# eda12131190311906

1.0

Generated by Doxygen 1.8.4

Sun May 26 2013 20:07:22

# Contents

# Chapter 1

# Namespace Index

## 1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

# Chapter 2

# Hierarchical Index

## 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

# Chapter 3

# Class Index

## 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

# Chapter 4

# Namespace Documentation

## 4.1 Package eda12131190311906

### Classes

- class AbortableBackgroundWorker

  *Executes an operation on a separate thread. Can be aborted without a wait time*
- class ApplicationSettings

  *Application settings*
- class Bubble

  *Bubble Sort Algorithm* http://en.wikipedia.org/wiki/Bubblesort
- class Bucket

  *Bucket Sort Algorithm* http://en.wikipedia.org/wiki/Bucket_sort
- class Comb

  *Comb Sort Algorithm* http://en.wikipedia.org/wiki/Comb_sort
- class Counting

  *Counting Sort Algorithm* http://en.wikipedia.org/wiki/Counting_sort
- class FrmMain

  *Main form / app*
- class Heap

  *Heap Sort Algorithm* http://en.wikipedia.org/wiki/Heapsort
- class Insertion

  *Insertion Sort Algorithm* http://en.wikipedia.org/wiki/Insertion_sort
- class Logging

  *Loggin class, provide a log model to the application*
- class Merge

  *Merge Sort Algorithm* http://en.wikipedia.org/wiki/Mergesort
- class **Program**

  *Main program*
- class Quick

  *Quick Sort Algorithm* http://en.wikipedia.org/wiki/Quicksort
- class Radix

  *Radix Sort Algorithm* http://en.wikipedia.org/wiki/Radix_sort
- class Report

  *Report algorithm execution to file and grafs*
- class Selection

  *Selection Sort Algorithm* http://en.wikipedia.org/wiki/Selection_sort

- class Shell

  *Shell Sort Algorithm* http://en.wikipedia.org/wiki/Shell_sort
- class StopwatchEx

  *Provides a set of methods and properties that you can use to accurately measure elapsed time. Extended Version*
- class SystemHelper

  *System Helper Utilities*

### 4.1.1 Detailed Description

Estruturas de Dados e Algoritmos (EDA) - Project I Tiago Conceicao N 11903 Goncalo Lampreia N 11906 https://code.google.com/p/eda12131190311906/

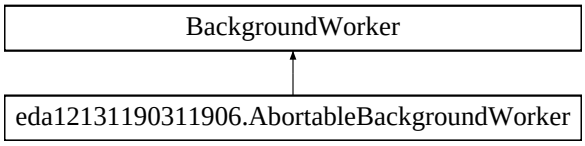
# Chapter 5

# Class Documentation

## 5.1 eda12131190311906.AbortableBackgroundWorker Class Reference

Executes an operation on a separate thread. Can be aborted without a wait time

Inheritance diagram for eda12131190311906.AbortableBackgroundWorker:




### Public Member Functions

- void Abort ()

  *Abort operation immediately*
- void AbortCancel ()

  *Abort operation immediately and try to cancel first*

### Protected Member Functions

- override void OnDoWork (DoWorkEventArgs e)

  *Occurs when M:System.ComponentModel.BackgroundWorker.RunWorkerAsync is called.*

### 5.1.1 Detailed Description

Executes an operation on a separate thread. Can be aborted without a wait time

Definition at line 10 of file AbortableBackgroundWorker.cs.

### 5.1.2 Member Function Documentation

#### 5.1.2.1 void eda12131190311906.AbortableBackgroundWorker.Abort ( ) [inline]

Abort operation immediately

Definition at line 37 of file AbortableBackgroundWorker.cs.

#### 5.1.2.2 void eda12131190311906.AbortableBackgroundWorker.AbortCancel ( ) [inline]

Abort operation immediately and try to cancel first

Definition at line 47 of file AbortableBackgroundWorker.cs.

#### 5.1.2.3 override void eda12131190311906.AbortableBackgroundWorker.OnDoWork ( DoWorkEventArgs *e* ) [inline], [protected]

Occurs when M:System.ComponentModel.BackgroundWorker.RunWorkerAsync is called.

Definition at line 20 of file AbortableBackgroundWorker.cs.

The documentation for this class was generated from the following file:

- AbortableBackgroundWorker.cs

## 5.2 eda12131190311906.ApplicationSettings Class Reference

Application settings

### Public Member Functions

- ApplicationSettings ()
  *Constructor, with default settings*
- override string ToString ()
  *Get a string represetantion of this class*

### Static Public Member Functions

- static void Reload (string filename)
  *Save settings to file*
- static void Reload ()
  *Reload settings from default file*
- static void Save (string filename)
  *Save settings to file*
- static void Save ()
  *Save settings to default file*

### Public Attributes

- const string Filename = "eda12131190311906.conf.xml"
  *Default filename to save application settings*

### Properties

- static ApplicationSettings Instance [get]
  *Settings instance*
- string ReportsPath [get, set]
  *Where to save reports to load with gnuplot*
- string GnuplotFullPath [get, set]

*Gnuplot executable path*

- bool AutoOpenPlot [get, set]

  *Auto open generated plot files (Gnuplot required)*
- byte NumberOfTests [get, set]

  *Number of tests to realize with sorting algorithms*
- byte ComputeAverageValueWith [get, set]

  *Compute time average repeating same code block x times*
- bool CutLowerHigherAverageValue [get, set]

  *Cut lower and higher time values for compute a better average*
- uint ArrayInitialSize [get, set]

  *Array initial size (First array size)*
- char ArrayGrowFactorType [get, set]

  *Array grow factor type*
- double ArrayGrowFactor [get, set]

  *Array grow factor*
- uint ArrayMinRandomNumber [get, set]

  *Array min random number*
- uint ArrayMaxRandomNumber [get, set]

  *Array max random number*
- char ArrayNumberGrowFactorType [get, set]

  *Array number grow factor type*
- double ArrayNumberGrowFactor [get, set]

  *Array numbers grow factor*
- bool ArrayRandomBetweenValues [get, set]

  *Array random numbers between min and max values*

### 5.2.1 Detailed Description

Application settings

Definition at line 17 of file ApplicationSettings.cs.

### 5.2.2 Constructor & Destructor Documentation

#### 5.2.2.1 eda12131190311906.ApplicationSettings.ApplicationSettings ( ) [inline]

Constructor, with default settings

Definition at line 132 of file ApplicationSettings.cs.

### 5.2.3 Member Function Documentation

#### 5.2.3.1 static void eda12131190311906.ApplicationSettings.Reload ( string *filename* ) [inline], [static]

Save settings to file

**Parameters**

| | |
|---|---|
| *filename* | File to save settings |

Definition at line 199 of file ApplicationSettings.cs.

#### 5.2.3.2 static void eda12131190311906.ApplicationSettings.Reload ( ) [inline], [static]

Reload settings from default file

Definition at line 221 of file ApplicationSettings.cs.

#### 5.2.3.3 static void eda12131190311906.ApplicationSettings.Save ( string *filename* ) [inline], [static]

Save settings to file

**Parameters**

| | |
|---|---|
| *filename* | File to save settings |

Definition at line 230 of file ApplicationSettings.cs.

#### 5.2.3.4 static void eda12131190311906.ApplicationSettings.Save ( ) [inline], [static]

Save settings to default file

Definition at line 247 of file ApplicationSettings.cs.

#### 5.2.3.5 override string eda12131190311906.ApplicationSettings.ToString ( ) [inline]

Get a string represetantion of this class

**Returns**

String represetantion of this class

Definition at line 159 of file ApplicationSettings.cs.

### 5.2.4 Member Data Documentation

#### 5.2.4.1 const string eda12131190311906.ApplicationSettings.Filename = "eda12131190311906.conf.xml"

Default filename to save application settings

Definition at line 23 of file ApplicationSettings.cs.

### 5.2.5 Property Documentation

#### 5.2.5.1 double eda12131190311906.ApplicationSettings.ArrayGrowFactor [get], [set]

Array grow factor

Definition at line 100 of file ApplicationSettings.cs.

#### 5.2.5.2 char eda12131190311906.ApplicationSettings.ArrayGrowFactorType [get], [set]

Array grow factor type

Definition at line 95 of file ApplicationSettings.cs.

#### 5.2.5.3 uint eda12131190311906.ApplicationSettings.ArrayInitialSize [get], [set]

Array initial size (First array size)

Definition at line 90 of file ApplicationSettings.cs.

#### 5.2.5.4 uint eda12131190311906.ApplicationSettings.ArrayMaxRandomNumber [get], [set]

Array max random number

Definition at line 110 of file ApplicationSettings.cs.

#### 5.2.5.5 uint eda12131190311906.ApplicationSettings.ArrayMinRandomNumber [get], [set]

Array min random number

Definition at line 105 of file ApplicationSettings.cs.

#### 5.2.5.6 double eda12131190311906.ApplicationSettings.ArrayNumberGrowFactor [get], [set]

Array numbers grow factor

Definition at line 120 of file ApplicationSettings.cs.

#### 5.2.5.7 char eda12131190311906.ApplicationSettings.ArrayNumberGrowFactorType [get], [set]

Array number grow factor type

Definition at line 115 of file ApplicationSettings.cs.

#### 5.2.5.8 bool eda12131190311906.ApplicationSettings.ArrayRandomBetweenValues [get], [set]

Array random numbers between min and max values

Definition at line 125 of file ApplicationSettings.cs.

#### 5.2.5.9 bool eda12131190311906.ApplicationSettings.AutoOpenPlot [get], [set]

Auto open generated plot files (Gnuplot required)

Definition at line 70 of file ApplicationSettings.cs.

#### 5.2.5.10 byte eda12131190311906.ApplicationSettings.ComputeAverageValueWith [get], [set]

Compute time average repeating same code block x times

Definition at line 80 of file ApplicationSettings.cs.

#### 5.2.5.11 bool eda12131190311906.ApplicationSettings.CutLowerHigherAverageValue [get], [set]

Cut lower and higher time values for compute a better average

Definition at line 85 of file ApplicationSettings.cs.

#### 5.2.5.12 string eda12131190311906.ApplicationSettings.GnuplotFullPath [get], [set]

Gnuplot executable path

Definition at line 65 of file ApplicationSettings.cs.

#### 5.2.5.13 ApplicationSettings eda12131190311906.ApplicationSettings.Instance [static], [get]

Settings instance

Definition at line 39 of file ApplicationSettings.cs.

#### 5.2.5.14 byte eda12131190311906.ApplicationSettings.NumberOfTests [get], [set]

Number of tests to realize with sorting algorithms

Definition at line 75 of file ApplicationSettings.cs.

#### 5.2.5.15 string eda12131190311906.ApplicationSettings.ReportsPath [get], [set]

Where to save reports to load with gnuplot

Definition at line 60 of file ApplicationSettings.cs.

The documentation for this class was generated from the following file:

- ApplicationSettings.cs

## 5.3 eda12131190311906.Bubble Class Reference

Bubble Sort Algorithm http://en.wikipedia.org/wiki/Bubblesort

### Static Public Member Functions

- static void Sort (int[] A)
  *Sort an array*

### 5.3.1 Detailed Description

Bubble Sort Algorithm http://en.wikipedia.org/wiki/Bubblesort

Definition at line 13 of file Bubble.cs.

### 5.3.2 Member Function Documentation

#### 5.3.2.1 static void eda12131190311906.Bubble.Sort ( int[] *A* ) [inline], [static]

Sort an array

**Parameters**

| | |
|---|---|
| *A* | Array to sort |

Definition at line 18 of file Bubble.cs.

The documentation for this class was generated from the following file:

- Bubble.cs

## 5.4 eda12131190311906.Bucket Class Reference

Bucket Sort Algorithm http://en.wikipedia.org/wiki/Bucket_sort

### Static Public Member Functions

- static void Sort (int[] A)
  *Sort an array*

### 5.4.1 Detailed Description

Bucket Sort Algorithm http://en.wikipedia.org/wiki/Bucket_sort

Definition at line 17 of file Bucket.cs.

### 5.4.2 Member Function Documentation

#### 5.4.2.1 static void eda12131190311906.Bucket.Sort ( int[] *A* ) [inline], [static]

Sort an array

**Parameters**

| | |
|---|---|
| *A* | Array to sort |

Definition at line 23 of file Bucket.cs.

The documentation for this class was generated from the following file:

- Bucket.cs

## 5.5 eda12131190311906.Comb Class Reference

Comb Sort Algorithm http://en.wikipedia.org/wiki/Comb_sort

### Static Public Member Functions

- static void Sort (int[] A)
  *Sort an array*

### 5.5.1 Detailed Description

Comb Sort Algorithm http://en.wikipedia.org/wiki/Comb_sort

Definition at line 13 of file Comb.cs.

### 5.5.2 Member Function Documentation

#### 5.5.2.1 static void eda12131190311906.Comb.Sort ( int[] *A* ) [inline], [static]

Sort an array

**Parameters**

| | |
|---|---|
| *A* | Array to sort |

Definition at line 18 of file Comb.cs.

The documentation for this class was generated from the following file:

- Comb.cs

## 5.6 eda12131190311906.Counting Class Reference

Counting Sort Algorithm http://en.wikipedia.org/wiki/Counting_sort

### Static Public Member Functions

- static void Sort (int[] A)

  *Sort an array*

### 5.6.1 Detailed Description

Counting Sort Algorithm http://en.wikipedia.org/wiki/Counting_sort

Definition at line 13 of file Counting.cs.

### 5.6.2 Member Function Documentation

#### 5.6.2.1 static void eda12131190311906.Counting.Sort ( int[] *A* ) [inline], [static]

Sort an array

**Parameters**

| | |
|---|---|
| *A* | Array to sort |

Definition at line 18 of file Counting.cs.

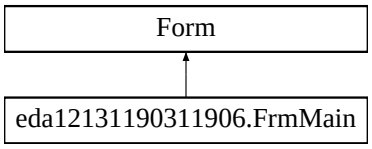
The documentation for this class was generated from the following file:

- Counting.cs

## 5.7 eda12131190311906.FrmMain Class Reference

Main form / app

Inheritance diagram for eda12131190311906.FrmMain:




### Public Member Functions

- FrmMain ()
  *Constructor*

### Protected Member Functions

- override void **OnLoad** (EventArgs e)
- override void Dispose (bool disposing)
  *Clean up any resources being used.*

### Properties

- Stopwatch Stopwatcher `[get, set]`
  *Background operation timer*

## 5.7.1 Detailed Description

Main form / app

Definition at line 21 of file FrmMain.cs.

## 5.7.2 Constructor & Destructor Documentation

### 5.7.2.1 eda12131190311906.FrmMain.FrmMain ( ) `[inline]`

Constructor

Definition at line 34 of file FrmMain.cs.

## 5.7.3 Member Function Documentation

### 5.7.3.1 override void eda12131190311906.FrmMain.Dispose ( bool *disposing* ) `[inline]`, `[protected]`

Clean up any resources being used.

**Parameters**

| | |
|---|---|
| *disposing* | true if managed resources should be disposed; otherwise, false. |

Definition at line 14 of file FrmMain.Designer.cs.

## 5.7.4 Property Documentation

### 5.7.4.1 Stopwatch eda12131190311906.FrmMain.Stopwatcher `[get]`, `[set]`

Background operation timer

Definition at line 27 of file FrmMain.cs.

The documentation for this class was generated from the following files:

- FrmMain.cs
- FrmMain.Designer.cs

## 5.8 eda12131190311906.Heap Class Reference

Heap Sort Algorithm http://en.wikipedia.org/wiki/Heapsort

### Static Public Member Functions

- static void Sort (int[] A)

  *Sort an array*

### 5.8.1 Detailed Description

Heap Sort Algorithm http://en.wikipedia.org/wiki/Heapsort

Definition at line 16 of file Heap.cs.

### 5.8.2 Member Function Documentation

#### 5.8.2.1 static void eda12131190311906.Heap.Sort ( int[] *A* ) [inline], [static]

Sort an array

**Parameters**

| | |
|---|---|
| *A* | Array to sort |

Definition at line 131 of file Heap.cs.

The documentation for this class was generated from the following file:

- Heap.cs

## 5.9 eda12131190311906.Insertion Class Reference

Insertion Sort Algorithm http://en.wikipedia.org/wiki/Insertion_sort

### Static Public Member Functions

- static void Sort (int[] A)

  *Sort an array*

### 5.9.1 Detailed Description

Insertion Sort Algorithm http://en.wikipedia.org/wiki/Insertion_sort

Definition at line 13 of file Insertion.cs.

### 5.9.2 Member Function Documentation

#### 5.9.2.1 static void eda12131190311906.Insertion.Sort ( int[] *A* ) [inline], [static]

Sort an array

**Parameters**

| | |
|---|---|
| *A* | Array to sort |

Definition at line 18 of file Insertion.cs.

The documentation for this class was generated from the following file:

- Insertion.cs

## 5.10 eda12131190311906.Logging.LogEventArgs Class Reference

Log event hander class

### Public Member Functions

- LogEventArgs (string text, bool isWriteLine, bool cleared)
  *Constructor*

### Properties

- string AddedText [get, set]
  *New text added to log*
- bool IsWriteLine [get, set]
  *Is text write using WriteLine*
- bool Cleared [get, set]
  *Is text cleared and set to empty*

### 5.10.1 Detailed Description

Log event hander class

Definition at line 22 of file Logging.cs.

### 5.10.2 Constructor & Destructor Documentation

#### 5.10.2.1 eda12131190311906.Logging.LogEventArgs.LogEventArgs ( string *text*, bool *isWriteLine*, bool *cleared* ) [inline]

Constructor

**Parameters**

| | |
|---|---|
| *text* | Text |

| | |
|---|---|
| *isWriteLine* | Is text write using WriteLine |
| *cleared* | Is text cleared and set to empty |

Definition at line 45 of file Logging.cs.

### 5.10.3 Property Documentation

#### 5.10.3.1 string eda12131190311906.Logging.LogEventArgs.AddedText [get], [set]

New text added to log

Definition at line 27 of file Logging.cs.

#### 5.10.3.2 bool eda12131190311906.Logging.LogEventArgs.Cleared [get], [set]

Is text cleared and set to empty

Definition at line 37 of file Logging.cs.

#### 5.10.3.3 bool eda12131190311906.Logging.LogEventArgs.IsWriteLine [get], [set]

Is text write using WriteLine

Definition at line 32 of file Logging.cs.

The documentation for this class was generated from the following file:

- Logging.cs

## 5.11 eda12131190311906.Logging Class Reference

Loggin class, provide a log model to the application

### Classes

- class LogEventArgs
  *Log event hander class*

### Public Member Functions

- delegate void LogEventHandler (Object sender, LogEventArgs e)
  *Log event handler delegate*
- Logging ()
  *Constructor*
- void Write (string text)
  *Write to log*
- void WriteLine ()
  *Write a new line to log*
- void WriteLine (string text)
  *Write to log*
- void Clear ()
  *Clear log text*

- bool WriteToFile (string path)

  *Write log to a default file (debug.log)*
- void WriteToFile ()

  *Write log to a default file (debug.log)*

### Protected Member Functions

- void OnLog (LogEventArgs args)

### Properties

- LogEventHandler Log

  *Log event, raised when something writes to log*
- string Header `[get, set]`

  *Gets or sets the log header text*
- string LogText `[get, set]`

  *Gets the string holding log*

## 5.11.1 Detailed Description

Loggin class, provide a log model to the application

Definition at line 16 of file Logging.cs.

## 5.11.2 Constructor & Destructor Documentation

### 5.11.2.1 eda12131190311906.Logging.Logging ( ) `[inline]`

Constructor

Definition at line 111 of file Logging.cs.

## 5.11.3 Member Function Documentation

### 5.11.3.1 void eda12131190311906.Logging.Clear ( ) `[inline]`

Clear log text

Definition at line 150 of file Logging.cs.

### 5.11.3.2 delegate void eda12131190311906.Logging.LogEventHandler ( Object *sender,* LogEventArgs *e* )

Log event handler delegate

**Parameters**

| | |
|---|---|
| *sender* | |
| *e* | |

### 5.11.3.3 void eda12131190311906.Logging.OnLog ( LogEventArgs *args* ) `[inline]`, `[protected]`

**Parameters**

| | |
|---|---|
| *args* | |

Definition at line 85 of file Logging.cs.

#### 5.11.3.4 void eda12131190311906.Logging.Write ( string *text* ) [inline]

Write to log

**Parameters**

| | |
|---|---|
| *text* | Text to write |

Definition at line 122 of file Logging.cs.

#### 5.11.3.5 void eda12131190311906.Logging.WriteLine ( ) [inline]

Write a new line to log

Definition at line 131 of file Logging.cs.

#### 5.11.3.6 void eda12131190311906.Logging.WriteLine ( string *text* ) [inline]

Write to log

**Parameters**

| | |
|---|---|
| *text* | Text to write |

Definition at line 141 of file Logging.cs.

#### 5.11.3.7 bool eda12131190311906.Logging.WriteToFile ( string *path* ) [inline]

Write log to a default file (debug.log)

**Parameters**

| | |
|---|---|
| *path* | Path to save file |

**Returns**

True if file write successfully, ortherwise false

Definition at line 161 of file Logging.cs.

#### 5.11.3.8 void eda12131190311906.Logging.WriteToFile ( ) [inline]

Write log to a default file (debug.log)

**Returns**

True if file write successfully, ortherwise false

Definition at line 190 of file Logging.cs.

### 5.11.4 Property Documentation

#### 5.11.4.1 string eda12131190311906.Logging.Header [get],[set]

Gets or sets the log header text

Definition at line 99 of file Logging.cs.

#### 5.11.4.2 LogEventHandler eda12131190311906.Logging.Log [add],[remove]

Log event, raised when something writes to log

Definition at line 69 of file Logging.cs.

#### 5.11.4.3 string eda12131190311906.Logging.LogText [get],[set]

Gets the string holding log

Definition at line 104 of file Logging.cs.

The documentation for this class was generated from the following file:

- Logging.cs

## 5.12 eda12131190311906.Merge Class Reference

Merge Sort Algorithm http://en.wikipedia.org/wiki/Mergesort

### Static Public Member Functions

- static void Sort (int[] A, int p, int q, int r)
  *Sort an array*
- static void Sort (int[] A)
  *Sort an array*

### 5.12.1 Detailed Description

Merge Sort Algorithm http://en.wikipedia.org/wiki/Mergesort

Definition at line 13 of file Merge.cs.

### 5.12.2 Member Function Documentation

#### 5.12.2.1 static void eda12131190311906.Merge.Sort ( int[] *A*, int *p*, int *q*, int *r* ) [inline],[static]

Sort an array

**Parameters**

| | |
|---|---|
| *A* | Array to sort |
| *p* | Start index |

| | |
|---|---|
| *q* | Middle index |
| *r* | Right index |

Definition at line 21 of file Merge.cs.

#### 5.12.2.2 static void eda12131190311906.Merge.Sort ( int[] *A* ) [inline],[static]

Sort an array

**Parameters**

| | |
|---|---|
| *A* | Array to sort |

Definition at line 61 of file Merge.cs.

The documentation for this class was generated from the following file:

- Merge.cs

## 5.13 eda12131190311906.Report.PlotLine Class Reference

Report item class, represents a single line on gnuplot files

### Public Member Functions

- PlotLine ()

  *Constructor*
- PlotLine (string xAxis)

  *Constructor*
- bool AddProfiler (StopwatchEx profiler)

  *Add a profiler*
- StopwatchEx AddProfiler (bool run)

  *Add a profiler*
- StopwatchEx AddProfiler ()

  *Add a profiler and run*

### Properties

- string XAxis [get, set]

  *X axis value for this line of results, 1st column on file*
- List< StopwatchEx > Columns [get, set]

  *Line columns data, each column represents a StopwatchEx holding a TimeSpan with execution time*

### 5.13.1 Detailed Description

Report item class, represents a single line on gnuplot files

Definition at line 27 of file Report.cs.

### 5.13.2 Constructor & Destructor Documentation

#### 5.13.2.1 eda12131190311906.Report.PlotLine.PlotLine ( ) [inline]

Constructor

Definition at line 45 of file Report.cs.

#### 5.13.2.2 eda12131190311906.Report.PlotLine.PlotLine ( string *xAxis* ) [inline]

Constructor

**Parameters**

| | |
|---|---|
| *xAxis* | X axis name for this line |

Definition at line 54 of file Report.cs.

### 5.13.3 Member Function Documentation

#### 5.13.3.1 bool eda12131190311906.Report.PlotLine.AddProfiler ( StopwatchEx *profiler* ) [inline]

Add a profiler

**Parameters**

| | |
|---|---|
| *profiler* | Profiler to add |

**Returns**

True if added, otherwise false (Duplicated name)

Definition at line 66 of file Report.cs.

#### 5.13.3.2 StopwatchEx eda12131190311906.Report.PlotLine.AddProfiler ( bool *run* ) [inline]

Add a profiler

**Parameters**

| | |
|---|---|
| *run* | Start profiling or not |

**Returns**

Profiler added to map

Definition at line 77 of file Report.cs.

#### 5.13.3.3 StopwatchEx eda12131190311906.Report.PlotLine.AddProfiler ( ) [inline]

Add a profiler and run

**Returns**

Profiler added to map

Definition at line 92 of file Report.cs.

### 5.13.4 Property Documentation

#### 5.13.4.1 List<StopwatchEx> eda12131190311906.Report.PlotLine.Columns [get], [set]

Line columns data, each column represents a StopwatchEx holding a TimeSpan with execution time

Definition at line 38 of file Report.cs.

#### 5.13.4.2 string eda12131190311906.Report.PlotLine.XAxis [get], [set]

X axis value for this line of results, 1st column on file

Definition at line 33 of file Report.cs.

The documentation for this class was generated from the following file:

- Report.cs

## 5.14 eda12131190311906.Quick Class Reference

Quick Sort Algorithm http://en.wikipedia.org/wiki/Quicksort

### Static Public Member Functions

- static void Sort (int[] A, int p, int r)
  *Sort an array*
- static void Sort (int[] A)
  *Sort an array*
- static void RandomizedSort (int[] A, int p, int r)
  *Randomized sort*
- static void RandomizedSort (int[] A)
  *Randomized sort*
- static void TailRecursiveSort (int[] A, int p, int r)
  *Tail recursive sort*
- static void TailRecursiveSort (int[] A)
  *Tail recursive sort*

### 5.14.1 Detailed Description

Quick Sort Algorithm http://en.wikipedia.org/wiki/Quicksort

Definition at line 16 of file Quick.cs.

### 5.14.2 Member Function Documentation

#### 5.14.2.1 static void eda12131190311906.Quick.RandomizedSort ( int[] *A*, int *p*, int *r* ) [inline], [static]

Randomized sort

**Parameters**

| | |
|---|---|
| *A* | Array to sort |
| *p* | Start index |
| *r* | End index |

Definition at line 79 of file Quick.cs.

#### 5.14.2.2 static void eda12131190311906.Quick.RandomizedSort ( int[] *A* ) [inline], [static]

Randomized sort

**Parameters**

| | |
|---|---|
| *A* | Array to sort |

Definition at line 91 of file Quick.cs.

#### 5.14.2.3 static void eda12131190311906.Quick.Sort ( int[] *A*, int *p*, int *r* ) [inline], [static]

Sort an array

**Parameters**

| | |
|---|---|
| *A* | Array to sort |
| *p* | Start index |
| *r* | End index |

Definition at line 23 of file Quick.cs.

#### 5.14.2.4 static void eda12131190311906.Quick.Sort ( int[] *A* ) [inline], [static]

Sort an array

**Parameters**

| | |
|---|---|
| *A* | Array to sort |

Definition at line 38 of file Quick.cs.

#### 5.14.2.5 static void eda12131190311906.Quick.TailRecursiveSort ( int[] *A*, int *p*, int *r* ) [inline], [static]

Tail recursive sort

**Parameters**

| | |
|---|---|
| *A* | Array to sort |
| *p* | Start index |
| *r* | End index |

Definition at line 118 of file Quick.cs.

#### 5.14.2.6 static void eda12131190311906.Quick.TailRecursiveSort ( int[] *A* ) [inline], [static]

Tail recursive sort

**Parameters**

| | |
|---|---|
| *A* | Array to sort |

Definition at line 133 of file Quick.cs.

The documentation for this class was generated from the following file:

- Quick.cs

## 5.15 eda121311903119O6.Radix Class Reference

Radix Sort Algorithm http://en.wikipedia.org/wiki/Radix_sort

### Static Public Member Functions

- static void Sort (int[] A)

  *Sort an array*

### 5.15.1 Detailed Description

Radix Sort Algorithm http://en.wikipedia.org/wiki/Radix_sort

Definition at line 16 of file Radix.cs.

### 5.15.2 Member Function Documentation

#### 5.15.2.1 static void eda121311903119O6.Radix.Sort ( int[] *A* ) [inline], [static]

Sort an array

**Parameters**

| | |
|---|---|
| *A* | Array to sort |

Definition at line 21 of file Radix.cs.

The documentation for this class was generated from the following file:

- Radix.cs

## 5.16 eda121311903119O6.Report Class Reference

Report algorithm execution to file and grafs

### Classes

- class PlotLine

  *Report item class, represents a single line on gnuplot files*

### Public Member Functions

- Report (string name)

  *Constructor*
- PlotLine GetPlotLine (string xAxis)

  *Get plot line based on X Axis value*
- void WriteToFile (string path)

  *Write reports to a file*
- void WriteToFile ()

  *Write reports to a file*

### Static Public Member Functions

- static Report BuildMaster (List< Report > reports)

  *Build a master report holding and comparing all reports*
- static void GenerateGnuplotFiles ()

  *Generate Gnuplot graf files*

### Properties

- string Name `[get, set]`

  *Gets or sets the report name*
- string XAxisLabel `[get, set]`

  *Gets or sets the X axis name / data name*
- string YAxisLabel `[get, set]`

  *Gets or sets the Y axis name / data name*
- List< string > Comments `[get, set]`

  *Comments to write on file header*
- List< string > PlotTitles `[get, set]`

  *Plot data titles*
- List< PlotLine > PlotLines `[get, set]`

  *Plot lines holding all data*

## 5.16.1 Detailed Description

Report algorithm execution to file and grafs

Definition at line 20 of file Report.cs.

## 5.16.2 Constructor & Destructor Documentation

### 5.16.2.1 eda12131190311906.Report.Report ( string *name* ) `[inline]`

Constructor

**Parameters**

| | |
|---|---|
| *name* | Report name |

Definition at line 138 of file Report.cs.

### 5.16.3 Member Function Documentation

#### 5.16.3.1 static Report eda12131190311906.Report.BuildMaster ( List< Report > *reports* ) [inline], [static]

Build a master report holding and comparing all reports

**Parameters**

| | |
|---|---|
| *reports* | List with all reports to include |

**Returns**

Master report

Definition at line 264 of file Report.cs.

#### 5.16.3.2 static void eda12131190311906.Report.GenerateGnuplotFiles ( ) [inline], [static]

Generate Gnuplot graf files

Definition at line 309 of file Report.cs.

#### 5.16.3.3 PlotLine eda12131190311906.Report.GetPlotLine ( string *xAxis* ) [inline]

Get plot line based on X Axis value

**Parameters**

| | |
|---|---|
| *xAxis* | X Axis value |

**Returns**

PlotLine holding line data

Definition at line 153 of file Report.cs.

#### 5.16.3.4 void eda12131190311906.Report.WriteToFile ( string *path* ) [inline]

Write reports to a file

**Parameters**

| | |
|---|---|
| *path* | Path to save the file |

Definition at line 169 of file Report.cs.

#### 5.16.3.5 void eda12131190311906.Report.WriteToFile ( ) [inline]

Write reports to a file

Definition at line 251 of file Report.cs.

### 5.16.4 Property Documentation

#### 5.16.4.1 List<string> eda12131190311906.Report.Comments [get], [set]

Comments to write on file header

Definition at line 120 of file Report.cs.

#### 5.16.4.2 string eda12131190311906.Report.Name [get], [set]

Gets or sets the report name

Definition at line 105 of file Report.cs.

#### 5.16.4.3 List<PlotLine> eda12131190311906.Report.PlotLines [get], [set]

Plot lines holding all data

Definition at line 130 of file Report.cs.

#### 5.16.4.4 List<string> eda12131190311906.Report.PlotTitles [get], [set]

Plot data titles

Definition at line 125 of file Report.cs.

#### 5.16.4.5 string eda12131190311906.Report.XAxisLabel [get], [set]

Gets or sets the X axis name / data name

Definition at line 110 of file Report.cs.

#### 5.16.4.6 string eda12131190311906.Report.YAxisLabel [get], [set]

Gets or sets the Y axis name / data name

Definition at line 115 of file Report.cs.

The documentation for this class was generated from the following file:

- Report.cs

## 5.17 eda12131190311906.Selection Class Reference

Selection Sort Algorithm http://en.wikipedia.org/wiki/Selection_sort

### Static Public Member Functions

- static void Sort (int[] A)
  *Sort an array*

### 5.17.1 Detailed Description

Selection Sort Algorithm http://en.wikipedia.org/wiki/Selection_sort

Definition at line 13 of file Selection.cs.

### 5.17.2 Member Function Documentation

#### 5.17.2.1 static void eda12131190311906.Selection.Sort ( int[] *A* ) [inline], [static]

Sort an array

**Parameters**

| | |
|---|---|
| *A* | Array to sort |

Definition at line 18 of file Selection.cs.

The documentation for this class was generated from the following file:

- Selection.cs

## 5.18 eda12131190311906.Shell Class Reference

Shell Sort Algorithm http://en.wikipedia.org/wiki/Shell_sort

### Static Public Member Functions

- static void Sort (int[] A)
  *Sort an array*

### 5.18.1 Detailed Description

Shell Sort Algorithm http://en.wikipedia.org/wiki/Shell_sort

Definition at line 13 of file Shell.cs.

### 5.18.2 Member Function Documentation

#### 5.18.2.1 static void eda12131190311906.Shell.Sort ( int[] *A* ) [inline], [static]

Sort an array

**Parameters**

| | |
|---|---|
| *A* | Array to sort |

Definition at line 19 of file Shell.cs.

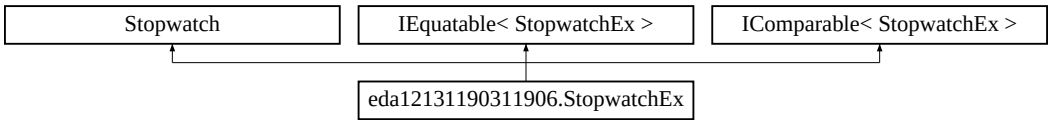
The documentation for this class was generated from the following file:

- Shell.cs

## 5.19 eda12131190311906.StopwatchEx Class Reference

Provides a set of methods and properties that you can use to accurately measure elapsed time. Extended Version

Inheritance diagram for eda12131190311906.StopwatchEx:




### Public Member Functions

- StopwatchEx ()

*Initializes a new instance of the T:eda12131190311906.StopwatchEx class.*

- StopwatchEx (TimeSpan editableElapsed)

  *Initializes a new instance of the T:eda12131190311906.StopwatchEx class.*
- bool Equals (StopwatchEx other)

  *Check if this object is equal to other*
- int CompareTo (StopwatchEx other)

  *Compare this object with other*

### Static Public Member Functions

- static new StopwatchEx StartNew ()

  *Initializes a new T:eda12131190311906.StopwatchEx instance, sets the elapsed time property to zero, and starts measuring elapsed time.*
- static StopwatchEx ComputeAverage (List< StopwatchEx > list, bool cutLowerHigher)

  *Compute average value from a list of values*

### Properties

- TimeSpan EditableElapsed `[get, set]`

  *Gets or sets the total elapsed time measured by the current instance.*
- new double ElapsedMilliseconds `[get]`

  *Gets the total elapsed time measured by the current instance, in milliseconds.*
- new long ElapsedTicks `[get]`

  *Gets the total elapsed time measured by the current instance, in timer ticks.*

## 5.19.1 Detailed Description

Provides a set of methods and properties that you can use to accurately measure elapsed time. Extended Version

Definition at line 19 of file StopwatchEx.cs.

## 5.19.2 Constructor & Destructor Documentation

### 5.19.2.1 eda12131190311906.StopwatchEx.StopwatchEx ( ) `[inline]`

Initializes a new instance of the T:eda12131190311906.StopwatchEx class.

Definition at line 59 of file StopwatchEx.cs.

### 5.19.2.2 eda12131190311906.StopwatchEx.StopwatchEx ( TimeSpan *editableElapsed* ) `[inline]`

Initializes a new instance of the T:eda12131190311906.StopwatchEx class.

**Parameters**

| | |
|---|---|
| *editableElapsed* | |

Definition at line 68 of file StopwatchEx.cs.

## 5.19.3 Member Function Documentation

### 5.19.3.1 int eda12131190311906.StopwatchEx.CompareTo ( StopwatchEx *other* ) `[inline]`

Compare this object with other

**Parameters**

| | |
|---|---|
| *other* | Other object |

**Returns**

-1 if is lower, 0 if equals, 1 if is higher

Definition at line 103 of file StopwatchEx.cs.

#### 5.19.3.2 static StopwatchEx eda12131190311906.StopwatchEx.ComputeAverage ( List< StopwatchEx > *list*, bool *cutLowerHigher* ) [inline],[static]

Compute average value from a list of values

**Parameters**

| | |
|---|---|
| *list* | List with all values |
| *cutLowerHigher* | If true the lower and higher values will be cut from calculations |

Definition at line 123 of file StopwatchEx.cs.

#### 5.19.3.3 bool eda12131190311906.StopwatchEx.Equals ( StopwatchEx *other* ) [inline]

Check if this object is equal to other

**Parameters**

| | |
|---|---|
| *other* | Other object |

**Returns**

True if is equal, otherwise false

Definition at line 93 of file StopwatchEx.cs.

#### 5.19.3.4 static new StopwatchEx eda12131190311906.StopwatchEx.StartNew ( ) [inline],[static]

Initializes a new T:eda12131190311906.StopwatchEx instance, sets the elapsed time property to zero, and starts measuring elapsed time.

**Returns**

A T:eda12131190311906.StopwatchEx that has just begun measuring elapsed time.

<filterpriority>1</filterpriority>

Definition at line 81 of file StopwatchEx.cs.

### 5.19.4 Property Documentation

#### 5.19.4.1 TimeSpan eda12131190311906.StopwatchEx.EditableElapsed [get],[set]

Gets or sets the total elapsed time measured by the current instance.

**Returns**

A read-only T:System.TimeSpan representing the total elapsed time measured by the current instance.

Definition at line 28 of file StopwatchEx.cs.

#### 5.19.4.2 new double eda12131190311906.StopwatchEx.ElapsedMilliseconds [get]

Gets the total elapsed time measured by the current instance, in milliseconds.

**Returns**

A read-only double representing the total number of milliseconds measured by the current instance.

<filterpriority>1</filterpriority>

Definition at line 39 of file StopwatchEx.cs.

#### 5.19.4.3 new long eda12131190311906.StopwatchEx.ElapsedTicks [get]

Gets the total elapsed time measured by the current instance, in timer ticks.

**Returns**

A read-only long integer representing the total number of timer ticks measured by the current instance.

<filterpriority>1</filterpriority>

Definition at line 52 of file StopwatchEx.cs.

The documentation for this class was generated from the following file:

- StopwatchEx.cs

## 5.20 eda12131190311906.SystemHelper Class Reference

System Helper Utilities

### Static Public Member Functions

- static string GetProgramFilesX86Path ()

  *Get program files X86 path, windows only*
- static bool IsWindows ()

  *Is windows OS*
- static bool IsUnix ()

  *Is UNIX OS*
- static int[] RandomIntegerArray (int size, int maxValue)

  *Generate a random integer array*
- static int[] RandomIntegerArray (int size, int minValue, int maxValue)

  *Generate a random integer array*
- static List< int[]> CloneListIntArray (List< int[]> list)

  *Clone an List int[]*
- static string ArrayToString< T > (T[] list)

  *Convert an array to string*
- static string ArrayToString< T > (T[] list, int limit)

  *Convert an array to string*
- static void OpenLink (string address)

  *Open website link*

### 5.20.1 Detailed Description

System Helper Utilities

Definition at line 21 of file SystemHelper.cs.

### 5.20.2 Member Function Documentation

#### 5.20.2.1 static string eda12131190311906.SystemHelper.ArrayToString< T >( T[] *list* ) [inline], [static]

Convert an array to string

**Template Parameters**

| | |
|---|---|
| *T* | |

**Parameters**

| | |
|---|---|
| *list* | |

**Returns**

Definition at line 105 of file SystemHelper.cs.

#### 5.20.2.2 static string eda12131190311906.SystemHelper.ArrayToString< T >( T[] *list*, int *limit* ) [inline], [static]

Convert an array to string

**Template Parameters**

| | |
|---|---|
| *T* | Array class |

**Parameters**

| | |
|---|---|
| *list* | List with arrays |
| *limit* | Elements limit to output to string |

**Returns**

A formated string

Definition at line 117 of file SystemHelper.cs.

#### 5.20.2.3 static List<int[]> eda12131190311906.SystemHelper.CloneListIntArray ( List< int[]> *list* ) [inline], [static]

Clone an List int[]

**Parameters**

| | |
|---|---|
| *list* | list List to clone |

**Returns**

Cloned list

Definition at line 94 of file SystemHelper.cs.

#### 5.20.2.4 static string eda12131190311906.SystemHelper.GetProgramFilesX86Path ( ) [inline], [static]

Get program files X86 path, windows only

**Returns**

Program files X86 path

Definition at line 26 of file SystemHelper.cs.

#### 5.20.2.5 static bool eda12131190311906.SystemHelper.IsUnix ( ) [inline], [static]

Is UNIX OS

**Returns**

True if Unix, otherwise false

Definition at line 53 of file SystemHelper.cs.

#### 5.20.2.6 static bool eda12131190311906.SystemHelper.IsWindows ( ) [inline], [static]

Is windows OS

**Returns**

True if Windows, otherwise false

Definition at line 44 of file SystemHelper.cs.

#### 5.20.2.7 static void eda12131190311906.SystemHelper.OpenLink ( string *address* ) [inline], [static]

Open website link

**Parameters**

| | |
|---|---|
| *address* | URL address |

Definition at line 140 of file SystemHelper.cs.

#### 5.20.2.8 static int [] eda12131190311906.SystemHelper.RandomIntegerArray ( int *size*, int *maxValue* ) [inline], [static]

Generate a random integer array

**Parameters**

| | |
|---|---|
| *size* | Size of array |
| *maxValue* | Max value for random numbers |

**Returns**

An array populated with random values

Definition at line 65 of file SystemHelper.cs.

#### 5.20.2.9 static int [] eda12131190311906.SystemHelper.RandomIntegerArray ( int *size*, int *minValue*, int *maxValue* ) [inline], [static]

Generate a random integer array

**Parameters**

| | |
|---|---|
| *size* | Size of array |
| *minValue* | Min value for random numbers |
| *maxValue* | Max value for random numbers |

**Returns**

An array populated with random values

Definition at line 77 of file SystemHelper.cs.

The documentation for this class was generated from the following file:

- SystemHelper.cs

# Index