```
PathfindImageForm
- _image: Bitmap
- _originalImage: Bitmap
- _width: Int32
- _height: Int32
- _graph: Graph<Coord>
- _traversalObject: Traversal<Coord>
- prevStartNode: Coord
- startNode: Coord
- endNode: Coord
- _dijkstras: Dictionary<Coord, Dictionary<Coord, Coord>>
- _preCalculatedTree: Dictionary<Coord, Coord>
- nodes: Int32
- components: IContainer
- imageBox: PictureBox
- goButton: Button
- exitButton: Button
- textBox: RichTextBox
- workingButton: Button
- runningBox: RichTextBox
- nodeBox: RichTextBox
- invalidCord: Coord
+ PathfindImageForm(image: Bitmap, traversal: Traversal<Coord>, graph: Graph<Coord>)
- PathfindImageForm()
- ViewImageForm_Load(sender: Object, e: EventArgs) : Void
- ConvertImageBoxToBitmapCord(location: Point) : Coord
- RedrawImage() : Void
- DrawCross(center: Coord, colour: Color) : Void
- commitSin() : Void
- imageBox_Click(sender: Object, eventArgs: EventArgs) : Void
- exitButton_Click(sender: Object, e: EventArgs) : Void
- SetRunningBox() : Void
- GetDistanceBetweenNodes(start: Coord, goal: Coord) : Int32
- UpdateNodes() : Void
- goButton_Click(sender: Object, e: EventArgs) : Void
# Dispose(disposing: Boolean) : Void
- InitializeComponent() : Void
```

```
Program
+ Program()
- Main() : Void
- Run(menuInstance: Menu, CLILoggingInstance: Log, settingsInstance: Settings) : Void
- Testing() : Void
- RunSaveFile(menu: Menu, logger: Log) : Void
- RunNewImage(menu: Menu, logger: Log) : Void
```

```
TextWall
+ SaveWelcome(menuInstance: Menu) : Void
+ ImageWelcome(menuInstance: Menu) : Void
+ FileDetails(menuInstance: Menu, rawImage: RawImage) : Void
```

```
ViewImageForm
- _image: Bitmap
- _width: Int32
- _height: Int32
- components: IContainer
- imageBox: PictureBox
- nextButton: Button
+ ViewImageForm(image: Bitmap)
- ViewImageForm_Load(sender: Object, e: EventArgs) : Void
- nextButton_Click(sender: Object, e: EventArgs) : Void
# Dispose(disposing: Boolean) : Void
- InitializeComponent() : Void
```

```
Pathfinder
- _input: Double[,]
- _originalBitmap: Bitmap
- _graph: Graph<Coord>
- _traversal: Traversal<Coord>
+ Pathfinder(originalImage: Bitmap, input: Double[,])
+ Start() : Void
- InstanceClasses() : Void
```

```
CannyEdgeDetection
+ KernelSize { get; set; } : Int32
+ RedRatio { get; set; } : Double
+ GreenRatio { get; set; } : Double
+ BlueRatio { get; set; } : Double
+ Sigma { get; set; } : Double
+ LowerThreshold { get; set; } : Double
+ UpperThreshold { get; set; } : Double
+ CannyEdgeDetection()
+ CannyEdgeDetection(kernelSize: Int32, redRatio: Double, greenRatio: Double, blueRatio: Double, sigma: Double, lowerThreshold: Double, upperThreshold: Double)
+ BlackWhiteFilter(input: RGB[,]) : Double[,]
+ GaussianFilter(input: Double[,]) : Double[,]
+ CalculateGradients(input: Double[,], updateMenu: Action) : Gradients
- CalculateGradientX(input: Double[,], updateMenu: Action) : Double[,]
- CalculateGradientY(input: Double[,], updateMenu: Action) : Double[,]
+ CombineGradients(grads: Gradients) : Double[,]
+ GradientAngle(grads: Gradients) : Double[,]
+ MagnitudeThreshold(gradCombined: Double[,], gradAngle: Double[,]) : Double[,]
+ DoubleThreshold(input: Double[,]) : ThresholdPixel[,]
+ EdgeTrackingHysteresis(input: ThresholdPixel[,]) : Double[,]
```

```
Utility
+ GaussianDistribution(x: Int32, y: Int32, sigma: Double) : Double
+ Bound(l: Int32, h: Int32, v: Double) : Double
+ TryBound(l: Int32, h: Int32, v: Double, out value: Double) : Boolean
+ RadianToDegree(input: Double) : Double
+ DegreeToRadian(input: Double) : Double
+ MapRadiansToPixel(input: Double) : Double
+ CombineBitmap(a: Bitmap, b: Bitmap) : Bitmap
+ SplitImage(image: RGB[,]) : RGB[,][]
+ CombineQuadrants(alpha: Double[,], beta: Double[,], gamma: Double[,], delta: Double[,]) : Double[,]
+ InverseImage(image: Double[,]) : Double[,]
+ RebuildPath(prev: Dictionary<T, T>, goal: T) : T[]
+ IsYes(input: String) : Boolean
+ GetRed(pixel: Color) : Double
+ GetGreen(pixel: Color) : Double
+ GetBlue(pixel: Color) : Double
+ GetAverage(pixel: Color) : Double
+ GetIndustryAverage(pixel: Color) : Double
+ GetIfExists(pixel: Color) : Double
+ GetDistanceBetweenNodes(a: Coord, b: Coord) : Double
```

```
Logger
- _localApplication: Boolean
- Lock: Object
+ Logger(local: Boolean)
- Logger()
- CreateDirStructure() : Void
+ CreateRun() : Guid
+ WriteLineToRunFile(currentGuid: Guid, message: String) : Void
+ WriteLineToMaster(message: String) : Void
+ SaveBitmap(currentGuid: Guid, image: Double[,], name: String) : Void
+ SaveBitmap(currentGuid: Guid, image: Bitmap, name: String) : Void
+ Uuid() : Guid
```

```
Extensions
+ ToBitmap(array: Double[,]) : Bitmap
+ ToDoubles(image: Bitmap, getPixelFunction: Func<Color, Double>) : Double[,]
+ ToBitmap(array: RGB[,]) : Bitmap
+ ToGraph(doubles: Double[,]) : Graph<Coord>
+ SetPixel(pixels: RGB[,], x: Int32, y: Int32, toSetPixel: RGB) : Void
+ GetPixel(pixels: RGB[,], x: Int32, y: Int32) : RGB
```

```
PreprocessingException
+ PreprocessingException()
+ PreprocessingException(message: String)
+ PreprocessingException(message: String, innerException: Exception)
# PreprocessingException(info: SerializationInfo, context: StreamingContext)
```

```
Post
- _imageDoubles: Double[,]
+ Post(input: Double[,])
+ Start(embossCount: Int32) : Void
- EmbossImage(input: Double[,]) : Double[,]
- FillPixelGaps(input: Double[,]) : Double[,]
+ Result() : Double[,]
```

```
Stack<T>
- _stack: List<T>
+ Size { get; } : Int32
+ Stack()
+ Stack(input: IEnumerable<T>)
+ Peek() : T
+ Push(item: T) : Void
+ Pop() : T
+ IsEmpty() : Boolean
+ Contains(item: T) : Boolean
```

```
MinPriorityQueue<T>
- _priorityQueue: List<Double>
- _queue: List<T>
+ Size { get; } : Int32
- _size { get; } : Int32
+ MinPriorityQueue()
- Parent(index: Int32) : Int32
- Left(index: Int32) : Int32
- Right(index: Int32) : Int32
+ Enqueue(value: T, priority: Double) : Void
+ ChangePriority(item: T, newPriority: Double) : Void
+ Dequeue() : T
- MinifyHeap(index: Int32) : Void
- Swap(indexX: Int32, indexY: Int32) : Void
+ Contains(neighbor: T) : Boolean
```

```
Graph<T>
+ _data: Dictionary<T, List<T>>
+ Graph()
+ Graph(graph: Dictionary<T, List<T>>)
+ AddNode(key: T) : Void
+ RemoveNode(key: T) : Void
+ AddConnection(key: T, value: T) : Void
+ GetNode(key: T) : List<T>
+ GetAllNodes() : T[]
+ ContainsNode(node: T) : Boolean
+ Clear() : Void
```

```
MaxPriorityQueue<T>
- _priorityQueue: List<Int32>
- _queue: List<T>
+ Size { get; } : Int32
- _size { get; } : Int32
+ MaxPriorityQueue()
- GetParent(index: Int32) : T
- Parent(index: Int32) : Int32
- GetLeftChild(index: Int32) : T
- LeftChild(index: Int32) : Int32
- GetRightChild(index: Int32) : T
- RightChild(index: Int32) : Int32
- ShiftNodeUp(index: Int32) : Void
+ ChangePriority(item: T, newPriority: Int32) : Void
- ShiftNodeDown(index: Int32) : Void
+ Enqueue(item: T, priority: Int32) : Void
+ Dequeue() : T
- RemoveMax() : ValueTuple<T, Int32>
- Swap(indexX: Int32, indexY: Int32) : Void
+ Contains(neighbor: T) : Boolean
```

```
GraphException
+ GraphException()
+ GraphException(message: String)
+ GraphException(message: String, innerException: Exception)
# GraphException(info: SerializationInfo, context: StreamingContext)
```

```
Traversal<T>
- _graph: Graph<T>
+ Traversal(graph: Graph<T>)
+ DFS(start: T) : T[]
+ BFS(start: T) : T[]
+ AStar(start: T, goal: T, weightFunction: Func<T, T, Int32>) : Dictionary<T, T>
+ Dijkstra(start: T, goal: T, endOnFind: Boolean, nodeUpdate: Action) : Dictionary<T, T>
```

```
KernelException
+ KernelException()
+ KernelException(message: String)
+ KernelException(message: String, innerException: Exception)
# KernelException(info: SerializationInfo, context: StreamingContext)
```

```
Kernel<T>
- _image: T[,]
- _width: Int32
- _height: Int32
+ Kernel(image: T[,])
+ Constant(x: Int32, y: Int32, size: Int32, constant: T) : T[,]
+ Duplication(x: Int32, y: Int32, size: Int32) : T[,]
+ Gaussian(sigma: Double, size: Int32) : Double[,]
```

```
MapFileException
+ MapFileException()
+ MapFileException(message: String)
+ MapFileException(message: String, innerException: Exception)
# MapFileException(info: SerializationInfo, context: StreamingContext)
```

```
LoggerException
+ LoggerException()
+ LoggerException(message: String)
+ LoggerException(message: String, innerException: Exception)
# LoggerException(info: SerializationInfo, context: StreamingContext)
```

```
RoadDetection
- _filledBitmap: Bitmap
- _pathBitmap: Bitmap
- _pathDoubles: Double[,]
- _imageDoubles: Double[,]
- _threshold: Double
- _gen: Random
+ RoadDetection(imageDoubles: Double[,], threshold: Double)
+ Start(updateAction: Action) : Void
- FillImage(updateAction: Action) : List<Color>
- RemoveColor(toRemove: List<Color>, updateAction: Action) : Void
+ Result() : RoadResult
```

```
Queue<T>
- _queue: List<T>
+ Size { get; } : Int32
+ Queue()
+ Queue(input: IEnumerable<T>)
+ Enqueue(item: T) : Void
+ Dequeue() : T
+ IsEmpty() : Boolean
+ Contains(item: T) : Boolean
```

```
Matrix
- _matrix: Double[,]
+ X { get; } : Int32
+ Y { get; } : Int32
+ this[Int32 y, Int32 x] { get; set; } : Double
+ Matrix(matrix: Double[,])
+ Matrix(x: Int32, y: Int32)
+ Minimize() : Void
+ Convolution(a: Matrix, b: Matrix) : Double
+ GetEnumerator() : IEnumerator
+ operator +(a: Matrix, b: Matrix) : Matrix
+ operator -(a: Matrix, b: Matrix) : Matrix
+ operator *(a: Matrix, b: Matrix) : Matrix
+ operator *(a: Int32, b: Matrix) : Matrix
```

```
MatrixException
+ MatrixException()
+ MatrixException(message: String)
+ MatrixException(message: String, innerException: Exception)
# MatrixException(info: SerializationInfo, context: StreamingContext)
```

```
«interface»
IHandler
+ Start() : Void
+ Result() : Double[,]
```

```
SettingsException
+ SettingsException()
+ SettingsException(message: String)
+ SettingsException(message: String, innerException: Exception)
# SettingsException(info: SerializationInfo, context: StreamingContext)
```