

Namespace ASE_Assignment

Classes

[AppCanvas](#)

Defines a canvas for the application that implements the ICanvas interface. Includes functionality to draw shapes, text, and manipulate the state of the canvas.

[AppCircle](#)

Represents a command to draw a circle on the canvas. Inherits from BOOSE.CommandTwoParameters and handles drawing circles on the canvas.

[AppCommandFactory](#)

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

[AppRect](#)

Represents the rectangle command in the application. Inherits from BOOSE.CommandThreeParameters and handles drawing rectangles on the canvas.

[AppTri](#)

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

[AppWrite](#)

Represents a command that writes text to a canvas. Inherits from BOOSE.CommandOneParameter and requires one parameter, which is the text to be written.

[Form1](#)

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

Class AppCanvas

Namespace: [ASE Assignment](#)

Assembly: ASE_Assignment.dll

Defines a canvas for the application that implements the ICanvas interface. Includes functionality to draw shapes, text, and manipulate the state of the canvas.

```
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. Sets up the canvas, pen, brush, and initial drawing position.

```
public AppCanvas()
```

Remarks

This constructor creates a new Bitmap to serve as the canvas, initializes a Graphics object to allow drawing on the Bitmap, and sets the default pen color to black with a specified size. It also creates a SolidBrush for filling shapes and sets the canvas size and initial drawing position to (0, 0).

Properties

PenColour

Gets or sets the pen color.

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

Property Value

[object](#)[↗]

Xpos

Gets or sets the X position of the drawing point.

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

Property Value

[int](#)[↗]

Ypos

Gets or sets the Y position of the drawing point.

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

Property Value

[int](#)[↗]

Methods

Circle(int, bool)

Draws a circle at the current position with the given radius.

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

Parameters

radius [int](#)

The radius of the circle.

filled [bool](#)

Indicates whether the circle should be filled.

Exceptions

CanvasException

Thrown when the radius is less than or equal to 0.

Clear()

Clears the canvas by filling it with a gray color.

```
public void Clear()
```

DrawTo(int, int)

Draws a line from the current drawing position to the specified destination coordinates. The line is drawn using the current pen, and the drawing position is updated to the destination point.

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

Parameters

toX [int](#)

The X coordinate of the destination point, where the line should end.

toY [int](#)

The Y coordinate of the destination point, where the line should end.

Exceptions

CanvasException

Thrown when the destination coordinates are outside the canvas bounds. For example, if either the X or Y coordinate exceeds the canvas size or is negative, this method will throw a CanvasException to indicate that the destination is invalid.

MoveTo(int, int)

Moves the drawing position to the specified coordinates on the canvas. This method updates the current drawing position to the new coordinates, and throws an exception if the coordinates are outside the valid canvas bounds.

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

Parameters

x [int](#)

The new X coordinate for the drawing position. It must be between 0 and the canvas width (inclusive).

y [int](#)

The new Y coordinate for the drawing position. It must be between 0 and the canvas height (inclusive).

Exceptions

CanvasException

Thrown when the provided coordinates are outside the valid bounds of the canvas. The valid range for X is [0, XCanvasSize], and for Y is [0, YCanvasSize].

Rect(int, int, bool)

Draws a rectangle at the current position with the specified width and height.

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

Parameters

width [int](#)

The width of the rectangle.

height [int](#)

The height of the rectangle.

filled [bool](#)

Indicates whether the rectangle should be filled.

Exceptions

CanvasException

Thrown when the width or height is less than or equal to 0.

Reset()

Resets the drawing position and the pen color to default values.

```
public void Reset()
```

Set(int, int)

Sets the size of the canvas and creates a new bitmap with the specified size.

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

Parameters

xsize [int](#)

The width of the canvas.

ysize [int](#)

The height of the canvas.

SetColour(int, int, int)

Sets the pen color using RGB values. This method adjusts the pen and brush color based on the provided red, green, and blue components, and updates the drawing color for subsequent drawing operations.

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

Parameters

red [int](#)

The red component of the color, which should be an integer in the range [0, 255].

green [int](#)

The green component of the color, which should be an integer in the range [0, 255].

blue [int](#)

The blue component of the color, which should be an integer in the range [0, 255].

Exceptions

CanvasException

Thrown when any of the RGB values are outside the valid range of [0, 255]. For example, if any component is less than 0 or greater than 255, the method will throw an exception to indicate invalid input.

Tri(int, int)

Draws a triangle at the current position with the specified width and height.

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

Parameters

width [int](#)

The width of the triangle.

height [int](#)

The height of the triangle.

Exceptions

CanvasException

Thrown when the width or height is less than or equal to 0.

WriteText(string)

Draws the specified text at the current drawing position on the canvas.

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

Parameters

text [string](#)

The text to be drawn on the canvas.

Remarks

This method checks if the provided text is null or empty and throws a `CanvasException` if it is. If the text is valid, it uses a font ("Arial", 14) and draws the text at the current position on the canvas, using the current pen color. The text is drawn starting at the current position (xPos, yPos) on the canvas.

Exceptions

CanvasException

Thrown when the provided text is null or empty.

getBitmap()

Gets the current bitmap representation of the canvas.


```
public object getBitmap()
```

Returns

[object](#)

The bitmap containing the drawing.

Class AppCircle

Namespace: [ASE Assignment](#)

Assembly: ASE_Assignment.dll

Represents a command to draw a circle on the canvas. Inherits from BOOSE.CommandTwoParameters and handles drawing circles on the canvas.

```
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. This constructor is the default constructor with no parameters.

```
public AppCircle()
```

AppCircle(Canvas, bool)

Initializes a new instance of the [AppCircle](#) class with the specified canvas and filled status.

```
public AppCircle(Canvas c, bool filled)
```

Parameters

c Canvas

The canvas where the circle will be drawn.

filled [bool](#)

Indicates whether the circle should be filled.

Methods

CheckParameters(string[])

Checks and validates the parameters for the Circle command. Ensures that exactly two parameters are passed to the command.

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

Parameters

parameterList [string](#)[]

The list of parameters to be validated.

Remarks

This method checks that the first parameter is a valid positive integer for the radius, and that the second parameter is a valid boolean representing whether the circle should be filled.

Exceptions

CommandException

Thrown when the parameters are invalid.

Execute()

Executes the command to draw a circle on the canvas.

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

Remarks

This method parses the radius from the parameters and draws the circle on the provided canvas using the specified filled status.

Class AppCommandFactory

Namespace: [ASE Assignment](#)

Assembly: ASE_Assignment.dll

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

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

Inheritance

[object](#) ← CommandFactory ← AppCommandFactory

Implements

ICommandFactory

Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Constructors

AppCommandFactory()

Initializes a new instance of the AppCommandFactory class.

```
public AppCommandFactory()
```

Methods

MakeCommand(string)

Creates a command object based on the given command type.

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

Parameters

commandType [string](#) 

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

Returns

ICommand

An ICommand object representing the specified command type.

Remarks

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

Class AppRect


Namespace: [ASE Assignment](#)

Assembly: ASE_Assignment.dll

Represents the rectangle command in the application. Inherits from BOOSE.CommandThreeParameters and handles drawing rectangles on the canvas.

```
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. This constructor is the default constructor with no parameters.

```
public AppRect()
```

AppRect(Canvas, bool)

Initializes a new instance of the [AppRect](#) class with specified canvas and fill option.

```
public AppRect(Canvas c, bool filled)
```

Parameters

c Canvas

The canvas on which to draw the rectangle.

filled [bool](#)

Indicates whether the rectangle should be filled.

Methods

CheckParameters(string[])

Validates the parameters for the rectangle drawing command. Ensures that exactly three parameters are passed to the command.

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

Parameters

parameterList [string](#)[]

The list of parameters to be validated.

Remarks

This method checks that the first two parameters are valid positive integers for the width and height, and the third parameter is a valid boolean indicating whether the rectangle should be filled.

Exceptions

CommandException

Thrown if the parameters are invalid.

Execute()

Executes the rectangle drawing command.

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

Remarks

This method parses the width and height from the parameters and draws the rectangle on the provided canvas using the specified filled status.

Class AppTri

Namespace: [ASE Assignment](#)

Assembly: ASE_Assignment.dll

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

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

Inheritance

[object](#) ← Command ← CanvasCommand ← CommandOneParameter ← CommandTwoParameters ← AppTri

Implements

ICommand

Inherited Members

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

Constructors

AppTri()

Initializes a new instance of the [AppTri](#) class. This constructor is the default constructor with no parameters.

```
public AppTri()
```

AppTri(Canvas, int, int)

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

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

Parameters

c Canvas

The canvas on which to draw the triangle.

width [int](#)

The width of the triangle.

height [int](#)

The height of the triangle.

Methods

CheckParameters(string[])

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

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

Parameters

parameterList [string](#)[]

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

Remarks

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

Exceptions

CommandException

Thrown if the number of parameters is incorrect or invalid.

Execute()

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

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

Remarks

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

Class AppWrite


Namespace: [ASE Assignment](#)

Assembly: ASE_Assignment.dll

Represents a command that writes text to a canvas. Inherits from BOOSE.CommandOneParameter and requires one parameter, which is the text to be written.

```
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. This constructor is the default constructor with no parameters.

```
public AppWrite()
```

AppWrite(Canvas, string)

Initializes a new instance of the [AppWrite](#) class with a specified canvas and text.

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

Parameters

c Canvas

The canvas on which the text will be written.

text [string](#)

The text to write on the canvas.

Methods

CheckParameters(string[])

Checks the validity of the parameters for the [AppWrite](#) command. Ensures that exactly one parameter is provided, which is the text to be written.

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

Parameters

parameterList [string](#)[]

An array of strings representing the parameters.

Remarks

This method ensures that the parameter list contains exactly one item and that the text is not null or empty.

Exceptions

CommandException

Thrown when the parameter list does not contain exactly one item.

Execute()

Executes the command by writing the specified text to the canvas. It retrieves the text to be written from the command's parameters and passes it to the canvas' [WriteText\(string\)](#) method to display it.

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

Remarks

This method retrieves the text parameter from the command and calls the [WriteText\(string\)](#) method to display the text on the canvas.

Class Form1







Namespace: [ASE Assignment](#)

Assembly: ASE_Assignment.dll






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

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

Inheritance

[object](#)  ← [MarshalByRefObject](#)  ← [Component](#)  ← [Control](#)  ← [ScrollableControl](#)  ← [ContainerControl](#)  ← [Form](#)  ← Form1

Implements

[IDropTarget](#) , [ISynchronizeInvoke](#) , [IWin32Window](#) , [IBindableComponent](#) , [IComponent](#) , [IDisposable](#) , [IContainerControl](#) 

Inherited Members

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

[Form.OnRightToLeftLayoutChanged\(EventArgs\)](#), [Form.OnShown\(EventArgs\)](#),
[Form.OnTextChanged\(EventArgs\)](#), [Form.ProcessCmdKey\(ref Message, Keys\)](#),
[Form.ProcessDialogKey\(Keys\)](#), [Form.ProcessDialogChar\(char\)](#),
[Form.ProcessKeyPreview\(ref Message\)](#), [Form.ProcessTabKey\(bool\)](#),
[Form.RemoveOwnedForm\(Form\)](#), [Form.Select\(bool, bool\)](#),
[Form.ScaleMinMaxSize\(float, float, bool\)](#),
[Form.GetScaledBounds\(Rectangle, SizeF, BoundsSpecified\)](#),
[Form.ScaleControl\(SizeF, BoundsSpecified\)](#), [Form.SetBoundsCore\(int, int, int, int, BoundsSpecified\)](#),
[Form.SetClientSizeCore\(int, int\)](#), [Form.SetDesktopBounds\(int, int, int, int\)](#),
[Form.SetDesktopLocation\(int, int\)](#), [Form.Show\(IWin32Window\)](#), [Form.ShowDialog\(\)](#),
[Form.ShowDialog\(IWin32Window\)](#), [Form.ToString\(\)](#), [Form.UpdateDefaultButton\(\)](#),
[Form.OnResizeBegin\(EventArgs\)](#), [Form.OnResizeEnd\(EventArgs\)](#),
[Form.OnStyleChanged\(EventArgs\)](#), [Form.ValidateChildren\(\)](#),
[Form.ValidateChildren\(ValidationConstraints\)](#), [Form.WndProc\(ref Message\)](#), [Form.AcceptButton](#),
[Form.ActiveForm](#), [Form.ActiveMdiChild](#), [Form.AllowTransparency](#), [Form.AutoScroll](#),
[Form.AutoSize](#), [Form.AutoSizeMode](#), [Form.AutoValidate](#), [Form.BackColor](#),
[Form.FormBorderStyle](#), [Form.CancelButton](#), [Form.ClientSize](#), [Form.ControlBox](#),
[Form.CreateParams](#), [Form.DefaultImeMode](#), [Form.DefaultSize](#), [Form.DesktopBounds](#),
[Form.DesktopLocation](#), [Form.DialogResult](#), [Form.HelpButton](#), [Form.Icon](#), [Form.IsMdiChild](#),
[Form.IsMdiContainer](#), [Form.IsRestrictedWindow](#), [Form.KeyPreview](#), [Form.Location](#),
[Form.MaximizedBounds](#), [Form.MaximumSize](#), [Form.MainMenuStrip](#), [Form.MinimumSize](#),
[Form.MaximizeBox](#), [Form.MdiChildren](#), [Form.MdiChildrenMinimizedAnchorBottom](#),
[Form.MdiParent](#), [Form.MinimizeBox](#), [Form.Modal](#), [Form.Opacity](#), [Form.OwnedForms](#),
[Form.Owner](#), [Form.RestoreBounds](#), [Form.RightToLeftLayout](#), [Form.ShowInTaskbar](#),
[Form.ShowIcon](#), [Form.ShowWithoutActivation](#), [Form.Size](#), [Form.SizeGripStyle](#),
[Form.StartPosition](#), [Form.Text](#), [Form.TopLevel](#), [Form.TopMost](#), [Form.TransparencyKey](#),
[Form.WindowState](#), [Form.AutoSizeChanged](#), [Form.AutoValidateChanged](#),
[Form.HelpButtonClicked](#), [Form.MaximizedBoundsChanged](#), [Form.MaximumSizeChanged](#),
[Form.MinimumSizeChanged](#), [Form.Activated](#), [Form.Deactivate](#), [Form.FormClosing](#),
[Form.FormClosed](#), [Form.Load](#), [Form.MdiChildActivate](#), [Form.MenuComplete](#),
[Form.MenuStart](#), [Form.InputLanguageChanged](#), [Form.InputLanguageChanging](#),
[Form.RightToLeftLayoutChanged](#), [Form.Shown](#), [Form.DpiChanged](#), [Form.ResizeBegin](#),
[Form.ResizeEnd](#), [ContainerControl.OnAutoValidateChanged\(EventArgs\)](#),
[ContainerControl.OnMove\(EventArgs\)](#), [ContainerControl.OnParentChanged\(EventArgs\)](#),
[ContainerControl.PerformAutoScale\(\)](#), [ContainerControl.RescaleConstantsForDpi\(int, int\)](#),
[ContainerControl.Validate\(\)](#), [ContainerControl.Validate\(bool\)](#),
[ContainerControl.AutoScaleDimensions](#), [ContainerControl.AutoScaleFactor](#),
[ContainerControl.AutoScaleMode](#), [ContainerControl.BindingContext](#),
[ContainerControl.CanEnableIme](#), [ContainerControl.ActiveControl](#),
[ContainerControl.CurrentAutoScaleDimensions](#), [ContainerControl.ParentForm](#),

[ScrollableControl.ScrollStateAutoScrolling](#) , [ScrollableControl.ScrollStateHScrollVisible](#) ,
[ScrollableControl.ScrollStateVScrollVisible](#) , [ScrollableControl.ScrollStateUserHasScrolled](#) ,
[ScrollableControl.ScrollStateFullDrag](#) , [ScrollableControl.GetScrollState\(int\)](#) ,
[ScrollableControl.OnMouseWheel\(MouseEventArgs\)](#) ,
[ScrollableControl.OnRightToLeftChanged\(EventArgs\)](#) ,
[ScrollableControl.OnPaintBackground\(PaintEventArgs\)](#) ,
[ScrollableControl.OnPaddingChanged\(EventArgs\)](#) , [ScrollableControl.SetDisplayRectLocation\(int, int\)](#) ,
[ScrollableControl.ScrollControlIntoView\(Control\)](#) , [ScrollableControl.ScrollToControl\(Control\)](#) ,
[ScrollableControl.OnScroll\(ScrollEventArgs\)](#) , [ScrollableControl.SetAutoScrollMargin\(int, int\)](#) ,
[ScrollableControl.SetScrollState\(int, bool\)](#) , [ScrollableControl.AutoScrollMargin](#) ,
[ScrollableControl.AutoScrollPosition](#) , [ScrollableControl.AutoScrollMinSize](#) ,
[ScrollableControl.DisplayRectangle](#) , [ScrollableControl.HScroll](#) , [ScrollableControl.HorizontalScroll](#) ,
[ScrollableControl.VScroll](#) , [ScrollableControl.VerticalScroll](#) , [ScrollableControl.Scroll](#) ,
[Control.GetAccessibilityObjectById\(int\)](#) , [Control.SetAutoSizeMode\(AutoSizeMode\)](#) ,
[Control.GetAutoSizeMode\(\)](#) , [Control.GetPreferredSize\(Size\)](#) ,
[Control.AccessibilityNotifyClients\(AccessibleEvents, int\)](#) ,
[Control.AccessibilityNotifyClients\(AccessibleEvents, int, int\)](#) , [Control.BeginInvoke\(Delegate\)](#) ,
[Control.BeginInvoke\(Action\)](#) , [Control.BeginInvoke\(Delegate, params object\[\]\)](#) ,
[Control.BringToFront\(\)](#) , [Control.Contains\(Control\)](#) , [Control.CreateGraphics\(\)](#) ,
[Control.CreateControl\(\)](#) , [Control.DestroyHandle\(\)](#) , [Control.DoDragDrop\(object, DragDropEffects\)](#) ,
[Control.DoDragDrop\(object, DragDropEffects, Bitmap, Point, bool\)](#) ,
[Control.DrawToBitmap\(Bitmap, Rectangle\)](#) , [Control.EndInvoke\(IAsyncResult\)](#) , [Control.FindForm\(\)](#) ,
[Control.GetTopLevel\(\)](#) , [Control.RaiseKeyEvent\(object, KeyEventArgs\)](#) ,
[Control.RaiseMouseEvent\(object, MouseEventArgs\)](#) , [Control.Focus\(\)](#) ,
[Control.FromChildHandle\(nint\)](#) , [Control.FromHandle\(nint\)](#) ,
[Control.GetChildAtPoint\(Point, GetChildAtPointSkip\)](#) , [Control.GetChildAtPoint\(Point\)](#) ,
[Control.GetContainerControl\(\)](#) , [Control.GetNextControl\(Control, bool\)](#) ,
[Control.GetStyle\(ControlStyles\)](#) , [Control.Hide\(\)](#) , [Control.InitLayout\(\)](#) , [Control.Invalidate\(Region\)](#) ,
[Control.Invalidate\(Region, bool\)](#) , [Control.Invalidate\(\)](#) , [Control.Invalidate\(bool\)](#) ,
[Control.Invalidate\(Rectangle\)](#) , [Control.Invalidate\(Rectangle, bool\)](#) , [Control.Invoke\(Action\)](#) ,
[Control.Invoke\(Delegate\)](#) , [Control.Invoke\(Delegate, params object\[\]\)](#) ,
[Control.Invoke<T>\(Func<T>\)](#) , [Control.InvokePaint\(Control, PaintEventArgs\)](#) ,
[Control.InvokePaintBackground\(Control, PaintEventArgs\)](#) , [Control.IsKeyLocked\(Keys\)](#) ,
[Control.IsInputChar\(char\)](#) , [Control.IsInputKey\(Keys\)](#) , [Control.IsMnemonic\(char, string\)](#) ,
[Control.LogicalToDeviceUnits\(int\)](#) , [Control.LogicalToDeviceUnits\(Size\)](#) ,
[Control.ScaleBitmapLogicalToDevice\(ref Bitmap\)](#) , [Control.NotifyInvalidate\(Rectangle\)](#) ,
[Control.InvokeOnClick\(Control, EventArgs\)](#) , [Control.OnAutoSizeChanged\(EventArgs\)](#) ,
[Control.OnBackColorChanged\(EventArgs\)](#) , [Control.OnBindingContextChanged\(EventArgs\)](#) ,
[Control.OnCausesValidationChanged\(EventArgs\)](#) , [Control.OnContextMenuStripChanged\(EventArgs\)](#) ,
[Control.OnCursorChanged\(EventArgs\)](#) , [Control.OnDataContextChanged\(EventArgs\)](#) ,

[Control.OnDockChanged\(EventArgs\)](#), [Control.OnForeColorChanged\(EventArgs\)](#),
[Control.OnNotifyMessage\(Message\)](#), [Control.OnParentBackColorChanged\(EventArgs\)](#),
[Control.OnParentBackgroundImageChanged\(EventArgs\)](#),
[Control.OnParentBindingContextChanged\(EventArgs\)](#), [Control.OnParentCursorChanged\(EventArgs\)](#),
[Control.OnParentDataContextChanged\(EventArgs\)](#), [Control.OnParentEnabledChanged\(EventArgs\)](#),
[Control.OnParentFontChanged\(EventArgs\)](#), [Control.OnParentForeColorChanged\(EventArgs\)](#),
[Control.OnParentRightToLeftChanged\(EventArgs\)](#), [Control.OnParentVisibleChanged\(EventArgs\)](#),
[Control.OnPrint\(PaintEventArgs\)](#), [Control.OnTabIndexChanged\(EventArgs\)](#),
[Control.OnTabStopChanged\(EventArgs\)](#), [Control.OnClick\(EventArgs\)](#),
[Control.OnClientSizeChanged\(EventArgs\)](#), [Control.OnControlAdded\(ControlEventArgs\)](#),
[Control.OnControlRemoved\(ControlEventArgs\)](#), [Control.OnLocationChanged\(EventArgs\)](#),
[Control.OnDoubleClick\(EventArgs\)](#), [Control.OnDragEnter\(DragEventArgs\)](#),
[Control.OnDragOver\(DragEventArgs\)](#), [Control.OnDragLeave\(EventArgs\)](#),
[Control.OnDragDrop\(DragEventArgs\)](#), [Control.OnGiveFeedback\(GiveFeedbackEventArgs\)](#),
[Control.InvokeGotFocus\(Control, EventArgs\)](#), [Control.OnHelpRequested\(HelpEventArgs\)](#),
[Control.OnInvalidated\(InvalidateEventArgs\)](#), [Control.OnKeyDown\(KeyEventArgs\)](#),
[Control.OnKeyPress\(KeyPressEventArgs\)](#), [Control.OnKeyUp\(KeyEventArgs\)](#),
[Control.OnLeave\(EventArgs\)](#), [Control.InvokeLostFocus\(Control, EventArgs\)](#),
[Control.OnLostFocus\(EventArgs\)](#), [Control.OnMarginChanged\(EventArgs\)](#),
[Control.OnMouseDoubleClick\(MouseEventArgs\)](#), [Control.OnMouseClick\(MouseEventArgs\)](#),
[Control.OnMouseCaptureChanged\(EventArgs\)](#), [Control.OnMouseDown\(MouseEventArgs\)](#),
[Control.OnMouseEnter\(EventArgs\)](#), [Control.OnMouseLeave\(EventArgs\)](#),
[Control.OnDpiChangedBeforeParent\(EventArgs\)](#), [Control.OnDpiChangedAfterParent\(EventArgs\)](#),
[Control.OnMouseHover\(EventArgs\)](#), [Control.OnMouseMove\(MouseEventArgs\)](#),
[Control.OnMouseUp\(MouseEventArgs\)](#),
[Control.OnQueryContinueDrag\(QueryContinueDragEventArgs\)](#),
[Control.OnRegionChanged\(EventArgs\)](#), [Control.OnPreviewKeyDown\(PreviewKeyDownEventArgs\)](#),
[Control.OnSizeChanged\(EventArgs\)](#), [Control.OnChangeUICues\(UICuesEventArgs\)](#),
[Control.OnSystemColorsChanged\(EventArgs\)](#), [Control.OnValidating\(CancelEventArgs\)](#),
[Control.OnValidated\(EventArgs\)](#), [Control.PerformLayout\(\)](#), [Control.PerformLayout\(Control, string\)](#),
[Control.PointToClient\(Point\)](#), [Control.PointToScreen\(Point\)](#),
[Control.PreProcessMessage\(ref Message\)](#), [Control.PreProcessControlMessage\(ref Message\)](#),
[Control.ProcessKeyEventArgs\(ref Message\)](#), [Control.ProcessKeyMessage\(ref Message\)](#),
[Control.RaiseDragEvent\(object, DragEventArgs\)](#), [Control.RaisePaintEvent\(object, PaintEventArgs\)](#),
[Control.RecreateHandle\(\)](#), [Control.RectangleToClient\(Rectangle\)](#),
[Control.RectangleToScreen\(Rectangle\)](#), [Control.ReflectMessage\(nint, ref Message\)](#),
[Control.Refresh\(\)](#), [Control.ResetMouseEventArgs\(\)](#), [Control.ResetText\(\)](#), [Control.ResumeLayout\(\)](#),
[Control.ResumeLayout\(bool\)](#), [Control.Scale\(SizeF\)](#), [Control.Select\(\)](#),
[Control.SelectNextControl\(Control, bool, bool, bool, bool\)](#), [Control.SendToBack\(\)](#),
[Control.SetBounds\(int, int, int, int\)](#), [Control.SetBounds\(int, int, int, int, BoundsSpecified\)](#),

[Control.SizeFromClientSize\(Size\)](#), [Control.SetStyle\(ControlStyles, bool\)](#), [Control.SetTopLevel\(bool\)](#), [Control.RtlTranslateAlignment\(HorizontalAlignment\)](#), [Control.RtlTranslateAlignment\(LeftRightAlignment\)](#), [Control.RtlTranslateAlignment\(ContentAlignment\)](#), [Control.RtlTranslateHorizontal\(HorizontalAlignment\)](#), [Control.RtlTranslateLeftRight\(LeftRightAlignment\)](#), [Control.RtlTranslateContent\(ContentAlignment\)](#), [Control.Show\(\)](#), [Control.SuspendLayout\(\)](#), [Control.Update\(\)](#), [Control.UpdateBounds\(\)](#), [Control.UpdateBounds\(int, int, int, int\)](#), [Control.UpdateBounds\(int, int, int, int, int, int\)](#), [Control.UpdateZOrder\(\)](#), [Control.UpdateStyles\(\)](#), [Control.OnImeModeChanged\(EventArgs\)](#), [Control.AccessibilityObject](#), [Control.AccessibleDefaultActionDescription](#), [Control.AccessibleDescription](#), [Control.AccessibleName](#), [Control.AccessibleRole](#), [Control.AllowDrop](#), [Control.Anchor](#), [Control.AutoScrollOffset](#), [Control.LayoutEngine](#), [Control.DataContext](#), [Control.BackgroundImage](#), [Control.BackgroundImageLayout](#), [Control.Bottom](#), [Control.Bounds](#), [Control.CanFocus](#), [Control.CanRaiseEvents](#), [Control.CanSelect](#), [Control.Capture](#), [Control.CausesValidation](#), [Control.CheckForIllegalCrossThreadCalls](#), [Control.ClientRectangle](#), [Control.CompanyName](#), [Control.ContainsFocus](#), [Control.ContextMenuStrip](#), [Control.Controls](#), [Control.Created](#), [Control.Cursor](#), [Control.DataBindings](#), [Control.DefaultBackColor](#), [Control.DefaultCursor](#), [Control.DefaultFont](#), [Control.DefaultForeColor](#), [Control.DefaultMargin](#), [Control.DefaultMaximumSize](#), [Control.DefaultMinimumSize](#), [Control.DefaultPadding](#), [Control.DeviceDpi](#), [Control.IsDisposed](#), [Control.Disposing](#), [Control.Dock](#), [Control.DoubleBuffered](#), [Control.Enabled](#), [Control.Focused](#), [Control.Font](#), [Control.FontHeight](#), [Control.ForeColor](#), [Control.Handle](#), [Control.HasChildren](#), [Control.Height](#), [Control.IsHandleCreated](#), [Control.InvokeRequired](#), [Control.IsAccessible](#), [Control.IsAncestorSiteInDesignMode](#), [Control.IsMirrored](#), [Control.Left](#), [Control.Margin](#), [Control.ModifierKeys](#), [Control.MouseButtons](#), [Control.MousePosition](#), [Control.Name](#), [Control.Parent](#), [Control.ProductName](#), [Control.ProductVersion](#), [Control.RecreatingHandle](#), [Control.Region](#), [Control.RenderRightToLeft](#), [Control.ResizeRedraw](#), [Control.Right](#), [Control.RightToLeft](#), [Control.ScaleChildren](#), [Control.Site](#), [Control.TabIndex](#), [Control.TabStop](#), [Control.Tag](#), [Control.Top](#), [Control.TopLevelControl](#), [Control.ShowKeyboardCues](#), [Control.ShowFocusCues](#), [Control.UseWaitCursor](#), [Control.Visible](#), [Control.Width](#), [Control.PreferredSize](#), [Control.Padding](#), [Control.ImeMode](#), [Control.ImeModeBase](#), [Control.PropagatingImeMode](#), [Control.BackColorChanged](#), [Control.BackgroundImageChanged](#), [Control.BackgroundImageLayoutChanged](#), [Control.BindingContextChanged](#), [Control.CausesValidationChanged](#), [Control.ClientSizeChanged](#), [Control.ContextMenuStripChanged](#), [Control.CursorChanged](#), [Control.DockChanged](#), [Control.EnabledChanged](#), [Control.FontChanged](#), [Control.ForeColorChanged](#), [Control.LocationChanged](#), [Control.MarginChanged](#), [Control.RegionChanged](#), [Control.RightToLeftChanged](#), [Control.SizeChanged](#), [Control.TabIndexChanged](#), [Control.TabStopChanged](#), [Control.TextChanged](#), [Control.VisibleChanged](#), [Control.Click](#),

[Control.ControlAdded](#) , [Control.ControlRemoved](#) , [Control.DataContextChanged](#) ,
[Control.DragDrop](#) , [Control.DragEnter](#) , [Control.DragOver](#) , [Control.DragLeave](#) ,
[Control.GiveFeedback](#) , [Control.HandleCreated](#) , [Control.HandleDestroyed](#) ,
[Control.HelpRequested](#) , [Control.Invalidated](#) , [Control.PaddingChanged](#) , [Control.Paint](#) ,
[Control.QueryContinueDrag](#) , [Control.QueryAccessibilityHelp](#) , [Control.DoubleClick](#) ,
[Control.Enter](#) , [Control.GotFocus](#) , [Control.KeyDown](#) , [Control.KeyPress](#) , [Control.KeyUp](#) ,
[Control.Layout](#) , [Control.Leave](#) , [Control.LostFocus](#) , [Control.MouseClick](#) ,
[Control.MouseDoubleClick](#) , [Control.MouseCaptureChanged](#) , [Control.MouseDown](#) ,
[Control.MouseEnter](#) , [Control.MouseLeave](#) , [Control.DpiChangedBeforeParent](#) ,
[Control.DpiChangedAfterParent](#) , [Control.MouseHover](#) , [Control.MouseMove](#) , [Control.MouseUp](#) ,
[Control.MouseWheel](#) , [Control.Move](#) , [Control.PreviewKeyDown](#) , [Control.Resize](#) ,
[Control.ChangeUICues](#) , [Control.StyleChanged](#) , [Control.SystemColorsChanged](#) ,
[Control.Validating](#) , [Control.Validated](#) , [Control.ParentChanged](#) , [Control.ImeModeChanged](#) ,
[Component.Dispose\(\)](#) , [Component.GetService\(Type\)](#) , [Component.Container](#) ,
[Component.DesignMode](#) , [Component.Events](#) , [Component.Disposed](#) ,
[MarshalByRefObject.GetLifetimeService\(\)](#) , [MarshalByRefObject.InitializeLifetimeService\(\)](#) ,
[MarshalByRefObject.MemberwiseClone\(bool\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#)

Constructors

Form1()

Initializes a new instance of the [Form1](#) class. Sets up the canvas, command factory, program, and parser for the application.

```
public Form1()
```


Methods

Dispose(bool)

Clean up any resources being used.

```
protected override void Dispose(bool disposing)
```

Parameters

`disposing` [bool](#)

true if managed resources should be disposed; otherwise, false.

Namespace TestProject

Classes

[Test1](#)

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

Class Test1

Namespace: [TestProject](#)

Assembly: TestProject1.dll

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

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

Inheritance

[object](#)  ← Test1

Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,
[object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

Methods

Clear_Canvas()

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

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

Setup()

Initializes the canvas before each test.

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

Test_Circle_InvalidFilledParameter()

Tests the CheckParameters method of AppCircle with an invalid 'filled' parameter. Verifies that a CommandException is thrown.

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

Test_Circle_InvalidParameter()

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

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

Test_Circle_ValidParameters()

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

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

Test_DrawTo_InvalidCoordinates()

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

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

Test_DrawTo_ValidCoordinates()

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

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

Test_InvalidRadius_Negative()

Tests the CheckParameters method of AppCircle with a negative radius. Verifies that a CommandException is thrown.

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

Test_Invalid_Radius_Zero()

Tests the CheckParameters method of AppCircle with a radius of zero. Verifies that a CommandException is thrown.

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

Test_MoveTo_InvalidCoordinates()

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

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

Test_MoveTo_ValidCoordinates()

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

```
[TestMethod]
```

```
public void Test_MoveTo_ValidCoordinates()
```

Test_MultilineDrawingProgram()

Tests a multiline drawing program. Verifies that the final position is correct after performing multiple actions such as moving and drawing.

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

Test_Rect_InvalidFilledParameter()

Tests the CheckParameters method of AppRect with an invalid 'filled' parameter. Verifies that a CommandException is thrown.

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

Test_Rect_InvalidHeight_Zero()

Tests the CheckParameters method of AppRect with a zero height. Verifies that a CommandException is thrown.

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

Test_Rect_InvalidParameters()

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

```
[TestMethod]  
[ExpectedException(typeof(CommandException))]
```

```
public void Test_Rect_InvalidParameters()
```

Test_Rect_InvalidWidth_Zero()

Tests the CheckParameters method of AppRect with a zero width. Verifies that a CommandException is thrown.

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

Test_Rect_ValidParameters()

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

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

Test_SetColour_InvalidBlue()

Tests setting an invalid blue color value. Verifies that a CanvasException is thrown when the blue value is out of bounds.

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

Test_SetColour_InvalidColourRange()

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

```
[TestMethod]  
[ExpectedException(typeof(CanvasException))]
```

```
public void Test_SetColour_InvalidColourRange()
```

Test_SetColour_InvalidGreen()

Tests setting an invalid green color value. Verifies that a CanvasException is thrown when the green value is out of bounds.

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

Test_SetColour_InvalidRed()

Tests setting an invalid red color value. Verifies that a CanvasException is thrown when the red value is out of bounds.

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

Test_SetColour_ValidColour()

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

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

Test_TriInvalidHeight_Zero()

Tests the CheckParameters method of AppTri with a zero height. Verifies that a CommandException is thrown.

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

Test_TriInvalidParameter()

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

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

Test_TriInvalidWidth_Zero()

Tests the CheckParameters method of AppTri with a zero width. Verifies that a CommandException is thrown.

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

Test_ValidParameters()

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

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

Test_ValidText()

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

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

Test_Write_EmptyText()

Tests the CheckParameters method of AppWrite with empty text. Verifies that a CommandException is thrown.

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

Test_Write_InvalidParameters()

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

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

Test_Write_NullText()

Tests the CheckParameters method of AppWrite with null text. Verifies that a CommandException is thrown.

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