommandFactory class.

```
public AppCommandFactory()
```

Methods

MakeCommand(string)

Creates and returns a command object based on the provided command type.

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

Parameters

`commandType` string

The type of the command to create.

Returns

ICommand

An instance of BOOSE.ICommand corresponding to the specified command type.

Class AppElse

Namespace: BooseApp

Assembly: BooseApp.dll

A new Else Class that inherits BOOSE Else to remove restrictions placed.

```
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)

Constructors

AppElse()

Initializes a new instance of the AppElse class.

```
public AppElse()
```

Methods

Restrictions()

Overrides the Restrictions function of Boolean Class to enable the use of call without variable limits.

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

Class AppEnd

Namespace: BooseApp

Assembly: BooseApp.dll

A new End Class that inherits BOOSE End to remove restrictions placed.

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

Inheritance

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

Implements

ICommand

Inherited Members

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

Constructors

AppEnd()

Initializes a new instance of the AppEnd class.

```
public AppEnd()
```

Methods

Restrictions()

Overrides the Restrictions function of Boolean Class to enable the use of call without variable limits.

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

Class AppExceptionHandler

Namespace: BooseApp

Assembly: BooseApp.dll

Handles exceptions by categorizing them into syntax, runtime, and unexpected errors. Provides methods for recording, retrieving, and clearing error messages.

```
public class AppExceptionHandler
```

Inheritance

object ← AppExceptionHandler

Inherited Members

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

Constructors

AppExceptionHandler()

Initializes a new instance of the AppExceptionHandler class. Sets up containers for syntax and runtime error messages.

```
public AppExceptionHandler()
```

Properties

HandlerInstance

```
public static AppExceptionHandler HandlerInstance { get; }
```

Property Value

AppExceptionHandler

Methods

AddRuntimeError(string)

Records a runtime error message.

```
public void AddRuntimeError(string message)
```

Parameters

`message` string

The error message to record.

AddSyntaxError(string)

Records a syntax error message.

```
public void AddSyntaxError(string message)
```

Parameters

`message` string

The error message to record.

AddUnexpectedError(string)

Records an unexpected error message.

```
public void AddUnexpectedError(string message)
```

Parameters

`message` string

The error message to record.

ClearErrors()

Clears all recorded error messages.

```
public void ClearErrors()
```

GetErrorMessages()

Retrieves all recorded error messages as a single string.

```
public string GetErrorMessages()
```

Returns

string

A concatenated string of syntax and runtime error messages.

Class AppFor

Namespace: BooseApp

Assembly: BooseApp.dll

A new For Class that inherits BOOSE For to remove restrictions placed.

```
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)

Constructors

AppFor()

Initializes a new instance of the AppFor class.

```
public AppFor()
```

Methods

Restrictions()

Overrides the Restrictions function of Boolean Class to enable the use of call without variable limits.

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

Class AppGame

Namespace: BooseApp

Assembly: BooseApp.dll

Represents the main form of the BooseApp application. Handles user interaction, command parsing, and error handling.

```
public class AppGame
```

Inheritance

object ← AppGame

Inherited Members

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

Constructors

AppGame()

Initializes a new instance of the AppGame class. Sets up the initial state of the game including player position, targets, and bullets.

```
public AppGame()
```

Properties

IsActive

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

Property Value

bool

Methods

Draw(Graphics)

Draws the current game state including the targets, player (triangle), and bullets.

```
public void Draw(Graphics g)
```

Parameters

g Graphics

The Graphics object used for drawing.

Fire()

Fires a bullet from the player's current position.

```
public void Fire()
```

GetScore()

Gets the current score of the game.

```
public int GetScore()
```

Returns

int

The current score.

MovePlayer(int)

Moves the player horizontally by the specified amount.

```
public void MovePlayer(int dx)
```

Parameters

dx int

The amount to move the player on the X-axis. Positive values move the player to the right, and negative values move the player to the left.

ResetGame()

Resets the game to its initial state, including resetting the score, clearing targets and bullets, and placing the player at the starting position.

```
public void ResetGame()
```

Update()

Updates the game state by moving bullets, checking for collisions with targets, and updating the score. Stops the game if all targets are hit.

```
public void Update()
```

Class AppIf

Namespace: BooseApp

Assembly: BooseApp.dll

A new If Class that inherits BOOSE If to remove restrictions placed.

```
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. Resets the restriction counter to enable the creation of arrays without limitations.

```
public AppIf()
```

Methods

Restrictions()

Overrides the Restrictions function of Boolean Class to enable the use of call without variable limits.

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

Class AppInt

Namespace: BooseApp

Assembly: BooseApp.dll

A new Int Class that inherits BOOSE Int to remove restrictions placed.

```
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)

Constructors

AppInt()

Initializes a new instance of the AppInt class.

```
public AppInt()
```

Methods

Restrictions()

Overrides the Restrictions function of Boolean Class to enable the use of call without variable limits.

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

Class AppMethod

Namespace: BooseApp

Assembly: BooseApp.dll

A new Method Class that inherits BOOSE Method to remove restrictions placed.

```
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.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. Resets the restriction counter to enable the creation of arrays without limitations.

```
public AppMethod()
```

Methods

Restrictions()

Overrides the Restrictions function of Boolean Class to enable the use of call without variable limits.

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

Class AppParser

Namespace: BooseApp

Assembly: BooseApp.dll

A custom implementation of the BOOSE BOOSE.Parser class. The AppParser is responsible for parsing a program string, processing commands, and updating the associated BOOSE.StoredProgram.

```
public class AppParser : Parser, IParser
```

Inheritance

object ← Parser ← AppParser

Implements

IParser

Inherited Members

Parser.ParseCommand(string) , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

AppParser(CommandFactory, StoredProgram)

Initializes a new instance of the AppParser class.

```
public AppParser(CommandFactory Factory, StoredProgram Program)
```

Parameters

`Factory` CommandFactory

The factory used to create commands.

`Program` StoredProgram

The BOOSE.StoredProgram instance for storing parsed program data.

Methods

ParseProgram(string)

Parses the given program string into individual commands or methods. Adds the parsed commands to the associated BOOSE.StoredProgram.

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

Parameters

`program` string

The program string to be parsed.

Exceptions

ParserException

Thrown when syntax errors are found in the program, with details of the errors.

Class AppReal

Namespace: BooseApp

Assembly: BooseApp.dll

A new Real Class that inherits BOOSE Real to remove restrictions placed.

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

Inheritance

object ← Command ← Evaluation ← Real ← AppReal

Implements

ICommand

Inherited Members

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

Constructors

AppReal()

Initializes a new instance of the AppReal class.

```
public AppReal()
```

Methods

Restrictions()

Overrides the Restrictions function of Boolean Class to enable the use of call without variable limits.

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

Class AppRect

Namespace: BooseApp

Assembly: BooseApp.dll

Represents a command to draw a rectangle on a canvas. The rectangle can be filled or unfilled, based on the parameters provided.

```
public class AppRect : CommandThreeParameters, ICommand
```

Inheritance

object ← Command ← CanvasCommand ← CommandOneParameter ← CommandTwoParameters ← CommandThreeParameters ← AppRect

Implements

ICommand

Inherited Members

CommandThreeParameters.param3 , CommandThreeParameters.param3unprocessed , 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

AppRect()

Initializes a new instance of the AppRect class with default values.

```
public AppRect()
```

AppRect(Canvas, int, int, bool)

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

```
public AppRect(Canvas c, int width, int height, bool fill)
```

Parameters

`c` Canvas

The canvas where the rectangle will be drawn.

`width` int

The width of the rectangle.

`height` int

The height of the rectangle.

`fill` bool

Specifies whether the rectangle should be filled.

Methods

CheckParameters(string[])

Validates the parameters for the rectangle command. Ensures the correct number of parameters is provided.

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

Parameters

`parameterList` string[]

The list of parameters provided for the command.

Exceptions

CommandException

Thrown if the number of parameters is less than 2 or greater than 3.

Execute()

Executes the rectangle drawing command. Reads parameters, validates them, and draws the rectangle on the canvas.

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

Exceptions

CommandException

Thrown if the number of parameters is invalid or if a parameter is out of expected bounds.

Class AppStoredProgram

Namespace: BooseApp

Assembly: BooseApp.dll

Represents a specialized version of the BOOSE.StoredProgram class, designed to execute a sequence of commands on a canvas while monitoring for potential issues like infinite loops.

```
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 with the specified canvas.

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

Parameters

`canvas` ICanvas

The canvas on which the program will operate.

Methods

Run()

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

Exceptions

StoredProgramException

Thrown when a BOOSE.BOOSEException occurs during command execution or when an infinite loop is detected.

Class AppTriangle

Namespace: BooseApp

Assembly: BooseApp.dll

Represents a command to draw a triangle on a canvas. Inherits functionality from BOOSE.CommandTwo Parameters.

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

Inheritance

object ← Command ← CanvasCommand ← CommandOneParameter ← CommandTwoParameters ← AppTriangle

Implements

ICommand

Inherited Members

CommandTwoParameters.param2 , CommandTwoParameters.param2unprocessed , CommandOneParameter.param1 , CommandOneParameter.param1unprocessed , CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas , Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , Command.Compile() , Command.ProcessParameters(string) , Command.ToString() , Command.Program , Command.Name , Command.ParameterList , Command.Parameters , Command.Paramsint , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

AppTriangle()

Initializes a new instance of the AppTriangle class.

```
public AppTriangle()
```

AppTriangle(Canvas, int, int)

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

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

Parameters

`c` Canvas

The canvas on which the triangle will be drawn.

`width` int

The width of the triangle.

`height` int

The height of the triangle.

Methods

CheckParameters(string[])

Validates the parameters required for the triangle command.

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

Parameters

`parameterList` string[]

An array of string parameters to be validated.

Exceptions

CommandException

Thrown if the parameter validation fails in the base class.

Execute()

Executes the command to draw a triangle on the canvas.

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

Remarks

The method retrieves the width and height from the command parameters and uses the Tri(int, int) method to draw the triangle.

Exceptions

CommandException

Thrown if the parameters are invalid or out of range.

Class AppWhile

Namespace: BooseApp

Assembly: BooseApp.dll

A new Write Class that inherits BOOSE Write to remove restrictions placed.

```
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. Resets the restriction counter to enable the creation of arrays without limitations.

```
public AppWhile()
```

Methods

Restrictions()

Overrides the Restrictions function of Boolean Class to enable the use of call without variable limits.

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

Class AppWrite

Namespace: BooseApp

Assembly: BooseApp.dll

Represents a command to write text on the canvas in the BooseApp application.

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

Inheritance

object ← Command ← CanvasCommand ← CommandOneParameter ← AppWrite

Implements

ICommand

Inherited Members

CommandOneParameter.param1 , CommandOneParameter.param1unprocessed , CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas , Command.program , Command.parameterList , Command.parameters , Command.paramsint , Command.Set(StoredProgram, string) , Command.Compile() , Command.ProcessParameters(string) , Command.ToString() , Command.Program , Command.Name , Command.ParameterList , Command.Parameters , Command.Paramsint , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

AppWrite()

Initializes a new instance of the AppWrite class.

```
public AppWrite()
```

Methods

CheckParameters(string[])

Validates the parameters provided for the command.

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

Parameters

`parameter` string[]

Array containing command parameters.

Execute()

Checks if the string passed is an variable or a integer, if yes it evaluates it with string and call the WriteText method of canvas. If the string contains variables with string concatenated or just string, it evaluates it with string and call the WriteText method of canvas.

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

Class Form1

Namespace: BooseApp

Assembly: BooseApp.dll

Represents the main form of the BooseApp application. Handles user interaction, command parsing, and error handling.

```
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.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 application components and initializes necessary objects.

```
public Form1()
```

Methods

CommandLoadStrip_Click(object, EventArgs)

Loads a program from a file selected by the user and displays it in the command box.

```
public void CommandLoadStrip_Click(object sender, EventArgs e)
```

Parameters

`sender` object

The source of the event.

`e` EventArgs

The event arguments.

CommandSaveStrip_Click(object, EventArgs)

Saves the program text to a file selected by the user.

```
public void CommandSaveStrip_Click(object sender, EventArgs e)
```

Parameters

`sender` object

The source of the event.

`e` EventArgs

The event arguments.

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.

GameLoop(object, EventArgs)

Game loop that updates the game state and refreshes the output box. Stops the timer if the game is not active.

```
public void GameLoop(object sender, EventArgs e)
```

Parameters

`sender` object

The source of the event.

`e` EventArgs

The event arguments.

ImageLoadStrip_Click(object, EventArgs)

Loads an image file into the output box and sets the load state.

```
public void ImageLoadStrip_Click(object sender, EventArgs e)
```

Parameters

`sender` object

The source of the event.

`e` EventArgs

The event arguments.

ImageSaveStrip_Click(object, EventArgs)

Saves the current canvas image to a file selected by the user.

```
public void ImageSaveStrip_Click(object sender, EventArgs e)
```

Parameters

`sender` object

The source of the event.

`e` EventArgs

The event arguments.

ProcessCmdKey(ref Message, Keys)

Processes keyboard commands to move the player or fire bullets during the game.

```
protected override bool ProcessCmdKey(ref Message msg, Keys keyData)
```

Parameters

`msg` Message

The message to process.

`keyData` Keys

The key data from the keyboard.

Returns

bool

True if the key is handled, false otherwise.

clearBtn_Click(object, EventArgs)

Handles the clear button click event. Clears the canvas, program text, and game state.

```
public void clearBtn_Click(object sender, EventArgs e)
```

Parameters

`sender` object

The source of the event.

`e` EventArgs

The event arguments.

gameButton_Click(object, EventArgs)

Starts the game by resetting the game state and starting the game timer.

```
public void gameButton_Click(object sender, EventArgs e)
```

Parameters

`sender` object

The source of the event.

`e` EventArgs

The event arguments.

outputBox_Paint(object, PaintEventArgs)

Handles the paint event for the output box. Draws the current game state or canvas image.

```
public void outputBox_Paint(object sender, PaintEventArgs e)
```

Parameters

`sender` object

The source of the event.

`e` PaintEventArgs

The event arguments containing drawing information.

runBtn_Click(object, EventArgs)

Handles the run button click event. Parses the program and runs it. Catches any exceptions and displays errors in the error box.

```
public void runBtn_Click(object sender, EventArgs e)
```

Parameters

`sender` object

The source of the event.

`e` EventArgs

The event arguments.

Namespace BooseAppTest

Classes

AppCanvasTests

Unit tests for the AppCanvas class and its methods.

ConditionalTests

Unit tests for conditional statements in the BOOSE language.

LoopTests

Unit tests for loop constructs in the BOOSE language.

MethodTests

Unit tests for the method functionality in the BOOSE language.

VariableTests

Unit tests for variable declarations and operations (int, real, and array) in the BOOSE language.

Class AppCanvasTests

Namespace: BooseAppTest

Assembly: BooseAppTest.dll

Unit tests for the AppCanvas class and its methods.

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

Inheritance

object ← AppCanvasTests

Inherited Members

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

Methods

Circle_InvalidParameters_ThrowsCanvasException()

Ensures an exception is thrown for invalid circle parameters.

```
[TestMethod]
[ExpectedException(typeof(CanvasException), "Expected Canvas Exception Not Thrown.")]
public void Circle_InvalidParameters_ThrowsCanvasException()
```

Circle_ValidParameters_GraphicsDrawn()

Verifies that a circle is drawn successfully with valid parameters.

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

DrawTo_InvalidParameters_ThrowsCanvasException()

Ensures an exception is thrown for invalid draw-to coordinates.

```
[TestMethod]
[ExpectedException(typeof(CanvasException), "Expected Canvas Exception not thrown.")]
public void DrawTo_InvalidParameters_ThrowsCanvasException()
```

DrawTo_ValidParameters_ChangedCoordinates()

Verifies that the canvas coordinates change correctly after drawing with valid parameters.

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

MoveTo_InvalidParameters_ThrowsCanvasException()

Ensures an exception is thrown for invalid move-to coordinates.

```
[TestMethod]
[ExpectedException(typeof(CanvasException), "Expected Canvas Exception was not thrown")]
public void MoveTo_InvalidParameters_ThrowsCanvasException()
```

MoveTo_ValidParameters_ChangedCoordinates()

Verifies that the canvas coordinates change correctly with valid parameters.

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

MultiLineCommandExecution_InvalidCommands_ThrowsParserException()

Ensures an exception is thrown for invalid multiline commands.

```
[TestMethod]
[ExpectedException(typeof(ParserException), "Excepted Parser Exception for
```

```
Invalid Commands.")]
public void MultiLineCommandExecution_InvalidCommands_ThrowsParserException()
```

MultiLineCommandExecution_NoCommands_BitMapCreationWithNoException()

Verifies bitmap creation after executing an empty command set.

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

MultiLineCommandExecution_OutOfBoundParameters_ThrowsStoredProgramException()

Ensures an exception is thrown for out-of-bound parameters in commands.

```
[TestMethod]
[ExpectedException(typeof(StoredProgramException), "Excepted StoredProgram Exception for Out of Bound Parameteres.")]
public void MultiLineCommandExecution_OutOfBoundParameters_ThrowsStoredProgramException()
```

MultiLineCommandExecution_ValidCommands_BitMapCreationWithNoException()

Verifies bitmap creation after executing valid multiline commands.

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

Rect_InvalidParameters_ThrowsCanvasException()

Ensures an exception is thrown for invalid rectangle parameters.

```
[TestMethod]
[ExpectedException(typeof(CanvasException), "Expected Canvas Exception Not Thrown.")]
```

```
public void Rect_InvalidParameters_ThrowsCanvasException()
```

Rect_ValidParameters_GraphicsDrawn()

Verifies that a rectangle is drawn successfully with valid parameters.

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

SetColor_InvalidColor_ThrowsCanvasException()

Ensures an exception is thrown for invalid color parameters.

```
[TestMethod]
[ExpectedException(typeof(CanvasException), "Expected Canvas Exception was not thrown")]
public void SetColor_InvalidColor_ThrowsCanvasException()
```

SetColor_ValidColor_ChangePenColor()

Verifies that the pen color changes correctly with valid color parameters.

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

Setup()

Initializes the test environment with required objects before each test.

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

Class ConditionalTests

Namespace: BooseAppTest

Assembly: BooseAppTest.dll

Unit tests for conditional statements in the BOOSE language.

```
[TestClass]
public class ConditionalTests
```

Inheritance

object ← ConditionalTests

Inherited Members

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

Methods

If_FalseCondition_RunsTheCodeWithinElseStatement()

Tests that code within an "else" statement runs correctly when the "if" condition is false.

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

If_TrueCondition_RunsTheCodeWithinIfStatement()

Tests that code within an "if" statement runs correctly when the condition is true.

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

Setup()

Initializes the test environment with required objects before each test.

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

Class LoopTests

Namespace: BooseAppTest

Assembly: BooseAppTest.dll

Unit tests for loop constructs in the BOOSE language.

```
[TestClass]
public class LoopTests
```

Inheritance

object ← LoopTests

Inherited Members

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

Methods

For_ValidStatement_RunsTheLoopTillThePredefinedNumberExlusively()

Tests a "for" loop with valid parameters and ensures it iterates the correct number of times.

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

For_VeryHighLoopStatement_ThrowsStoredProgramExceptionForInfiniteLoop()

Tests a "for" loop with a very high iteration count to ensure it detects an infinite loop and throws an exception.

```
[TestMethod]
[ExpectedException(typeof(StoredProgramException), "Expected Exception:
```

```
StoredProgramException was not thrown.")]
public void For_VeryHighLoopStatement_ThrowsStoredProgramExceptionForInfiniteLoop()
```

Setup()

Initializes the test environment with required objects before each test.

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

While_AlwaysTrueStatement_ThrowsStoredProgramExceptionForInfiniteLoop()

Tests a "while" loop with an always-true condition to ensure it detects an infinite loop and throws an exception.

```
[TestMethod]
[ExpectedException(typeof(StoredProgramException), "Expected Exception:
StoredProgramException was not thrown.")]
public void While_AlwaysTrueStatement_ThrowsStoredProgramExceptionForInfiniteLoop()
```

While_ValidStatement_RunsTheLoopTillThePredefinedConditionIsMet()

Tests a "while" loop with valid parameters and ensures it terminates when the condition is met.

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

Class MethodTests

Namespace: BooseAppTest

Assembly: BooseAppTest.dll

Unit tests for the method functionality in the BOOSE language.

```
[TestClass]
public class MethodTests
```

Inheritance

object ← MethodTests

Inherited Members

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

Methods

Method_InvalidParameter_Exception()

Tests an invalid method parameter and ensures a ParserException is thrown.

```
[TestMethod]
[ExpectedException(typeof(ParserException), "Expected Exception: Parser Exception
not thrown.")]
public void Method_InvalidParameter_Exception()
```

Method_InvalidSyntax_Exception()

Tests an invalid method syntax and ensures a ParserException is thrown.

```
[TestMethod]
[ExpectedException(typeof(ParserException), "Expected Exception: Parser Exception
not thrown.")]
public void Method_InvalidSyntax_Exception()
```

Method_ValidSyntax_MethodRunsAndTheResultIsReturned()

Tests a method with valid syntax to ensure it runs and returns the expected result.

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

Setup()

Initializes the test environment with required objects before each test.

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

Class VariableTests

Namespace: BooseAppTest

Assembly: BooseAppTest.dll

Unit tests for variable declarations and operations (int, real, and array) in the BOOSE language.

```
[TestClass]
public class VariableTests
```

Inheritance

object ← VariableTests

Inherited Members

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

Methods

Array_EmptyArryCorrectInitialization_EmptyArrayWithCorrectSizeAndType()

Tests creating an empty array with the correct size and type.

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

Array_InvalidValue_ThrowsStoredProgramException()

Tests setting an invalid value for an array element and ensures it throws a StoredProgramException.

```
[TestMethod]
[ExpectedException(typeof(StoredProgramException), "Expected Exception:
StoredProgramException was not thrown.")]
public void Array_InvalidValue_ThrowsStoredProgramException()
```

Array_ValidValue_ArrayValueSetCorrectly()

Tests setting and retrieving values from an array and checks if the values are set correctly.

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

Int_EmptyValue_ZeroValueStoredAsDefault()

Tests setting an empty int variable and ensures it stores the default value of 0.

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

Int_InvalidValue_ThrowsStoredProgramException()

Tests setting an invalid value for an int variable and ensures it throws a StoredProgramException.

```
[TestMethod]
[ExpectedException(typeof(StoredProgramException), "Expected Exception:
StoredProgramException was not thrown.")]
public void Int_InvalidValue_ThrowsStoredProgramException()
```

Int_ValidValue_VariableValueSetCorrectly()

Tests setting a valid int value for a variable and checks if the value is set correctly.

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

Real_EmptyValue_ZeroValueStoredAsDefault()

Tests setting an empty real variable and ensures it stores the default value of 0.0.

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

Real_InvalidValue_ThrowsStoredProgramException()

Tests setting an invalid value for a real variable and ensures it throws a StoredProgramException.

```
[TestMethod]
[ExpectedException(typeof(StoredProgramException), "Expected Exception:
StoredProgramException was not thrown.")]
public void Real_InvalidValue_ThrowsStoredProgramException()
```

Real_ValidValue_VariableValueSetCorrectly()

Tests setting a valid real value for a variable and checks if the value is set correctly.

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

Setup()

Initializes the test environment with required objects before each test.

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