

Zeus Utilities

Ver. 1.3

Manual

Copyright © 2019-2023 Χρήστος Μουρατίδης

Πίνακας περιεχομένων

ZEUS UTILITIES VERSION 1.3	5
ΟΙ ΟΡΙΣΜΟΙ ΤΩΝ ΚΛΑΣΕΩΝ	8
ΔΙΑΝΟΜΗ	10
ΕΠΙΚΟΙΝΩΝΙΑ	11
ΤΙ ΝΕΟ ΥΠΑΡΧΕΙ ΣΤΗΝ ΕΚΔΟΣΗ 1.1	12
ΤΙ ΝΕΟ ΥΠΑΡΧΕΙ ΣΤΗΝ ΕΚΔΟΣΗ 1.2	14
ΤΙ ΝΕΟ ΥΠΑΡΧΕΙ ΣΤΗΝ ΕΚΔΟΣΗ 1.3	15
ANIMATIONUTILITIES	16
ΑΠΑΡΙΘΜΗΣΕΙΣ	18
BANNER - ΜΕΘΟΔΟΙ	19
<i>Banner(element, offSetX, bannerSpeed)</i>	20
<i>Banner(element, offSetX, duration)</i>	22
<i>Banner(element, offSetX, beginTime, bannerSpeed)</i>	24
<i>Banner(element, offSetX, beginTime, duration)</i>	26
<i>StopBannerAnimation(element)</i>	28
FADEIN – ΜΕΘΟΔΟΙ	29
<i>FadeIn(element, animationSpeed)</i>	30
<i>FadeIn(element, duration)</i>	31
<i>FadeIn(element, beginTime, animationSpeed)</i>	32
<i>FadeIn(element, beginTime, duration)</i>	34
FADEINANDMOVEIN – ΜΕΘΟΔΟΙ	36
<i>FadeInAndMoveIn(element, offsetX, offsetY, animationSpeed)</i>	37
<i>FadeInAndMoveIn(element, offsetX, offsetY, duration)</i>	39
<i>FadeInAndMoveIn(element, offsetX, offsetY, beginTime, animationSpeed)</i>	41
<i>FadeInAndMoveIn(element, offsetX, offsetY, beginTime, duration)</i>	43
FADEINANDOUT – ΜΕΘΟΔΟΙ	45
<i>FadeInAndOut(element, animationSpeed, repeatTimes, Optional fillBehavior, Optional repeatForever, Optional reverse)</i>	47
<i>FadeInAndOut(element, duration, repeatTimes, Optional fillBehavior, Optional repeatForever, Optional reverse)</i>	49
<i>FadeInAndOut(element, beginTime, animationSpeed, repeatTimes, Optional fillBehavior, Optional repeatForever, Optional reverse)</i>	51
<i>FadeInAndOut(element, beginTime, duration, repeatTimes, Optional fillBehavior, Optional repeatForever, Optional reverse)</i>	54
<i>StopFadeInAndOutAnimation(element)</i>	57
FADEOUT – ΜΕΘΟΔΟΙ	58
<i>FadeOut(element, animationSpeed)</i>	59
<i>FadeOut(element, duration)</i>	60
<i>FadeOut(element, beginTime, animationSpeed)</i>	61
<i>FadeOut(element, beginTime, duration)</i>	63
FADEOUTANDMOVEOUT – ΜΕΘΟΔΟΙ	65
<i>FadeOutAndMoveOut(element, offsetX, offsetY, animationSpeed)</i>	66
<i>FadeOutAndMoveOut(element, offsetX, offsetY, duration)</i>	68
<i>FadeOutAndMoveOut(element, offsetX, offsetY, beginTime, animationSpeed)</i>	70
<i>FadeOutAndMoveOut(element, offsetX, offsetY, beginTime, duration)</i>	72
ZOOMIN – ΜΕΘΟΔΟΙ	74

<i>ZoomIn(element, zoomFactor, animationSpeed)</i>	76
<i>ZoomIn(element, zoomFactor, beginTime, animationSpeed)</i>	77
<i>ZoomIn(element, zoomFactor, pointOrigin, animationSpeed)</i>	79
<i>ZoomIn(element, zoomFactor, pointOrigin, beginTime, animationSpeed)</i>	81
<i>ZoomIn(element, zoomFactor, duration)</i>	83
<i>ZoomIn(element, zoomFactor, beginTime, duration)</i>	84
<i>ZoomIn(element, zoomFactor, pointOrigin, duration)</i>	86
<i>ZoomIn(element, zoomFactor, pointOrigin, beginTime, duration)</i>	88
ZoomInAndOut – ΜΕΘΟΔΟΙ	90
<i>ZoomInAndOut(element, zoomFactor, animationSpeed, repeatTimes, Optional repeatForever)</i>	93
<i>ZoomInAndOut(element, zoomFactor, beginTime, animationSpeed, repeatTimes, Optional repeatForever)</i>	95
<i>ZoomInAndOut(element, zoomFactor, pointOrigin, animationSpeed, repeatTimes, Optional repeatForever)</i>	97
<i>ZoomInAndOut(element, zoomFactor, pointOrigin, beginTime, animationSpeed, repeatTimes, Optional repeatForever)</i>	99
<i>ZoomInAndOut(element, zoomFactor, duration, repeatTimes, Optional repeatForever)</i>	101
<i>ZoomInAndOut(element, zoomFactor, beginTime, duration, repeatTimes, Optional repeatForever)</i>	103
<i>ZoomInAndOut(element, zoomFactor, pointOrigin, duration, repeatTimes, Optional repeatForever)</i>	105
<i>ZoomInAndOut(element, zoomFactor, pointOrigin, beginTime, duration, repeatTimes, Optional repeatForever)</i>	107
<i>StopZoomInAndOutAnimation(element)</i>	109
ZoomOut – ΜΕΘΟΔΟΙ	110
<i>ZoomOut(element, zoomFactor, animationSpeed)</i>	112
<i>ZoomOut(element, zoomFactor, beginTime, animationSpeed)</i>	113
<i>ZoomOut(element, zoomFactor, pointOrigin, animationSpeed)</i>	115
<i>ZoomOut(element, zoomFactor, pointOrigin, beginTime, animationSpeed)</i>	117
<i>ZoomOut(element, zoomFactor, duration)</i>	119
<i>ZoomOut(element, zoomFactor, beginTime, duration)</i>	120
<i>ZoomOut(element, zoomFactor, pointOrigin, duration)</i>	122
<i>ZoomOut(element, zoomFactor, pointOrigin, beginTime, duration)</i>	124
DATAUTILITIES	126
ΑΠΑΡΙΘΜΗΣΕΙΣ	127
GETDBVALUE – ΜΕΘΟΔΟΙ	128
<i>GetDBValueBoolean(value)</i>	131
<i>GetDBValueBooleanNullable(value)</i>	134
<i>GetDBValueByte(value)</i>	136
<i>GetDBValueByteNullable(value)</i>	138
<i>GetDBValueShort(value)</i>	140
<i>GetDBValueShortNullable(value)</i>	142
<i>GetDBValueInteger(value)</i>	144
<i>GetDBValueIntegerNullable(value)</i>	146
<i>GetDBValueLong(value)</i>	148
<i>GetDBValueLongNullable(value)</i>	150
<i>GetDBValueSingle(value)</i>	152
<i>GetDBValueSingleNullable(value)</i>	154
<i>GetDBValueDouble(value)</i>	156
<i>GetDBValueDoubleNullable(value)</i>	158
<i>GetDBValueDecimal(value)</i>	160
<i>GetDBValueDecimalNullable(value)</i>	162
<i>GetDBValueDateTime(value)</i>	164
<i>GetDBValueDateTimeNullable(value)</i>	166
<i>GetDBValueDate(value)</i>	168
<i>GetDBValueDateNullable(value)</i>	170

SETDBVALUE – ΜΕΘΟΔΟΙ.....	172
<i>SetDBValueBooleanNullable(value)</i>	174
<i>SetDBValueByteNullable(value)</i>	177
<i>SetDBValueShortNullable(value)</i>	178
<i>SetDBValueIntegerNullable(value)</i>	179
<i>SetDBValueLongrNullable(value)</i>	180
<i>SetDBValueSingleNullable(value)</i>	181
<i>SetDBValueDoubleNullable(value)</i>	182
<i>SetDBValueDecimalNullable(value)</i>	183
<i>SetDBValueDateTimeNullable(value)</i>	184
<i>SetDBValueDateNullable(value)</i>	185
ΕΛΕΓΧΟΣ ΣΥΝΔΕΣΗΣ ΣΕ ΒΑΣΗ – ΜΕΘΟΔΟΙ	186
<i>CheckOLEDBConnection(connectionString, pathDB)</i>	187
<i>CheckSqlServerConnection(connectionString, pathDB)</i>	189
FILEUTILITIES.....	191
FILE/FOLDER - ΜΕΘΟΔΟΙ	192
<i>GetFileExtension(filename)</i>	193
<i>RemoveFileAttributeReadOnly(filename)</i>	194
ASSEMBLIES – ΜΕΘΟΔΟΙ.....	195
<i>GetTitleAndFileVersionFromAssembly(whatType)</i>	196
<i>GetAssemblyVersionFromAssembly(whatType)</i>	198
<i>GetCompanyNameFromAssembly(whatType)</i>	200
FLOWDOCUMENTUTILITIES.....	202
SHARED ΙΔΙΟΤΗΤΕΣ	203
FLOWDOCUMENT – ΜΕΘΟΔΟΙ.....	204
<i>FlowDocumentCopy(fromFlowDocument, toFlowDocument)</i>	205
<i>FlowDocumentGetContent(flowDocument)</i>	211
<i>FlowDocumentGetPlainText(flowDocument)</i>	214
<i>FlowDocumentSetContent(flowDocument, xamlText)</i>	216
RICHTEXTBOX – ΜΕΘΟΔΟΙ	218
<i>RichTextBoxClearDocument(richTextBox)</i>	219
<i>RichTextBoxCopy(fromRichTextBox, toRichTextBox)</i>	226
<i>RichTextBoxIsEmpty(richTextBox)</i>	228
<i>RichTextBoxSetDocument(richTextBox, newDocument)</i>	229
<i>RichTextBoxSetDocument(richTextBox, flowDocumentXamlText)</i>	231
IMAGEUTILITIES.....	233
<i>GetColor(colorHexString)</i>	235
<i>GetImageBytes(filePath)</i>	236
<i>GetImageBytes(bitmapSource)</i>	237
<i>GetImageSource(filePath)</i>	239
<i>GetImageSource(componentAssembly, imagePathInComponent)</i>	241
<i>GetThumbnailImage(filePath, newHeight)</i>	243
<i>GetThumbnailImage(imgBytes, newHeight)</i>	245
MEASUREUTILITIES.....	247
<i>ConvertCentimetersToInches(centimeters)</i>	248
<i>ConvertCentimetersToPixels(centimeters, dpi)</i>	249
<i>ConvertInchesToCentimeters(inches)</i>	251
<i>ConvertPixelsToCentimeters(pixels, dpiScaleFactor)</i>	252
<i>GetSystemDpiY()</i>	254

<i>GetSystemDpiX()</i>	255
SECURITYUTILITIES	256
ΚΡΥΠΤΟΓΡΑΦΗΣΗ MD5	257
<i>GetMD5Hash(inputString)</i>	258
<i>GetMD5Hash(md5, inputString)</i>	259
<i>VerifyMD5Hash(inputString, hashString)</i>	261
<i>VerifyMD5Hash(md5, inputString, hashString)</i>	263
ΚΡΥΠΤΟΓΡΑΦΗΣΗ SHA-1	265
<i>GetSHA1Hash(inputString)</i>	266
<i>GetSHA1Hash(sha1, inputString)</i>	267
<i>VerifySHA1Hash(inputString, hashString)</i>	269
<i>VerifySHA1Hash(sha1, inputString, hashString)</i>	271
WINDOWUTILITIES	274
ΑΠΑΡΙΘΜΗΣΕΙΣ	275
MOUSE CURSOR	276
<i>SetNormalCursor()</i>	277
<i>SetWaitCursor()</i>	278
ΕΞΕΙΔΙΚΕΥΜΕΝΑ ΠΑΡΑΘΥΡΑ	279
<i>InputMessage</i>	280
<i>InputMessage(prompt, Optional leftPos, Optional topPos)</i>	284
<i>InputMessage(prompt, title, Optional leftPos, Optional topPos)</i>	286
<i>InputMessage(prompt, title, defaultResponse, Optional leftPos, Optional topPos)</i>	288
<i>InputMessage(prompt, title, defaultResponse, icon, Optional leftPos, Optional topPos)</i>	290
<i>InputMessage(prompt, title, defaultResponse, icon, cornerRadius, Optional leftPos, Optional topPos)</i>	292
<i>InputMessage(prompt, title, defaultResponse, icon, cornerRadius, width, height, Optional leftPos, Optional topPos)</i>	294
<i>InputMessage(prompt, title, defaultResponse, icon, cornerRadius, width, height, foreground, background, Optional leftPos, Optional topPos)</i>	297
<i>InputMessage(prompt, title, defaultResponse, icon, cornerRadius, width, height, foreground, background, titleStyle, promptStyle, inputTextStyle, Optional leftPos, Optional topPos)</i>	300
<i>InputMessage(prompt, title, defaultResponse, icon, cornerRadius, width, height, foreground, background, titleStyle, promptStyle, inputTextStyle, okButtonStyle, cancelButtonStyle, Optional leftPos, Optional topPos)</i>	304
<i>ShowMessage</i>	309
<i>ShowMessage(messageText)</i>	312
<i>ShowMessage(messageText, caption)</i>	313
<i>ShowMessage(messageText, caption, button)</i>	315
<i>ShowMessage(messageText, caption, button, icon)</i>	317
<i>ShowMessage(messageText, caption, button, icon, defaultResult)</i>	319
<i>ShowMessage(messageText, caption, button, icon, defaultResult, cornerRadius)</i>	321
<i>ShowMessage(messageText, caption, button, icon, defaultResult, cornerRadius, width, height)</i>	323
<i>ShowMessage(messageText, caption, button, icon, defaultResult, cornerRadius, width, height, foreground, background)</i>	325
<i>ShowMessage(messageText, caption, button, icon, defaultResult, cornerRadius, width, height, foreground, background, captionStyle, messageTextStyle)</i>	328
<i>ShowMessage(messageText, caption, button, icon, defaultResult, cornerRadius, width, height, foreground, background, captionStyle, messageTextStyle, okOrYesButtonStyle, cancelOrNoButtonStyle)</i>	332
<i>WaitWindow – Μέθοδοι</i>	336
<i>CloseWaitWindow(windowWait)</i>	337
<i>ShowWaitWindow(windowWait, message, Optional gradientBackColor1, Optional gradientBackColor2)</i>	338

Zeus Utilities version 1.3



Κλάσεις: AnimationUtilities, DataUtilities, FileUtilities, FlowDocumentUtilities, ImageUtilities, MeasureUtilities, SecurityUtilities, WindowUtilities.

Inherits: -

Namespace: Zeus.WPF.Classes.Utilities

Assembly: ZeusUtilities (in ZeusUtilities.dll)

Dependencies: -

Περιγραφή

Η βιβλιοθήκη **Zeus Utilities** περιλαμβάνει (στην έκδοση 1.3) **οκτώ κλάσεις Utilities**. Κάθε μία, περιλαμβάνει έναν αριθμό από στατικές μεθόδους χρήσιμες για την διεκπεραίωση συγκεκριμένων λειτουργιών, σχετικές με την κλάση που ανήκουν. Για παράδειγμα, η κλάση AnimationUtilities περιλαμβάνει την μέθοδο FadeIn που εφαρμόζει ένα fade-in effect σε ένα Framework element. Ή η κλάση SecurityUtilities περιλαμβάνει την μέθοδο GetMD5Hash που επιστρέφει ένα hash key για ένα συγκεκριμένο input string, κάτι που είναι εξαιρετικά χρήσιμο και σύνηθες όταν διαβάζουμε έναν κωδικό που δίνει ο χρήστης (π.χ. login code) και θέλουμε να εξετάσουμε αν είναι έγκυρος.

Την βιβλιοθήκη αυτή, την χρησιμοποιώ στις εφαρμογές μου και φυσικά θα επεκτείνεται στο μέλλον.

Παρακάτω, παρατίθενται οι **κλάσεις Utilities**.

	AnimationUtilities	<p>Περιλαμβάνει μία συλλογή από στατικές μεθόδους σχετικές με animation.</p> <p>Για παράδειγμα, η μέθοδος Banner εφαρμόζει ένα banner εφέ σε ένα TextBlock, κάνοντάς το να πηγαίνει συνεχώς από αριστερά προς τα δεξιά.</p>
--	---------------------------	--

<p>Ut</p>	<p>DataUtilities</p>	<p>Περιλαμβάνει μία συλλογή από στατικές μεθόδους σχετικές με διαχείριση δεδομένων, ιδιαιτέρως για το διάβασμα/γράψιμο από/σε database πεδία.</p> <p>Για παράδειγμα, η μέθοδος <code>GetDBValueBooleanNullable</code> λαμβάνει μία τιμή τύπου <code>Nullable(of Boolean)</code> από ένα πεδίο. Επιστρέφει, δηλαδή είτε <code>Nothing</code> (αν η τιμή του πεδίου είναι <code>DBNull.Value</code>) είτε την τιμή <code>Boolean</code>.</p>
<p>Ut</p>	<p>FileUtilities</p>	<p>Περιλαμβάνει μία συλλογή από στατικές μεθόδους σχετικές με διαχείριση αρχείων/φακέλων και assemblies.</p> <p>Για παράδειγμα, η μέθοδος <code>RemoveFileAttributeReadOnly</code> απομακρύνει την σημείωση "read-only" από τα attributes ενός αρχείου.</p>
<p>Ut</p>	<p>FlowDocumentUtilities</p>	<p>Περιλαμβάνει μία συλλογή από στατικές μεθόδους σχετικές με διαχείριση FlowDocuments και RichTextBoxes.</p> <p>Για παράδειγμα, η μέθοδος <code>RichTextBoxCopy</code> αντιγράφει το περιεχόμενο ενός <code>RichTextBox</code> σε άλλο <code>RichTextBox</code>. Ως γνωστόν, το περιεχόμενο ενός <code>RichTextBox</code> είναι ένα <code>FlowDocument</code> και δεν μπορούμε άμεσα να θέσουμε το περιεχόμενο ενός <code>RichTextBox</code> σε άλλο. Η μέθοδος αυτή πραγματοποιεί την λειτουργία της αντιγραφής.</p> <p>Τα <code>RichTextBoxes</code> είναι εξαιρετικά χρήσιμα ως πεδία σημειώσεων (notes) σε μία φόρμα δεδομένων, παρέχοντας στον χρήστη τη δυνατότητα να εισάγει και αποθηκεύει μορφοποιημένο κείμενο.</p>
<p>Ut</p>	<p>ImageUtilities</p>	<p>Περιλαμβάνει μία συλλογή από στατικές μεθόδους σχετικές με διαχείριση εικόνων (images).</p> <p>Για παράδειγμα, η μέθοδος <code>GetColor</code> λαμβάνει ένα Hex color string (π.χ. <code>#FFFFFFA500</code>) κι επιστρέφει ένα αντικείμενο <code>Nullable(of Color)</code>.</p>

Ut	MeasureUtilities	<p>Περιλαμβάνει μία συλλογή από στατικές μεθόδους σχετικές με διαχείριση και μετατροπή μετρήσεων (measures).</p> <p>Για παράδειγμα, η μέθοδος <code>ConvertCentimetersToPixels</code> μεταρρέπει cm σε pixels.</p>
Ut	SecurityUtilities	<p>Περιλαμβάνει μία συλλογή από στατικές μεθόδους σχετικές με ασφάλεια δεδομένων.</p> <p>Για παράδειγμα, η μέθοδος <code>GetMD5Hash</code> που επιστρέφει ένα hash key για ένα συγκεκριμένο input string, κάτι που είναι εξαιρετικά χρήσιμο και σύνηθες όταν διαβάζουμε έναν κωδικό που δίνει ο χρήστης (π.χ. login code) και θέλουμε να εξετάσουμε αν είναι έγκυρος.</p>
Ut	WindowUtilities	<p>Περιλαμβάνει μία συλλογή από στατικές μεθόδους σχετικές με διαχείριση παραθύρων (windows).</p> <p>Για παράδειγμα, η μέθοδος <code>ShowWaitWindow</code> εμφανίζει ένα παράθυρο με ένα ενημερωτικό μήνυμα (π.χ. "Περιμένετε παρακαλώ...") και παραμετρικό χρωματισμό gradient και μπορεί να χρησιμοποιηθεί σε χρονοβόρες διαδικασίες.</p> <p>-----</p> <p>NEO: Στην έκδοση 1.1 προστέθηκαν δύο ακόμα στατικές μέθοδοι:</p> <ul style="list-style-type: none"> - ShowMessage, η οποία μπορεί να αντικαταστήσει την παραδοσιακή Win32 <code>MessageBox.Show</code> με περισσότερες δυνατότητες που ταιριάζουν στη φιλοσοφία του WPF. - InputMessage, η οποία μπορεί να αντικαταστήσει την παραδοσιακή Win32 <code>InputBox</code> με περισσότερες δυνατότητες που ταιριάζουν στη φιλοσοφία του WPF.

Παρατηρήσεις

Περισσότερες μέθοδοι θα προστίθενται στη βιβλιοθήκη σε μελλοντικές εκδόσεις.

Οι ορισμοί των κλάσεων

Οι κλάσεις έχουν οριστεί ως εξής:

- Για την κλάση **AnimationUtilities**:

Σύνταξη:

VB:

```
Public Class AnimationUtilities
```

- Για την κλάση **DataUtilities**:

Σύνταξη:

VB:

```
Public Class DataUtilities
```

- Για την κλάση **FlowDocumentUtilities**:

Σύνταξη:

VB:

```
Public Class FlowDocumentUtilities
```

- Για την κλάση **ImageUtilities**:

Σύνταξη:

VB:

```
Public Class ImageUtilities
```

- Για την κλάση **MeasureUtilities**:

Σύνταξη:

VB:

```
Public Class MeasureUtilities
```

- Για την κλάση **SecurityUtilities**:

Σύνταξη:

VB:

```
Public Class SecurityUtilities
```

- Για την κλάση **WindowUtilities**:

Σύνταξη:

VB:

```
Public Class WindowUtilities
```

Διανομή

Κατά τη διανομή, στο φάκελο της εφαρμογής σας πρέπει να αντιγράψετε το **assembly αρχείο ZeusUtilities.dll**.

Επικοινωνία

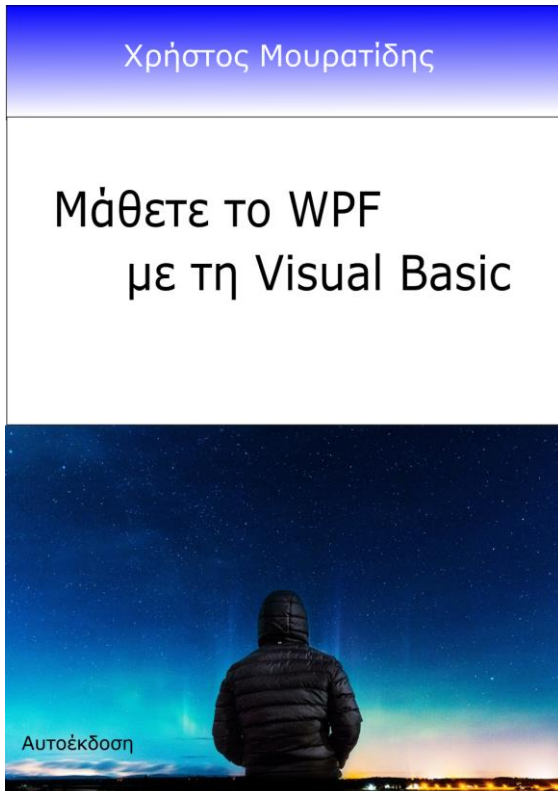
Για οποιαδήποτε πληροφορία ή διευκρίνιση παρακαλώ επικοινωνήστε στο:

mouratx@yahoo.com ή mouratx@hotmail.com



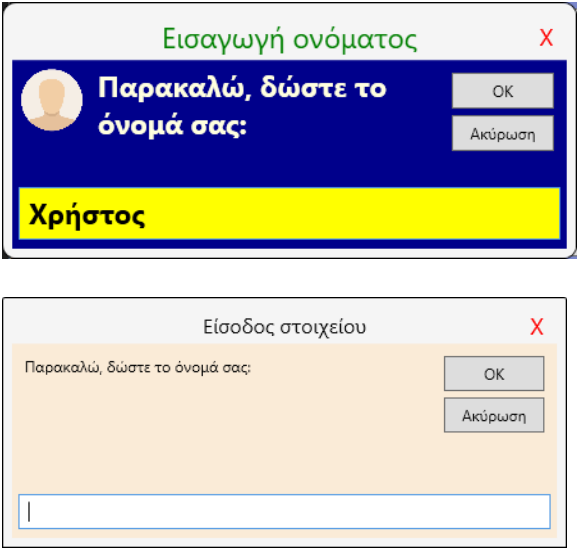
Χρήστος Μουρατίδης,
Πειραιάς, Οκτώβριος 2023, Οκτώβριος 2023

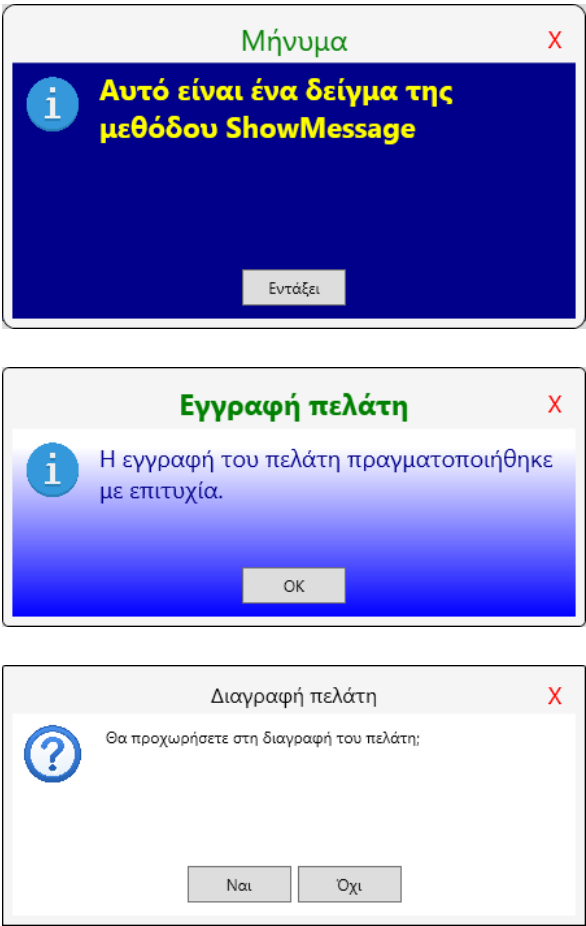
Υ.Γ. Μπορείτε να επικοινωνήσετε μαζί μου για να προμηθευτείτε το **βιβλίο** μου
"Μάθετε το WPF με τη Visual Basic" (1.333 σελίδες, Αυτοέκδοση 2018).



Τί νέο υπάρχει στην έκδοση 1.1

Στην έκδοση 1.1 έχουν προστεθεί 2 νέες στατικές μέθοδοι της κλάσης **WindowUtilities** :

InputMessage	<p>Μπορεί να χρησιμοποιηθεί αντί της κλασσικής InputBox με περισσότερες δυνατότητες που ταιριάζουν στη φιλοσοφία του WPF, όπως: καθορισμός χρωμάτων, στρογγυλότητας των γωνιών του παραθύρου, icon, styles για τον τίτλο, το μήνυμα prompt, το input textbox και τα buttons.</p>	
---------------------	--	--

ShowMessage	<p>Μπορεί να χρησιμοποιηθεί αντί της κλασσικής MessageBox.Show με περισσότερες δυνατότητες που ταιριάζουν στη φιλοσοφία του WPF, όπως: καθορισμός χρωμάτων, στρογγυλότητας των γωνιών του παραθύρου, icon, styles για τον τίτλο, το μήνυμα και τα buttons.</p>	
--------------------	--	---

Περισσότερα, θα δείτε στην ενότητα **WindowUtilities > Εξειδικευμένα παράθυρα**.

Τί νέο υπάρχει στην έκδοση 1.2

Δεν προστέθηκε κάτι νέο.

Έγιναν κάποιες τροποποιήσεις στις στατικές **μεθόδους GetDBValueXXXNullable** της κλάσης **DataUtilities**.

Τί νέο υπάρχει στην έκδοση 1.3

Στην **έκδοση 1.3** έχουν προστεθεί **3 νέες στατικές μέθοδοι** της κλάσης **ImageUtilities**:

GetImageSizeFromFile	Επιστρέφει την ανάλυση (μέγεθος) της εικόνας του συγκεκριμένου αρχείου εικόνας.
GetImageBytes_ (2 υπερφορτωμένες εκδόσεις)	Επιστρέφει έναν πίνακα bytes με τα image data ενός αρχείου εικόνας ή ενός BitmapImage source.
GetThumbnailImage_ (2 υπερφορτωμένες εκδόσεις)	Επιστρέφει μία BitmapImage που έχει ένα συγκεκριμένο ύψος (και αυτόματο width, λόγω της διατήρησης του aspect ratio) . Η πηγή είναι από ένα αρχείο εικόνας ή ένα image data byte array.

Περισσότερα, θα δείτε στην ενότητα **ImageUtilities**.



AnimationUtilities

Περιλαμβάνει μία συλλογή από στατικές μεθόδους σχετικές με **animation**.

Σύνταξη:

VB:

`Public Class AnimationUtilities`

Ο παρακάτω πίνακας περιέχει τις διαθέσιμες **κατηγορίες**:

Κατηγορία	Περιγραφή
Banner	Περιλαμβάνει στατικές μεθόδους για ένα banner εφέ . Πολύ χρήσιμο, αν επιθυμούμε να εμφανίζουμε συνεχώς ένα ενημερωτικό μήνυμα, σαν ταινία που κυλάει συνεχώς από αριστερά προς τα δεξιά.
FadeIn	Περιλαμβάνει στατικές μεθόδους για ένα fade-in εφέ . Χρήσιμο, αν επιθυμούμε να εμφανίσουμε σταδιακά ένα framework element.
FadeInAndMoveIn	Περιλαμβάνει στατικές μεθόδους για ένα fade-in and move-in εφέ . Χρήσιμο, αν επιθυμούμε να εμφανίσουμε σταδιακά ένα framework element, το οποίο ταυτόχρονα μετακινείται από μία offset θέση στην κανονική του θέση.

FadeInAndOut	<p>Περιλαμβάνει στατικές μεθόδους για ένα fade-in and out εφέ.</p> <p>Χρήσιμο, αν επιθυμούμε να "αναβοσβήνουμε" σταδιακά ένα framework element είτε για κάποιες επαναλήψεις είτε συνεχώς.</p>
FadeOut	<p>Περιλαμβάνει στατικές μεθόδους για ένα fade-out εφέ.</p> <p>Χρήσιμο, αν επιθυμούμε να εξαφανίσουμε σταδιακά ένα framework element.</p>
FadeOutAndMoveOut	<p>Περιλαμβάνει στατικές μεθόδους για ένα fade-out and move-out εφέ.</p> <p>Χρήσιμο, αν επιθυμούμε να εξαφανίσουμε σταδιακά ένα framework element, το οποίο ταυτόχρονα μετακινείται από την κανονική του θέση σε μία offset θέση.</p>
ZoomIn	<p>Περιλαμβάνει στατικές μεθόδους για ένα zoom-in εφέ.</p> <p>Χρήσιμο, αν επιθυμούμε να μεγενθύνουμε σταδιακά ένα framework element από το 0 προς ένα ποσοστό μεγέθυνσης</p>
ZoomInAndOut	<p>Περιλαμβάνει στατικές μεθόδους για ένα zoom-in and out εφέ.</p> <p>Χρήσιμο, αν επιθυμούμε να αυξομειώνουμε σταδιακά ένα framework element είτε για κάποιες επαναλήψεις είτε συνεχώς.</p>
ZoomOut	<p>Περιλαμβάνει στατικές μεθόδους για ένα zoom-out εφέ.</p> <p>Χρήσιμο, αν επιθυμούμε να σμικρύνουμε σταδιακά ένα framework element από ένα ποσοστό προς το 0.</p>

- **Να σημειώσουμε ότι σε όλες τις κατηγορίες υπάρχουν υπερφορτωμένες εκδόσεις για την καθυστέρηση έναρξης του animation (παράμετρος beginTime). Με αυτόν τον τρόπο μπορούμε να σχεδιάσουμε διαδοχικά animations των elements ενός παραθύρου, το ένα μετά το άλλο.**

Απαριθμήσεις

Όνομα	Περιγραφή
AnimationSpeedEnum	Περιλαμβάνει προκαθορισμένες τιμές Ταχύτητας του animation .
BannerSpeedEnum	Περιλαμβάνει προκαθορισμένες τιμές Ταχύτητας του banner animation .

Banner - Μέθοδοι

Όνομα	Περιγραφή
Banner (element As FrameworkElement, offSetX As Double, bannerSpeed As BannerSpeedEnum)	Εκτελεί το banner εφέ στο element , ξεκινώντας αριστερά από τη θέση offSetX και με ταχύτητα που προσδιορίζεται από την απαρίθμηση bannerSpeedEnum .
Banner (element As FrameworkElement, offSetX As Double, duration As TimeSpan)	Εκτελεί το banner εφέ στο element , ξεκινώντας αριστερά από τη θέση offSetX και σε χρόνο duration .
Banner (element As FrameworkElement, offSetX As Double, beginTime As TimeSpan, bannerSpeed As BannerSpeedEnum)	Εκτελεί το banner εφέ στο element , ξεκινώντας αριστερά από τη θέση offSetX και με ταχύτητα που προσδιορίζεται από την απαρίθμηση bannerSpeedEnum . Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime .
Banner (element As FrameworkElement, offSetX As Double, beginTime As TimeSpan, duration As TimeSpan)	Εκτελεί το banner εφέ στο element , ξεκινώντας αριστερά από τη θέση offSetX και σε χρόνο duration . Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime .
StopBannerAnimation (element As FrameworkElement)	Σταματάει το banner animation στο element .

Banner(*element*, *offsetX*, *bannerSpeed*)

Εκτελεί το banner εφέ στο **element**, ξεκινώντας αριστερά από τη θέση **offsetX** και με ταχύτητα που προσδιορίζεται από την απαρίθμηση **bannerSpeedEnum**.

Σύνταξη:

VB :

```
Public Shared Sub Banner(  
    element As FrameworkElement,  
    offsetX As Double,  
    bannerSpeed As BannerSpeedEnum  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

offset

Τύπος: System.Double
Η απόσταση από το αριστερό σημείο (x-point) του element, όπου θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή -100 σημαίνει 100 pixels αριστερά από την original θέση του element, λαμβανομένου υπόψη του left margin.

bannerSpeed

Τύπος: AnimationUtilities.BannerSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης BannerSpeedEnum. Για παράδειγμα, BannerSpeedEnum.VerySlow.

Παρατηρήσεις:

Για να δουλέψει επιτυχώς το εφέ ορίζουμε σαφώς την ιδιότητα Width του element.

Επίσης, από τη στιγμή που ξεκινήσει το banner animation ο μόνος τρόπος να σταματήσει είναι μέσω της μεθόδου StopBannerAnimation. Μπορούμε να δώσουμε τη δυνατότητα στο χρήστη να καλέσει τη μέθοδο StopBannerAnimation είτε μέσω κάποιου button είτε μέσω κάποιου πλήκτρου.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα TextBlock, με όνομα tbAuthor, μέσα σε ένα Border container το οποίο εκτείνεται εκτείνεται σε όλο το πλάτος του παραθύρου. Το εφέ θα εφαρμοστεί στο tbAuthor, με πολύ χαμηλή ταχύτητα:

XAML:

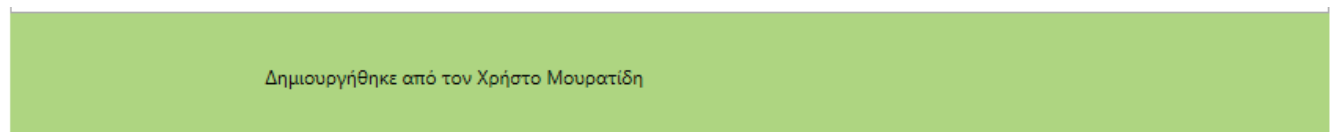
```
<Border Grid.Row="1" Background="#AED581" Padding="20" Height="80"
        VerticalAlignment="Bottom" >
    <TextBlock Name="tbAuthor"
        Text="Δημιουργήθηκε από τον Χρήστο Μουρατίδη" Width="250"
        VerticalAlignment="Center" HorizontalAlignment="Left" />
</Border>
```

VB:

```
Private Sub Window_Loaded(sender As Object, e As RoutedEventArgs)
    'Ξεκινάει το banner animation με τα στοιχεία μου.
    Banner(tbAuthor, -tbAuthor.ActualWidth, BannerSpeedEnum.VerySlow)
End Sub

'Όταν το παράθυρο αλλάζει να επανεκκινεί το banner animation.
Private Sub Window_SizeChanged(sender As Object, e As SizeChangedEventArgs)
    StopBannerAnimation(tbAuthor)
    Banner(tbAuthor, -tbAuthor.ActualWidth, BannerSpeedEnum.VerySlow)
End Sub
```

Το αποτέλεσμα, φαίνεται από τις εικόνες διαδοχικά παρακάτω:



Banner(element, offSetX, duration)

Εκτελεί το banner εφέ στο **element**, ξεκινώντας αριστερά από τη θέση **offSetX** και σε χρόνο **duration**.

Σύνταξη:

VB :

```
Public Shared Sub Banner(  
    element As FrameworkElement,  
    offSetX As Double,  
    duration As TimeSpan  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

offSet

Τύπος: System.Double
Η απόσταση από το αριστερό σημείο (x-point) του element, όπου θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή -100 σημαίνει 100 pixels αριστερά από την original θέση του element, λαμβανομένου υπόψη του left margin.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(12) σημαίνει 12 secs.

Παρατηρήσεις:

Για να δουλέψει επιτυχώς το εφέ ορίζουμε σαφώς την ιδιότητα Width του element.

Επίσης, από τη στιγμή που ξεκινήσει το banner animation ο μόνος τρόπος να σταματήσει είναι μέσω της μεθόδου StopBannerAnimation. Μπορούμε να δώσουμε τη δυνατότητα στο χρήστη να καλέσει τη μέθοδο StopBannerAnimation είτε μέσω κάποιου button είτε μέσω κάποιου πλήκτρου.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα TextBlock, με όνομα tbAuthor, μέσα σε ένα Border container το οποίο εκτείνεται εκτείνεται σε όλο το πλάτος του παραθύρου. Το εφέ θα εφαρμοστεί στο tbAuthor, με διάρκεια 10 secs:

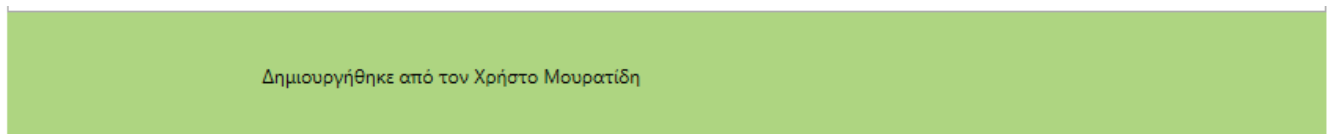
XAML:

```
<Border Grid.Row="1" Background="#AED581" Padding="20" Height="80" VerticalAlignment="Bottom" >  
    <TextBlock Name="tbAuthor" Text="Δημιουργήθηκε από τον Χρήστο Μουρατίδη" Width="250"  
        VerticalAlignment="Center" HorizontalAlignment="Left" />  
</Border>
```

VB:

```
Private Sub Window_Loaded(sender As Object, e As RoutedEventArgs)  
    'Ξεκινάει το banner animation με τα στοιχεία μου.  
    Banner(tbAuthor, -tbAuthor.ActualWidth, TimeSpan.FromSeconds(10))  
  
End Sub  
  
'Όταν το παράθυρο αλλάζει να επανεκκινεί το banner animation.  
Private Sub Window_SizeChanged(sender As Object, e As SizeChangedEventArgs)  
    StopBannerAnimation(tbAuthor)  
    Banner(tbAuthor, -tbAuthor.ActualWidth, TimeSpan.FromSeconds(10))  
  
End Sub
```

Το αποτέλεσμα, φαίνεται από τις εικόνες διαδοχικά παρακάτω:



Banner(element, offSetX, beginTime, bannerSpeed)

Εκτελεί το banner εφέ στο **element**, ξεκινώντας αριστερά από τη θέση **offSetX** και με ταχύτητα που προσδιορίζεται από την απαρίθμηση **bannerSpeedEnum**. Το εφέ θα ξεκινήσει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub Banner(  
    element As FrameworkElement,  
    offSetX As Double,  
    beginTime As TimeSpan,  
    bannerSpeed As BannerSpeedEnum  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

offSet

Τύπος: System.Double
Η απόσταση από το αριστερό σημείο (x-point) του element, όπου θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή -100 σημαίνει 100 pixels αριστερά από την original θέση του element, λαμβανομένου υπόψη του left margin.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

bannerSpeed

Τύπος: AnimationUtilities.BannerSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης BannerSpeedEnum. Για παράδειγμα, BannerSpeedEnum.VerySlow.

Παρατηρήσεις:

Για να δουλέψει επιτυχώς το εφέ ορίζουμε σαφώς την ιδιότητα Width του element.

Επίσης, από τη στιγμή που ξεκινήσει το banner animation ο μόνος τρόπος να σταματήσει είναι μέσω της μεθόδου StopBannerAnimation. Μπορούμε να δώσουμε τη δυνατότητα στο χρήστη να καλέσει τη μέθοδο StopBannerAnimation είτε μέσω κάποιου button είτε μέσω κάποιου πλήκτρου.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα TextBlock, με όνομα tbAuthor, μέσα σε ένα Border container το οποίο εκτείνεται εκτείνεται σε όλο το πλάτος του παραθύρου. Το εφέ θα εφαρμοστεί στο tbAuthor, με πολύ χαμηλή ταχύτητα και καθυστέρηση έναρξης 1.5 sec:

XAML:

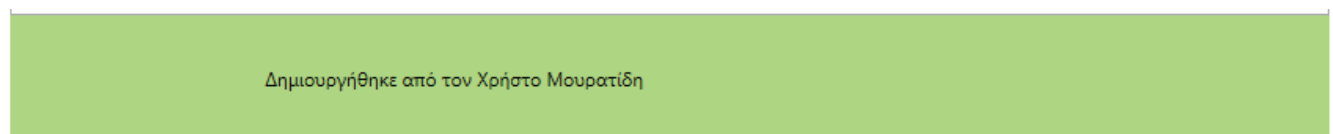
```
<Border Grid.Row="1" Background="#AED581" Padding="20" Height="80"
        VerticalAlignment="Bottom" >
    <TextBlock Name="tbAuthor"
        Text="Δημιουργήθηκε από τον Χρήστο Μουρατίδη" Width="250"
        VerticalAlignment="Center" HorizontalAlignment="Left" />
</Border>
```

VB:

```
Private Sub Window_Loaded(sender As Object, e As RoutedEventArgs)
    'Ξεκινάει το banner animation με τα στοιχεία μου.
    Banner(tbAuthor, -tbAuthor.ActualWidth, TimeSpan.FromSeconds(1.5), _
        BannerSpeedEnum.VerySlow)
End Sub

'Όταν το παράθυρο αλλάζει να επανεκκινεί το banner animation.
Private Sub Window_SizeChanged(sender As Object, e As SizeChangedEventArgs)
    StopBannerAnimation(tbAuthor)
    Banner(tbAuthor, -tbAuthor.ActualWidth, TimeSpan.FromSeconds(1.5), _
        BannerSpeedEnum.VerySlow)
End Sub
```

Το αποτέλεσμα, φαίνεται από τις εικόνες διαδοχικά παρακάτω:



Banner(element, offsetX, beginTime, duration)

Εκτελεί το banner εφέ στο **element**, ξεκινώντας αριστερά από τη θέση **offsetX** και σε χρόνο **duration**. Το εφέ θα ξεκινήσει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub Banner(  
    element As FrameworkElement,  
    offsetX As Double,  
    beginTime As TimeSpan,  
    duration As TimeSpan  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

offset

Τύπος: System.Double
Η απόσταση από το αριστερό σημείο (x-point) του element, όπου θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή -100 σημαίνει 100 pixels αριστερά από την original θέση του element, λαμβανομένου υπόψη του left margin.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(12) σημαίνει 12 secs.

Παρατηρήσεις:

Για να δουλέψει επιτυχώς το εφέ ορίζουμε σαφώς την ιδιότητα Width του element.

Επίσης, από τη στιγμή που ξεκινήσει το banner animation ο μόνος τρόπος να σταματήσει είναι μέσω της μεθόδου StopBannerAnimation. Μπορούμε να δώσουμε τη δυνατότητα στο χρήστη να καλέσει τη μέθοδο StopBannerAnimation είτε μέσω κάποιου button είτε μέσω κάποιου πλήκτρου.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα TextBlock, με όνομα tbAuthor, μέσα σε ένα Border container το οποίο εκτείνεται εκτείνεται σε όλο το πλάτος του παραθύρου. Το εφέ θα εφαρμοστεί στο tbAuthor, με διάρκεια 10 secs και καθυστέρηση έναρξης 1.5 sec:

XAML:

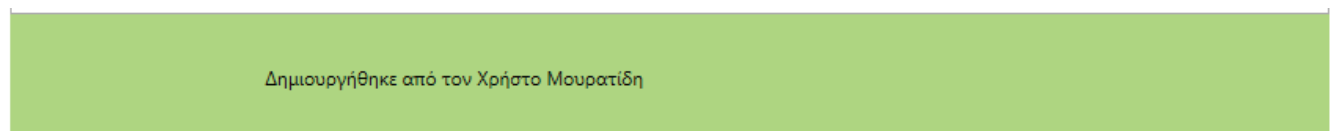
```
<Border Grid.Row="1" Background="#AED581" Padding="20" Height="80"
        VerticalAlignment="Bottom" >
    <TextBlock Name="tbAuthor"
        Text="Δημιουργήθηκε από τον Χρήστο Μουρατίδη" Width="250"
        VerticalAlignment="Center" HorizontalAlignment="Left" />
</Border>
```

VB:

```
Private Sub Window_Loaded(sender As Object, e As RoutedEventArgs)
    'Ξεκινάει το banner animation με τα στοιχεία μου.
    Banner(tbAuthor, -tbAuthor.ActualWidth, TimeSpan.FromSeconds(1.5), _
        TimeSpan.FromSeconds(10))
End Sub

'Όταν το παράθυρο αλλάζει να επανεκκινεί το banner animation.
Private Sub Window_SizeChanged(sender As Object, e As SizeChangedEventArgs)
    StopBannerAnimation(tbAuthor)
    Banner(tbAuthor, -tbAuthor.ActualWidth, TimeSpan.FromSeconds(1.5), _
        TimeSpan.FromSeconds(10))
End Sub
```

Το αποτέλεσμα, φαίνεται από τις εικόνες διαδοχικά παρακάτω:



StopBannerAnimation(element)

Σταματάει το banner animation στο **element**.

Σύνταξη:

VB :

```
Public Shared Sub StopBannerAnimation(  
    element As FrameworkElement  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement

Το element στο οποίο ήδη τρέχει το εφέ.

Παρατηρήσεις:

Δεν πρόκειται να προκληθεί σφάλμα αν καλέσουμε τη μέθοδο χωρίς να τρέχει σε αυτό το banner animation.

Παράδειγμα:

VB:

```
StopBannerAnimation(tbAuthor)
```

FadeIn - Μέθοδοι

Όνομα	Περιγραφή
<p>FadeIn(element As FrameworkElement, animationSpeed As AnimationSpeedEnum)</p>	<p>Εκτελεί το fade-in εφέ στο element με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum.</p>
<p>FadeIn(element As FrameworkElement, duration As TimeSpan)</p>	<p>Εκτελεί το fade-in εφέ στο element σε χρόνο duration.</p>
<p>FadeIn(element As FrameworkElement, beginTime As TimeSpan, animationSpeed As AnimationSpeedEnum)</p>	<p>Εκτελεί το fade-in εφέ στο element με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum.</p> <p>Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime.</p>
<p>FadeIn(element As FrameworkElement, beginTime As TimeSpan, Duration As TimeSpan)</p>	<p>Εκτελεί το fade-in εφέ στο element σε χρόνο duration.</p> <p>Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime.</p>

FadeIn(element, animationSpeed)

Εκτελεί το fade-in εφέ στο **element** με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**.

Σύνταξη:

VB :

```
Public Shared Sub FadeIn(  
    element As FrameworkElement,  
    animationSpeed As AnimationSpeedEnum  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα TextBlock, με όνομα tbLastNameData, συνδεδεμένο με την ιδιότητα LastName ενός source αντικειμένου Student, του οποίου το περιεχόμενο εμφανίζεται σταδιακά με ένα fade-in εφέ, μόλις ο χρήστης πατήσει το button μετακίνησης σε μία νέα εγγραφή μαθητή:

XAML:

```
<TextBlock Name="tbLastNameData" Text="{Binding LastName}" />
```

VB:

```
FadeIn(tbLastNameData, AnimationSpeedEnum.Normal)
```

FadeIn(element, duration)

Εκτελεί το fade-in εφέ στο **element** σε χρόνο **duration**.

Σύνταξη:

VB :

```
Public Shared Sub FadeIn(  
    element As FrameworkElement,  
    duration As TimeSpan  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα TextBlock, με όνομα tbLastNameData, συνδεδεμένο με την ιδιότητα LastName ενός source αντικειμένου Student, του οποίου το περιεχόμενο εμφανίζεται σταδιακά με ένα fade-in εφέ, μόλις ο χρήστης πατήσει το button μετακίνησης σε μία νέα εγγραφή μαθητή:

XAML:

```
<TextBlock Name="tbLastNameData" Text="{Binding LastName}" />
```

VB:

```
FadeIn(tbLastNameData, TimeSpan.FromSeconds(0.7))
```


FadeIn(element, beginTime, animationSpeed)

Εκτελεί το fade-in εφέ στο **element** με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**. Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub FadeIn(  
    element As FrameworkElement,  
    beginTime As TimeSpan,  
    animationSpeed As AnimationSpeedEnum  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει δύο TextBlocks με όνομα tbLastNameData και tbFirstNameData, αντίστοιχα. Το πρώτο είναι συνδεδεμένο με την ιδιότητα LastName και το δεύτερο με την ιδιότητα FirstName ενός source αντικειμένου Student. Τα περιεχόμενά τους εμφανίζονται διαδοχικά το ένα μετά το άλλο με ένα fade-in εφέ (με διαφορά 0.4 secs), μόλις ο χρήστης πατήσει το button μετακίνησης σε μία νέα εγγραφή μαθητή.

XAML:

```
<TextBlock Name="tbLastNameData" Text="{Binding LastName}" />  
<TextBlock Name="tbFirstNameData" Text="{Binding FirstName}" />
```

VB:

```
FadeIn(tbLastNameData, AnimationSpeedEnum.Normal)  
FadeIn(tbFirstNameData, TimeSpan.FromSeconds(0.4), AnimationSpeedEnum.Normal)
```

FadeIn(element, beginTime, duration)

Εκτελεί το fade-in εφέ στο **element** σε χρόνο **duration**. Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub FadeIn(  
    element As FrameworkElement,  
    beginTime As TimeSpan,  
    duration As TimeSpan  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει δύο TextBlocks με όνομα tbLastNameData και tbFirstName, αντίστοιχα. Το πρώτο είναι συνδεδεμένο με την ιδιότητα LastName και το δεύτερο με την ιδιότητα FirstName ενός source αντικειμένου Student. Τα περιεχόμενά τους εμφανίζονται διαδοχικά το ένα μετά το άλλο με ένα fade-in εφέ (με διαφορά 0.4 secs), μόλις ο χρήστης πατήσει το button μετακίνησης σε μία νέα εγγραφή μαθητή.

XAML:

```
<TextBlock Name="tbLastNameData" Text="{Binding LastName}" />  
<TextBlock Name="tbFirstNameData" Text="{Binding FirstName}" />
```

VB:

```
FadeIn(tbLastNameData, TimeSpan.FromSeconds(0.7))  
FadeIn(tbFirstNameData, TimeSpan.FromSeconds(0.4), TimeSpan.FromSeconds(0.7))
```

FadeInAndMoveIn – Μέθοδοι

Όνομα	Περιγραφή
FadeInAndMoveIn (element As FrameworkElement, offsetX As Double, offsetY As Double, animationSpeed As AnimationSpeedEnum)	Εκτελεί το fade-in και move-in εφέ στο element , ξεκινώντας από τη θέση (offsetX, offsetY), με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum .
FadeInAndMoveIn (element As FrameworkElement, offsetX As Double, offsetY As Double, duration As TimeSpan)	Εκτελεί το fade-in και move-in εφέ στο element , ξεκινώντας από τη θέση (offsetX, offsetY) και σε χρόνο duration .
FadeInAndMoveIn (element As FrameworkElement, offsetX As Double, offsetY As Double, beginTime As TimeSpan, animationSpeed As AnimationSpeedEnum)	Εκτελεί το fade-in και move-in εφέ στο element , ξεκινώντας από τη θέση (offsetX, offsetY), με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum . Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime .
FadeInAndMoveIn (element As FrameworkElement, offsetX As Double, offsetY As Double, beginTime As TimeSpan, duration As TimeSpan)	Εκτελεί το fade-in και move-in εφέ στο element , ξεκινώντας από τη θέση (offsetX, offsetY) και σε χρόνο duration . Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime .

FadeInAndMoveIn(element, offsetX, offsetY, animationSpeed)

Εκτελεί το fade-in και move-in εφέ στο **element**, ξεκινώντας από τη θέση (**offsetX**, **offsetY**), με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**.

Σύνταξη:

VB :

```
Public Shared Sub FadeInAndMoveIn(  
    element As FrameworkElement,  
    offsetX As Double,  
    offsetY As Double,  
    animationSpeed As AnimationSpeedEnum  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

offsetX

Τύπος: System.Double
Το σημείο x-point της θέσης (offsetX, offsetY), από το οποίο θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή -100 σημαίνει 100 pixels αριστερά (στον οριζόντιο άξονα), από την original θέση του element.

offsetY

Τύπος: System.Double
Το σημείο y-point της θέσης (offsetX, offsetY), από το οποίο θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή 50 σημαίνει 50 pixels παρακάτω (στον κατακόρυφο άξονα), από την original θέση του element.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα TextBlock, με όνομα tbLastName, του οποίου το περιεχόμενο εμφανίζεται και μετακινείται σταδιακά με ένα fade-in και move-in εφέ, μόλις ο χρήστης πατήσει το button μετακίνησης σε μία νέα εγγραφή μαθητή. Το εφέ, ξεκινάει από τη θέση (-100, 50), δηλαδή 100 pixels αριστερά και 50 pixels πιο κάτω από την original θέση του tbLastName.

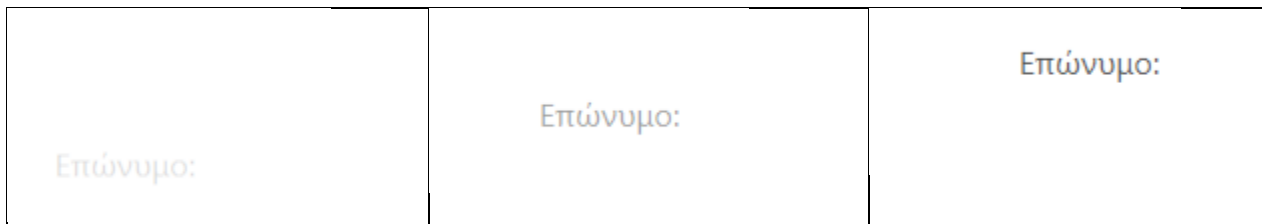
XAML:

```
<TextBlock Name="tbLastName" />
```

VB:

```
FadeInAndMoveIn(tbLastName, -100, 50, AnimationSpeedEnum.Normal)
```

Το αποτέλεσμα, φαίνεται από τις εικόνες διαδοχικά παρακάτω:



FadeInAndMoveIn(element, offsetX, offsetY, duration)

Εκτελεί το fade-in και move-in εφέ στο **element**, ξεκινώντας από τη θέση (**offsetX, offsetY**) και σε χρόνο **duration**.

Σύνταξη:

VB :

```
Public Shared Sub FadeInAndMoveIn(  
    element As FrameworkElement,  
    offsetX As Double,  
    offsetY As Double,  
    duration As TimeSpan  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

offsetX

Τύπος: System.Double
Το σημείο x-point της θέσης (offsetX, offsetY), από το οποίο θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή -100 σημαίνει 100 pixels αριστερά (στον οριζόντιο άξονα), από την original θέση του element.

offsetY

Τύπος: System.Double
Το σημείο y-point της θέσης (offsetX, offsetY), από το οποίο θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή 50 σημαίνει 50 pixels παρακάτω (στον κατακόρυφο άξονα), από την original θέση του element.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα TextBlock, με όνομα tbLastName, του οποίου το περιεχόμενο εμφανίζεται και μετακινείται σταδιακά με ένα fade-in και move-in εφέ, μόλις ο χρήστης πατήσει το button μετακίνησης σε μία νέα εγγραφή μαθητή. Το εφέ, ξεκινάει από τη θέση (-100, 50), δηλαδή 100 pixels αριστερά και 50 pixels πιο κάτω από την original θέση του tbLastName.

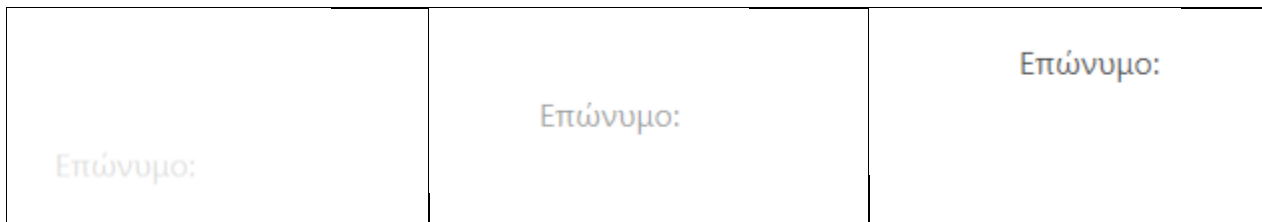
XAML:

```
<TextBlock Name="tbLastName" />
```

VB:

```
FadeInAndMoveIn(tbLastName, -100, 50, TimeSpan.FromSeconds(0.7))
```

Το αποτέλεσμα, φαίνεται από τις εικόνες διαδοχικά παρακάτω:



FadeInAndMoveIn(element, offsetX, offsetY, beginTime, animationSpeed)

Εκτελεί το fade-in και move-in εφέ στο **element**, ξεκινώντας από τη θέση (**offsetX**, **offsetY**), με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**. Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub FadeInAndMoveIn(  
    element As FrameworkElement,  
    offsetX As Double,  
    offsetY As Double,  
    beginTime As TimeSpan,  
    animationSpeed As AnimationSpeedEnum  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

offsetX

Τύπος: System.Double
Το σημείο x-point της θέσης (offsetX, offsetY), από το οποίο θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή -100 σημαίνει 100 pixels αριστερά (στον οριζόντιο άξονα), από την original θέση του element.

offsetY

Τύπος: System.Double
Το σημείο y-point της θέσης (offsetX, offsetY), από το οποίο θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή 50 σημαίνει 50 pixels παρακάτω (στον κατακόρυφο άξονα), από την original θέση του element.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα `TextBlock`, με όνομα `tbLastName`, του οποίου το περιεχόμενο εμφανίζεται και μετακινείται σταδιακά με ένα `fade-in` και `move-in` εφέ, μόλις ο χρήστης πατήσει το `button` μετακίνησης σε μία νέα εγγραφή μαθητή. Το εφέ, ξεκινάει από τη θέση `(-100, 50)`, δηλαδή 100 pixels αριστερά και 50 pixels πιο κάτω από την `original` θέση του `tbLastName` και με χρονική καθυστέρηση έναρξης 1.5 secs.

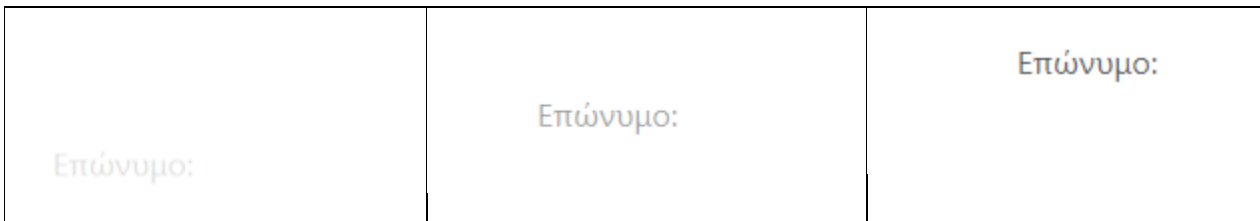
XAML:

```
<TextBlock Name="tbLastName" />
```

VB:

```
FadeInAndMoveIn(tbLastName, -100,  
50, TimeSpan.FromSeconds(1.5), AnimationSpeedEnum.Normal)
```

Το αποτέλεσμα, φαίνεται από τις εικόνες διαδοχικά παρακάτω:



FadeInAndMoveIn(element, offsetX, offsetY, beginTime, duration)

Εκτελεί το fade-in και move-in εφέ στο **element**, ξεκινώντας από τη θέση (**offsetX**, **offsetY**) και σε χρόνο **duration**. Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub FadeInAndMoveIn(
    element As FrameworkElement,
    offsetX As Double,
    offsetY As Double,
    beginTime As TimeSpan,
    duration As TimeSpan
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

offsetX

Τύπος: System.Double
Το σημείο x-point της θέσης (offsetX, offsetY), από το οποίο θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή -100 σημαίνει 100 pixels αριστερά (στον οριζόντιο άξονα), από την original θέση του element.

offsetY

Τύπος: System.Double
Το σημείο y-point της θέσης (offsetX, offsetY), από το οποίο θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή 50 σημαίνει 50 pixels παρακάτω (στον κατακόρυφο άξονα), από την original θέση του element.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα TextBlock, με όνομα tbLastName, του οποίου το περιεχόμενο εμφανίζεται και μετακινείται σταδιακά με ένα fade-in και move-in

εφέ, μόλις ο χρήστης πατήσει το button μετακίνησης σε μία νέα εγγραφή μαθητή. Το εφέ, ξεκινάει από τη θέση (-100, 50), δηλαδή 100 pixels αριστερά και 50 pixels πιο κάτω από την original θέση του tbLastName και με χρονική καθυστέρηση έναρξης 1.5 secs.

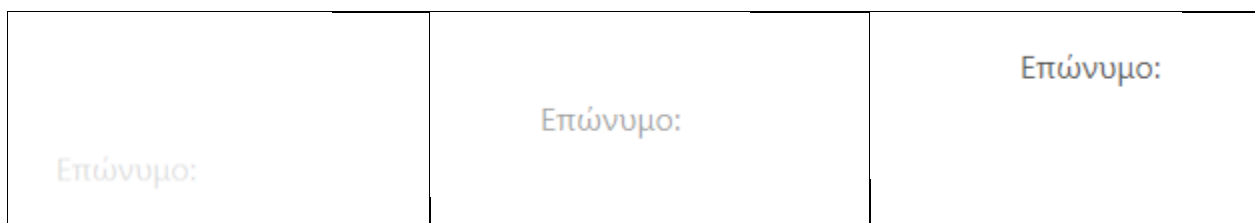
XAML:

```
<TextBlock Name="tbLastName" />
```

VB:

```
FadeInAndMoveIn(tbLastName, -100,
50, TimeSpan.FromSeconds(1.5), TimeSpan.FromSeconds(0.7))
```

Το αποτέλεσμα, φαίνεται από τις εικόνες διαδοχικά παρακάτω:



FadeInAndOut – Μέθοδοι

Όνομα	Περιγραφή
<p>FadeInAndOut(element As FrameworkElement, animationSpeed As AnimationSpeedEnum, repeatTimes As Integer, Optional fillBehavior As FillBehavior = FillBehavior.HoldEnd, Optional repeatForever As Boolean = False, Optional reverse As Boolean = False)</p>	<p>Εκτελεί το fade-in and out εφέ (αίσθηση να "αναβοσβήνει") στο element με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum.</p> <p>Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την repeatTimes εκτός και αν θέσουμε σαφώς την repeatForever, οπότε θα επαναλαμβάνεται συνεχώς.</p> <p>Η fillBehavior καθορίζει αν θα κρατήσει την τελική τιμή στο τέλος του εφέ και η reverse ξεκινάει το εφέ αντίστροφα (out->in).</p>
<p>FadeInAndOut(element As FrameworkElement, duration As TimeSpan, repeatTimes As Integer, Optional fillBehavior As FillBehavior = FillBehavior.HoldEnd, Optional repeatForever As Boolean = False, Optional reverse As Boolean = False)</p>	<p>Εκτελεί το fade-in and out εφέ (αίσθηση να "αναβοσβήνει") στο element σε χρόνο duration.</p> <p>Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την repeatTimes εκτός και αν θέσουμε σαφώς την repeatForever, οπότε θα επαναλαμβάνεται συνεχώς.</p> <p>Η fillBehavior καθορίζει αν θα κρατήσει την τελική τιμή στο τέλος του εφέ και η reverse ξεκινάει το εφέ αντίστροφα (out->in).</p>

<p>FadeInAndOut(element As FrameworkElement, beginTime As TimeSpan, animationSpeed As AnimationSpeedEnum, repeatTimes As Integer, Optional fillBehavior As FillBehavior = FillBehavior.HoldEnd, Optional repeatForever As Boolean = False, Optional reverse As Boolean = False)</p>	<p>Εκτελεί το fade-in and out εφέ (αίσθηση να "αναβοσβήνει") στο element με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum.</p> <p>Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την repeatTimes εκτός και αν θέσουμε σαφώς την repeatForever, οπότε θα επαναλαμβάνεται συνεχώς.</p> <p>Η fillBehavior καθορίζει αν θα κρατήσει την τελική τιμή στο τέλος του εφέ και η reverse ξεκινάει το εφέ αντίστροφα (out->in).</p> <p>Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime.</p>
<p>FadeInAndOut(element As FrameworkElement, beginTime As TimeSpan, duration As TimeSpan, repeatTimes As Integer, Optional fillBehavior As FillBehavior = FillBehavior.HoldEnd, Optional repeatForever As Boolean = False, Optional reverse As Boolean = False)</p>	<p>Εκτελεί το fade-in and out εφέ (αίσθηση να "αναβοσβήνει") στο element σε χρόνο duration.</p> <p>Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την repeatTimes εκτός και αν θέσουμε σαφώς την repeatForever, οπότε θα επαναλαμβάνεται συνεχώς.</p> <p>Η fillBehavior καθορίζει αν θα κρατήσει την τελική τιμή στο τέλος του εφέ και η reverse ξεκινάει το εφέ αντίστροφα (out->in).</p> <p>Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime.</p>
<p>StopFadeInAndOutAnimation(element As FrameworkElement)</p>	<p>Σταματάει το fade-in and out animation στο element.</p>

FadeInAndOut(element, animationSpeed, repeatTimes, Optional fillBehavior, Optional repeatForever, Optional reverse)

Εκτελεί το fade-in and out εφέ (αίσθηση να "αναβοσβήνει") στο **element** με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**. Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την **repeatTimes** εκτός και αν θέσουμε σαφώς την **repeatForever**, οπότε θα επαναλαμβάνεται συνεχώς. Η **fillBehavior** καθορίζει αν θα κρατήσει την τελική τιμή στο τέλος του εφέ και η **reverse** ξεκινάει το εφέ αντίστροφα (out->in).

Σύνταξη:

VB :

```
Public Shared Sub FadeInAndOut(  
    element As FrameworkElement,  
    animationSpeed As AnimationSpeedEnum,  
    repeatTimes As Integer,  
    Optional fillBehavior As FillBehavior,  
    Optional repeatForever As Boolean,  
    Optional reverse As Boolean  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

repeatTimes

Τύπος: System.Integer
Καθορίζει πόσες φορές θα επαναλαμβάνεται το εφέ (δηλαδή, πόσες φορές θα "αναβοσβήνει"). Για παράδειγμα, η τιμή 4 σημαίνει 4 φορές θα "αναβοσβήνει". Τεχνικά μιλώντας, πόσες φορές θα κάνει τον κύκλο in->out->in.

Optional fillBehavior

Τύπος: System.Windows.Media.Animation.FillBehavior
Καθορίζει αν θα κρατήσει την τελική τιμή της ιδιότητας που γίνεται animation (στην προκειμένη περίπτωση η Opacity). Η default τιμή είναι FillBehavior.HoldEnd

Optional repeatForever

Τύπος: System.Boolean
Αν είναι True, τότε το εφέ θα εκτελείται συνεχώς. Σε αυτήν την περίπτωση αγνοείται η τιμή της παραμέτρου repeatTimes. Η default τιμή είναι False.

Ο μόνος τρόπος να σταματήσει είναι να καλέσουμε τη μέθοδο `StopFadeInAndOutAnimation`.

Optional reverse

Τύπος: `System.Boolean`

Αν είναι `True`, τότε το εφέ θα εκτελείται ανάποδα, δηλαδή ο κύκλος θα είναι `out->in->out`.

Παράδειγμα 1:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα `Image` element, με όνομα `iconStudent`. Στον VB κώδικα, κάνουμε την εικόνα να "αναβοσβήσει" 4 φορές και μόλις ολοκληρωθεί το εφέ να επιστρέψει στην `original` τιμή (`Opacity=1`), καθορίζοντας `FillBehavior.Stop`. Υποθέτουμε ότι το αρχείο εικόνας `Student64x64.png` βρίσκεται στον υποφάκελο `Images` του project μας.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"
        Width="64" Height="64"/>
```

VB:

```
FadeInAndOut(iconStudent, AnimationSpeedEnum.Fast, 4, FillBehavior.Stop)
```

Παράδειγμα 2:

Στο ίδιο παράδειγμα με το παραπάνω, αλλά κάνοντας το εφέ να επαναλαμβάνεται συνεχώς:

VB:

```
FadeInAndOut(iconStudent, AnimationSpeedEnum.Fast, 0, repeatForever:=True)
```

Παράδειγμα 3:

Στο ίδιο παράδειγμα με το παραπάνω, αλλά κάνοντας κάνουμε την εικόνα να "αναβοσβήσει" 4 φορές ανάποδα (`out->in->out`):

VB:

```
FadeInAndOut(iconStudent, AnimationSpeedEnum.Fast, 4, reverse:=True)
```

FadeInAndOut(element, duration, repeatTimes, Optional fillBehavior, Optional repeatForever, Optional reverse)

Εκτελεί το fade-in and out εφέ (αίσθηση να "αναβοσβήνει") στο **element** σε χρόνο **duration**. Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την **repeatTimes** εκτός και αν θέσουμε σαφώς την **repeatForever**, οπότε θα επαναλαμβάνεται συνεχώς. Η **fillBehavior** καθορίζει αν θα κρατήσει την τελική τιμή στο τέλος του εφέ και η **reverse** ξεκινάει το εφέ αντίστροφα (out->in).

Σύνταξη:

VB :

```
Public Shared Sub FadeInAndOut(  
    element As FrameworkElement,  
    duration As TimeSpan,  
    repeatTimes As Integer,  
    Optional fillBehavior As FillBehavior,  
    Optional repeatForever As Boolean,  
    Optional reverse As Boolean  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

repeatTimes

Τύπος: System.Integer
Καθορίζει πόσες φορές θα επαναλαμβάνεται το εφέ (δηλαδή, πόσες φορές θα "αναβοσβήνει"). Για παράδειγμα, η τιμή 4 σημαίνει 4 φορές θα "αναβοσβήνει". Τεχνικά μιλώντας, πόσες φορές θα κάνει τον κύκλο in->out->in.

Optional fillBehavior

Τύπος: System.Windows.Media.Animation.FillBehavior
Καθορίζει αν θα κρατήσει την τελική τιμή της ιδιότητας που γίνεται animation (στην προκειμένη περίπτωση η Opacity). Η default τιμή είναι FillBehavior.HoldEnd

Optional repeatForever

Τύπος: System.Boolean
Αν είναι True, τότε το εφέ θα εκτελείται συνεχώς. Σε αυτήν την περίπτωση αγνοείται η τιμή της παραμέτρου repeatTimes. Η default τιμή είναι False.
Ο μόνος τρόπος να σταματήσει είναι να καλέσουμε τη μέθοδο StopFadeInAndOutAnimation.

Optional reverse

Τύπος: System.Boolean

Αν είναι True, τότε το εφέ θα εκτελείται ανάποδα, δηλαδή ο κύκλος θα είναι out->in->out .

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να "αναβοσβήσει" 4 φορές και μόλις ολοκληρωθεί το εφέ να επιστρέψει στην original τιμή (Opacity=1), καθορίζοντας FillBehavior.Stop. Υποθέτουμε ότι το αρχείο εικόνας Student64x64.png βρίσκεται στον υποφάκελο Images του project μας.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"
        Width="64" Height="64"/>
```

VB:

```
FadeInAndOut(iconStudent, TimeSpan.FromSeconds(0.7), 4, FillBehavior.Stop)
```

Παράδειγμα 2:

Στο ίδιο παράδειγμα με το παραπάνω, αλλά κάνοντας το εφέ να επαναλαμβάνεται συνεχώς:

VB:

```
FadeInAndOut(iconStudent, TimeSpan.FromSeconds(0.7), 0, repeatForever:=True)
```

Παράδειγμα 3:

Στο ίδιο παράδειγμα με το παραπάνω, αλλά κάνοντας κάνουμε την εικόνα να "αναβοσβήσει" 4 φορές ανάποδα (out->in->out):

VB:

```
FadeInAndOut(iconStudent, TimeSpan.FromSeconds(0.7), 4, reverse:=True)
```

FadeInAndOut(element, beginTime, animationSpeed, repeatTimes, Optional fillBehavior, Optional repeatForever, Optional reverse)

Εκτελεί το fade-in and out εφέ (αίσθηση να "αναβοσβήνει") στο **element** με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**. Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την **repeatTimes** εκτός και αν θέσουμε σαφώς την **repeatForever**, οπότε θα επαναλαμβάνεται συνεχώς. Η **fillBehavior** καθορίζει αν θα κρατήσει την τελική τιμή στο τέλος του εφέ και η **reverse** ξεκινάει το εφέ αντίστροφα (out->in). Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub FadeInAndOut(
    element As FrameworkElement,
    beginTime As TimeSpan,
    animationSpeed As AnimationSpeedEnum,
    repeatTimes As Integer,
    Optional fillBehavior As FillBehavior,
    Optional repeatForever As Boolean,
    Optional reverse As Boolean
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

repeatTimes

Τύπος: System.Integer
Καθορίζει πόσες φορές θα επαναλαμβάνεται το εφέ (δηλαδή, πόσες φορές θα "αναβοσβήνει"). Για παράδειγμα, η τιμή 4 σημαίνει 4 φορές θα "αναβοσβήνει". Τεχνικά μιλώντας, πόσες φορές θα κάνει τον κύκλο in->out->in.

Optional fillBehavior

Τύπος: System.Windows.Media.Animation.FillBehavior

Καθορίζει αν θα κρατήσει την τελική τιμή της ιδιότητας που γίνεται animation (στην προκειμένη περίπτωση η Opacity). Η default τιμή είναι FillBehavior.HoldEnd

Optional repeatForever

Τύπος: System.Boolean

Αν είναι True, τότε το εφέ θα εκτελείται συνεχώς. Σε αυτήν την περίπτωση αγνοείται η τιμή της παραμέτρου repeatTimes. Η default τιμή είναι False.

Ο μόνος τρόπος να σταματήσει είναι να καλέσουμε τη μέθοδο StopFadeInAndOutAnimation.

Optional reverse

Τύπος: System.Boolean

Αν είναι True, τότε το εφέ θα εκτελείται ανάποδα, δηλαδή ο κύκλος θα είναι out->in->out .

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να "αναβοσβήσει" 4 φορές και μόλις ολοκληρωθεί το εφέ να επιστρέψει στην original τιμή (Opacity=1), καθορίζοντας FillBehavior.Stop. Το εφέ ξεκινάει με χρονοκαθυστέρηση 1.5 secs.

Υποθέτουμε ότι το αρχείο εικόνας Student64x64.png βρίσκεται στον υποφάκελο Images του project μας.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"
        Width="64" Height="64"/>
```

VB:

```
FadeInAndOut(iconStudent, TimeSpan.FromSeconds(1.5), _
              AnimationSpeedEnum.Fast, 4, FillBehavior.Stop)
```

Παράδειγμα 2:

Στο ίδιο παράδειγμα με το παραπάνω, αλλά κάνοντας το εφέ να επαναλαμβάνεται συνεχώς:

VB:

```
FadeInAndOut(iconStudent, TimeSpan.FromSeconds(1.5), _
              AnimationSpeedEnum.Fast, 0, repeatForever:=True)
```

Παράδειγμα 3:

Στο ίδιο παράδειγμα με το παραπάνω, αλλά κάνοντας κάνουμε την εικόνα να "αναβοσβήσει" 4 φορές ανάποδα (out->in->out):

VB:

```
FadeInAndOut(iconStudent, TimeSpan.FromSeconds(1.5), _  
              AnimationSpeedEnum.Fast, 4, reverse:=True)
```

FadeInAndOut(element, beginTime, duration, repeatTimes, Optional fillBehavior, Optional repeatForever, Optional reverse)

Εκτελεί το fade-in and out εφέ (αίσθηση να "αναβοσβήνει") στο **element** σε χρόνο **duration**. Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την **repeatTimes** εκτός και αν θέσουμε σαφώς την **repeatForever**, οπότε θα επαναλαμβάνεται συνεχώς. Η **fillBehavior** καθορίζει αν θα κρατήσει την τελική τιμή στο τέλος του εφέ και η **reverse** ξεκινάει το εφέ αντίστροφα (out->in).

Σύνταξη:

VB :

```
Public Shared Sub FadeInAndOut(
    element As FrameworkElement,
    beginTime As TimeSpan,
    duration As TimeSpan,
    repeatTimes As Integer,
    Optional fillBehavior As FillBehavior,
    Optional repeatForever As Boolean,
    Optional reverse As Boolean
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

repeatTimes

Τύπος: System.Integer
Καθορίζει πόσες φορές θα επαναλαμβάνεται το εφέ (δηλαδή, πόσες φορές θα "αναβοσβήνει"). Για παράδειγμα, η τιμή 4 σημαίνει 4 φορές θα "αναβοσβήνει". Τεχνικά μιλώντας, πόσες φορές θα κάνει τον κύκλο in->out->in.

Optional fillBehavior

Τύπος: System.Windows.Media.Animation.FillBehavior
Καθορίζει αν θα κρατήσει την τελική τιμή της ιδιότητας που γίνεται animation (στην προκειμένη περίπτωση η Opacity). Η default τιμή είναι FillBehavior.HoldEnd

Optional repeatForever

Τύπος: System.Boolean

Αν είναι True, τότε το εφέ θα εκτελείται συνεχώς. Σε αυτήν την περίπτωση αγνοείται η τιμή της παραμέτρου repeatTimes. Η default τιμή είναι False.

Ο μόνος τρόπος να σταματήσει είναι να καλέσουμε τη μέθοδο StopFadeInAndOutAnimation.

Optional reverse

Τύπος: System.Boolean

Αν είναι True, τότε το εφέ θα εκτελείται ανάποδα, δηλαδή ο κύκλος θα είναι out->in->out .

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να "αναβοσβήσει" 4 φορές και μόλις ολοκληρωθεί το εφέ να επιστρέψει στην original τιμή (Opacity=1), καθορίζοντας FillBehavior.Stop. Το εφέ ξεκινάει με χρονοκαθυστέρηση 1.5 secs.

Υποθέτουμε ότι το αρχείο εικόνας Student64x64.png βρίσκεται στον υποφάκελο Images του project μας.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"
        Width="64" Height="64"/>
```

VB:

```
FadeInAndOut(iconStudent, TimeSpan.FromSeconds(1.5), _
              TimeSpan.FromSeconds(0.7), 4, FillBehavior.Stop)
```

Παράδειγμα 2:

Στο ίδιο παράδειγμα με το παραπάνω, αλλά κάνοντας το εφέ να επαναλαμβάνεται συνεχώς:

VB:

```
FadeInAndOut(iconStudent, TimeSpan.FromSeconds(1.5), _
              TimeSpan.FromSeconds(0.7), 0, repeatForever:=True)
```

Παράδειγμα 3:

Στο ίδιο παράδειγμα με το παραπάνω, αλλά κάνοντας κάνουμε την εικόνα να "αναβοσβήσει" 4 φορές ανάποδα (out->in->out):

VB:

```
FadeInAndOut(iconStudent, TimeSpan.FromSeconds(1.5), _  
              TimeSpan.FromSeconds(0.7), 4, reverse:=True)
```

StopFadeInAndOutAnimation(element)

Σταματάει το fade-in and out animation στο **element**.

Σύνταξη:

VB :

```
Public Shared Sub StopFadeInAndOutAnimation(  
    element As FrameworkElement  
)
```

Παράμετροι:

eLement

Τύπος: System.Windows.FrameworkElement

To element στο οποίο ήδη τρέχει το εφέ.

Παρατηρήσεις:

Δεν πρόκειται να προκληθεί σφάλμα αν καλέσουμε τη μέθοδο χωρίς να τρέχει σε αυτό το fade-in and out animation.

Παράδειγμα:

VB:

```
StopFadeInAndOutAnimation(iconStudent)
```

FadeOut – Μέθοδοι

Όνομα	Περιγραφή
FadeOut (element As FrameworkElement, animationSpeed As AnimationSpeedEnum)	Εκτελεί το fade-out εφέ στο element με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum .
FadeOut (element As FrameworkElement, duration As TimeSpan)	Εκτελεί το fade-out εφέ στο element σε χρόνο duration .
FadeOut (element As FrameworkElement, beginTime As TimeSpan, animationSpeed As AnimationSpeedEnum)	Εκτελεί το fade-out εφέ στο element με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum . Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime .
FadeOut (element As FrameworkElement, beginTime As TimeSpan, Duration As TimeSpan)	Εκτελεί το fade-out εφέ στο element σε χρόνο duration . Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime .

FadeOut(element, animationSpeed)

Εκτελεί το fade-out εφέ στο **element** με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**.

Σύνταξη:

VB :

```
Public Shared Sub FadeOut(  
    element As FrameworkElement,  
    animationSpeed As AnimationSpeedEnum  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα TextBlock, με όνομα tbLastNameData, συνδεδεμένο με την ιδιότητα LastName ενός source αντικειμένου Student, του οποίου το περιεχόμενο εξαφανίζεται σταδιακά με ένα fade-out εφέ, μόλις ο χρήστης πατήσει το button μετακίνησης σε μία νέα εγγραφή μαθητή:

XAML:

```
<TextBlock Name="tbLastNameData" Text="{Binding LastName}" />
```

VB:

```
FadeOut(tbLastNameData, AnimationSpeedEnum.Normal)
```

FadeOut(element, duration)

Εκτελεί το fade-out εφέ στο **element** σε χρόνο **duration**.

Σύνταξη:

VB :

```
Public Shared Sub FadeOut(  
    element As FrameworkElement,  
    duration As TimeSpan  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα TextBlock, με όνομα tbLastNameData, συνδεδεμένο με την ιδιότητα LastName ενός source αντικειμένου Student, του οποίου το περιεχόμενο εξαφανίζεται σταδιακά με ένα fade-out εφέ, μόλις ο χρήστης πατήσει το button μετακίνησης σε μία νέα εγγραφή μαθητή:

XAML:

```
<TextBlock Name="tbLastNameData" Text="{Binding LastName}" />
```

VB:

```
FadeOut(tbLastNameData, TimeSpan.FromSeconds(0.7))
```

FadeOut(element, beginTime, animationSpeed)

Εκτελεί το fade-out εφέ στο **element** με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**. Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub FadeOut(  
    element As FrameworkElement,  
    beginTime As TimeSpan,  
    animationSpeed As AnimationSpeedEnum  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει δύο TextBlocks με όνομα tbLastNameData και tbFirstNameData, αντίστοιχα. Το πρώτο είναι συνδεδεμένο με την ιδιότητα LastName και το δεύτερο με την ιδιότητα FirstName ενός source αντικειμένου Student. Τα περιεχόμενά τους εξαφανίζονται διαδοχικά το ένα μετά το άλλο με ένα fade-out εφέ (με διαφορά 0.4 secs), μόλις ο χρήστης πατήσει το button μετακίνησης σε μία νέα εγγραφή μαθητή.

XAML:

```
<TextBlock Name="tbLastNameData" Text="{Binding LastName}" />  
<TextBlock Name="tbFirstNameData" Text="{Binding FirstName}" />
```

VB:

```
FadeOut(tbLastNameData, AnimationSpeedEnum.Normal)
```

```
FadeOut(tbFirstNameData, TimeSpan.FromSeconds(0.4), AnimationSpeedEnum.Normal)
```

FadeOut(element, beginTime, duration)

Εκτελεί το fade-out εφέ στο **element** σε χρόνο **duration**. Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub FadeOut(  
    element As FrameworkElement,  
    beginTime As TimeSpan,  
    duration As TimeSpan  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει δύο TextBlocks με όνομα tbLastNameData και tbFirstNameData, αντίστοιχα. Το πρώτο είναι συνδεδεμένο με την ιδιότητα LastName και το δεύτερο με την ιδιότητα FirstName ενός source αντικειμένου Student. Τα περιεχόμενά τους εξαφανίζονται διαδοχικά το ένα μετά το άλλο με ένα fade-out εφέ (με διαφορά 0.4 secs), μόλις ο χρήστης πατήσει το button μετακίνησης σε μία νέα εγγραφή μαθητή.

XAML:

```
<TextBlock Name="tbLastNameData" Text="{Binding LastName}" />  
<TextBlock Name="tbFirstNameData" Text="{Binding FirstName}" />
```


VB:

```
FadeOut(tbLastNameData, TimeSpan.FromSeconds(0.7))  
FadeOut(tbFirstNameData, TimeSpan.FromSeconds(0.4), TimeSpan.FromSeconds(0.7))
```

FadeOutAndMoveOut – Μέθοδοι

Όνομα	Περιγραφή
FadeOutAndMoveOut (element As FrameworkElement, offsetX As Double, offsetY As Double, animationSpeed As AnimationSpeedEnum)	Εκτελεί το fade-out και move-out εφέ στο element , ξεκινώντας από την original θέση προς την τελική θέση (offsetX, offsetY), με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum .
FadeOutAndMoveOut (element As FrameworkElement, offsetX As Double, offsetY As Double, duration As TimeSpan)	Εκτελεί το fade-out και move-out εφέ στο element , ξεκινώντας από την original θέση προς την τελική θέση (offsetX, offsetY) και σε χρόνο duration .
FadeOutAndMoveOut (element As FrameworkElement, offsetX As Double, offsetY As Double, beginTime As TimeSpan, animationSpeed As AnimationSpeedEnum)	Εκτελεί το fade-out και move-out εφέ στο element , ξεκινώντας από την original θέση προς την τελική θέση (offsetX, offsetY), με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum . Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime .
FadeOutAndMoveOut (element As FrameworkElement, offsetX As Double, offsetY As Double, beginTime As TimeSpan, duration As TimeSpan)	Εκτελεί το fade-out και move-out εφέ στο element , ξεκινώντας από την original θέση προς την τελική θέση (offsetX, offsetY) και σε χρόνο duration . Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime .

FadeOutAndMoveOut(element, offsetX, offsetY, animationSpeed)

Εκτελεί το fade-out και move-out εφέ στο **element**, ξεκινώντας από την original θέση προς την τελική θέση (**offsetX, offsetY**), με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**.

Σύνταξη:

VB :

```
Public Shared Sub FadeOutAndMoveOut(  
    element As FrameworkElement,  
    offsetX As Double,  
    offsetY As Double,  
    animationSpeed As AnimationSpeedEnum  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

offsetX

Τύπος: System.Double
Το σημείο x-point της θέσης (offsetX, offsetY), στο οποίο θα καταλήξει το εφέ. Για παράδειγμα, η τιμή -100 σημαίνει 100 pixels αριστερά (στον οριζόντιο άξονα), από την original θέση του element.

offsetY

Τύπος: System.Double
Το σημείο y-point της θέσης (offsetX, offsetY), στο οποίο θα καταλήξει το εφέ. Για παράδειγμα, η τιμή 50 σημαίνει 50 pixels παρακάτω (στον κατακόρυφο άξονα), από την original θέση του element.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα TextBlock, με όνομα tbLastName, του οποίου το περιεχόμενο εξαφανίζεται και μετακινείται σταδιακά με ένα fade-out και move-out εφέ, μόλις ο χρήστης πατήσει το button μετακίνησης σε μία νέα εγγραφή μαθητή. Το εφέ, ξεκινάει από την original θέση προς την τελική θέση (300, 0), δηλαδή 300 pixels

δεξιά και στην ίδια οριζόντια γραμμή. Το αποτέλεσμα δείχνει σαν να "σβήνει" το element κατά την μετακίνησή του, χρήσιμο για την αίσθηση της εξόδου του από τα δρώμενα.

XAML:

```
<TextBlock Name="tbLastName" />
```

VB:

```
FadeOutAndMoveOut(tbLastName, 300, 0, AnimationSpeedEnum.Fast)
```

FadeOutAndMoveOut(element, offsetX, offsetY, duration)

Εκτελεί το fade-out και move-out εφέ στο **element**, ξεκινώντας από την original θέση προς την τελική θέση (**offsetX, offsetY**) και σε χρόνο **duration**.

Σύνταξη:

VB :

```
Public Shared Sub FadeOutAndMoveOut(  
    element As FrameworkElement,  
    offsetX As Double,  
    offsetY As Double,  
    duration As TimeSpan  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

offsetX

Τύπος: System.Double
Το σημείο x-point της θέσης (offsetX, offsetY), στο οποίο θα καταλήξει το εφέ. Για παράδειγμα, η τιμή -100 σημαίνει 100 pixels αριστερά (στον οριζόντιο άξονα), από την original θέση του element.

offsetY

Τύπος: System.Double
Το σημείο y-point της θέσης (offsetX, offsetY), στο οποίο θα καταλήξει το εφέ. Για παράδειγμα, η τιμή 50 σημαίνει 50 pixels παρακάτω (στον κατακόρυφο άξονα), από την original θέση του element.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα TextBlock, με όνομα tbLastName, του οποίου το περιεχόμενο εξαφανίζεται και μετακινείται σταδιακά (σε χρόνο 0.7 secs) με ένα fade-out και move-out εφέ, μόλις ο χρήστης πατήσει το button μετακίνησης σε μία νέα εγγραφή μαθητή. Το εφέ, ξεκινάει από την original θέση προς την τελική θέση (300, 0), δηλαδή 300 pixels δεξιά και στην ίδια οριζόντια γραμμή. Το αποτέλεσμα δείχνει σαν να "σβήνει" το element κατά την μετακίνησή του, χρήσιμο για την αίσθηση της εξόδου του από τα δρώμενα.

XAML:

```
<TextBlock Name="tbLastName" />
```

VB:

```
FadeOutAndMoveOut(tbLastName, 300, 0, TimeSpan.FromSeconds(0.7))
```

FadeOutAndMoveOut(element, offsetX, offsetY, beginTime, animationSpeed)

Εκτελεί το fade-out και move-out εφέ στο **element**, ξεκινώντας από την original θέση προς την τελική θέση (**offsetX, offsetY**), με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**. Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub FadeOutAndMoveOut(  
    element As FrameworkElement,  
    offsetX As Double,  
    offsetY As Double,  
    beginTime As TimeSpan,  
    animationSpeed As AnimationSpeedEnum  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

offsetX

Τύπος: System.Double
Το σημείο x-point της θέσης (offsetX, offsetY), στο οποίο θα καταλήξει το εφέ. Για παράδειγμα, η τιμή -100 σημαίνει 100 pixels αριστερά (στον οριζόντιο άξονα), από την original θέση του element.

offsetY

Τύπος: System.Double
Το σημείο y-point της θέσης (offsetX, offsetY), στο οποίο θα καταλήξει το εφέ. Για παράδειγμα, η τιμή 50 σημαίνει 50 pixels παρακάτω (στον κατακόρυφο άξονα), από την original θέση του element.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα `TextBlock`, με όνομα `tbLastName`, του οποίου το περιεχόμενο εξαφανίζεται και μετακινείται σταδιακά με ένα `fade-out` και `move-out` εφέ, μόλις ο χρήστης πατήσει το `button` μετακίνησης σε μία νέα εγγραφή μαθητή. Το εφέ, ξεκινάει από την `original` θέση προς την τελική θέση (300, 0), δηλαδή 300 pixels δεξιά και στην ίδια οριζόντια γραμμή. Το αποτέλεσμα δείχνει σαν να "σβήνει" το `element` κατά την μετακίνησή του, χρήσιμο για την αίσθηση της εξόδου του από τα δρώμενα. Η έναρξη του εφέ είναι μετά από 1.5 secs.

XAML:

```
<TextBlock Name="tbLastName" />
```

VB:

```
FadeOutAndMoveOut(tbLastName, 300, 0, TimeSpan.FromSeconds(1.5), _  
                    AnimationSpeedEnum.Fast)
```


FadeOutAndMoveOut(element, offsetX, offsetY, beginTime, duration)

Εκτελεί το fade-out και move-out εφέ στο **element**, ξεκινώντας από την original θέση προς την τελική θέση (**offsetX, offsetY**) και σε χρόνο **duration**. Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub FadeOutAndMoveOut(  
    element As FrameworkElement,  
    offsetX As Double,  
    offsetY As Double,  
    beginTime As TimeSpan,  
    duration As TimeSpan  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

offsetX

Τύπος: System.Double
Το σημείο x-point της θέσης (offsetX, offsetY), στο οποίο θα καταλήξει το εφέ. Για παράδειγμα, η τιμή -100 σημαίνει 100 pixels αριστερά (στον οριζόντιο άξονα), από την original θέση του element.

offsetY

Τύπος: System.Double
Το σημείο y-point της θέσης (offsetX, offsetY), στο οποίο θα καταλήξει το εφέ. Για παράδειγμα, η τιμή 50 σημαίνει 50 pixels παρακάτω (στον κατακόρυφο άξονα), από την original θέση του element.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα TextBlock, με όνομα tbLastName, του οποίου το περιεχόμενο εξαφανίζεται και μετακινείται σταδιακά (σε χρόνο 0.7 secs) με ένα

fade-out και move-out εφέ, μόλις ο χρήστης πατήσει το button μετακίνησης σε μία νέα εγγραφή μαθητή. Το εφέ, ξεκινάει από την original θέση προς την τελική θέση (300, 0), δηλαδή 300 pixels δεξιά και στην ίδια οριζόντια γραμμή. Το αποτέλεσμα δείχνει σαν να "σβήνει" το element κατά την μετακίνησή του, χρήσιμο για την αίσθηση της εξόδου του από τα δρώμενα. Η έναρξη του εφέ είναι μετά από 1.5 secs.

XAML:

```
<TextBlock Name="tbLastName" />
```

VB:

```
FadeOutAndMoveOut(tbLastName, 300, 0, TimeSpan.FromSeconds(1.5), _  
                    TimeSpan.FromSeconds(0.7))
```

ZoomIn - Μέθοδοι

Όνομα	Περιγραφή
ZoomIn (element As FrameworkElement, zoomFactor As Single, animationSpeed As AnimationSpeedEnum)	Εκτελεί το zoom-in εφέ στο element , με συντελεστή μεγέθυνσης zoomFactor και με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum .
ZoomIn (element As FrameworkElement, zoomFactor As Single, beginTime As TimeSpan, animationSpeed As AnimationSpeedEnum)	Εκτελεί το zoom-in εφέ στο element , με συντελεστή μεγέθυνσης zoomFactor και με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum . Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime .
ZoomIn (element As FrameworkElement, zoomFactor As Single, pointOrigin As Point, animationSpeed As AnimationSpeedEnum)	Εκτελεί το zoom-in εφέ στο element , με συντελεστή μεγέθυνσης zoomFactor , με κεντρικό σημείο το pointOrigin και με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum .
ZoomIn (element As FrameworkElement, zoomFactor As Single, pointOrigin As Point, beginTime As TimeSpan, animationSpeed As AnimationSpeedEnum)	Εκτελεί το zoom-in εφέ στο element , με συντελεστή μεγέθυνσης zoomFactor , με κεντρικό σημείο το pointOrigin και με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum . Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime .
ZoomIn (element As FrameworkElement, zoomFactor As Single, duration As TimeSpan)	Εκτελεί το zoom-in εφέ στο element , με συντελεστή μεγέθυνσης zoomFactor και σε χρόνο duration .
ZoomIn (element As FrameworkElement, zoomFactor As Single, beginTime As TimeSpan, duration As TimeSpan)	Εκτελεί το zoom-in εφέ στο element , με συντελεστή μεγέθυνσης zoomFactor και σε χρόνο duration . Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime .

ZoomIn (element As FrameworkElement, zoomFactor As Single, pointOrigin As Point, duration As TimeSpan)	Εκτελεί το zoom-in εφέ στο element , με συντελεστή μεγέθυνσης zoomFactor , με κεντρικό σημείο το pointOrigin και σε χρόνο duration .
ZoomIn (element As FrameworkElement, zoomFactor As Single, pointOrigin As Point, beginTime As TimeSpan, duration As TimeSpan)	Εκτελεί το zoom-in εφέ στο element , με συντελεστή μεγέθυνσης zoomFactor , με κεντρικό σημείο το pointOrigin και σε χρόνο duration . Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime .

ZoomIn(element, zoomFactor, animationSpeed)

Εκτελεί το zoom-in εφέ στο **element**, με συντελεστή μεγέθυνσης **zoomFactor** και με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomIn(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    animationSpeed As AnimationSpeedEnum  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single
Ο συντελεστής μεγέθυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η μεγέθυνση θα ξεκινήσει από το 0 προς το 1.2. Δηλαδή, μεγέθυνση 120%.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να μεγεθυνθεί 120%.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"  
    Width="64" Height="64"/>
```

VB:

```
ZoomIn(iconStudent, 1.2, AnimationSpeedEnum.Normal)
```

ZoomIn(element, zoomFactor, beginTime, animationSpeed)

Εκτελεί το zoom-in εφέ στο **element**, με συντελεστή μεγέθυνσης **zoomFactor** και με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**. Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomIn(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    beginTime As TimeSpan,  
    animationSpeed As AnimationSpeedEnum  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single
Ο συντελεστής μεγέθυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η μεγέθυνση θα ξεκινήσει από το 0 προς το 1.2. Δηλαδή, μεγέθυνση 120%.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να μεγεθυνθεί 120%. Το εφέ ξεκινάει με χρονοκαθυστέρηση 1.5 secs.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"  
    Width="64" Height="64"/>
```

VB:

```
ZoomIn(iconStudent, 1.2, TimeSpan.FromSeconds(1.5), AnimationSpeedEnum.Normal)
```

ZoomIn(element, zoomFactor, pointOrigin, animationSpeed)

Εκτελεί το zoom-in εφέ στο **element**, με συντελεστή μεγέθυνσης **zoomFactor**, με κεντρικό σημείο το **pointOrigin** και με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomIn(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    pointOrigin As Point,  
    animationSpeed As AnimationSpeedEnum  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single
Ο συντελεστής μεγέθυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η μεγέθυνση θα ξεκινήσει από το 0 προς το 1.2. Δηλαδή, μεγέθυνση 120%.

pointOrigin

Τύπος: System.Windows.Point
Καθορίζει το κεντρικό σημείο από το οποίο γίνεται η μεγέθυνση (pivot). Η τιμή εκτείνεται στο διάστημα από το [0,0] έως το [1,1], όπου το [0,0] υποδηλώνει το πάνω αριστερό μέρος του element ενώ το [1,1] υποδηλώνει το κάτω δεξί. Για παράδειγμα, η τιμή (0.5, 1) υποδηλώνει το κάτω μεσαίο μέρος. Η default τιμή είναι (0.5, 0.5), δηλαδή το μέσον του.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να μεγεθυνθεί 120%. Ορίζουμε ως κεντρικό σημείο μεγέθυνσης το πάνω μεσαίο μέρος (θέση (0.5, 0)).

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"  
      Width="64" Height="64"/>
```

VB:

```
ZoomIn(iconStudent, 1.2, new Point(0.5, 0), AnimationSpeedEnum.Normal)
```

ZoomIn(element, zoomFactor, pointOrigin, beginTime, animationSpeed)

Εκτελεί το zoom-in εφέ στο **element**, με συντελεστή μεγέθυνσης **zoomFactor**, με κεντρικό σημείο το **pointOrigin** και με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**. Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomIn(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    pointOrigin As Point,  
    beginTime As TimeSpan,  
    animationSpeed As AnimationSpeedEnum  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement

Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single

Ο συντελεστής μεγέθυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η μεγέθυνση θα ξεκινήσει από το 0 προς το 1.2. Δηλαδή, μεγέθυνση 120%.

pointOrigin

Τύπος: System.Windows.Point

Καθορίζει το κεντρικό σημείο από το οποίο γίνεται η μεγέθυνση (pivot). Η τιμή εκτείνεται στο διάστημα από το [0,0] έως το [1,1], όπου το [0,0] υποδηλώνει το πάνω αριστερό μέρος του element ενώ το [1,1] υποδηλώνει το κάτω δεξί. Για παράδειγμα, η τιμή (0.5, 1) υποδηλώνει το κάτω μεσαίο μέρος. Η default τιμή είναι (0.5, 0.5), δηλαδή το μέσον του.

beginTime

Τύπος: System.TimeSpan

Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum

Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να μεγεθυνθεί 120%. Ορίζουμε ως κεντρικό σημείο μεγέθυνσης το πάνω μεσαίο μέρος (θέση (0.5, 0)). Το εφέ ξεκινάει με χρονοκαθυστέρηση 2 secs.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"
        Width="64" Height="64"/>
```

VB:

```
ZoomIn(iconStudent, 1.2, new Point(0.5, 0), TimeSpan.FromSeconds(2), _
        AnimationSpeedEnum.Normal)
```

ZoomIn(element, zoomFactor, duration)

Εκτελεί το zoom-in εφέ στο **element**, με συντελεστή μεγέθυνσης **zoomFactor** και σε χρόνο **duration**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomIn(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    duration As TimeSpan  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single
Ο συντελεστής μεγέθυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η μεγέθυνση θα ξεκινήσει από το 0 προς το 1.2. Δηλαδή, μεγέθυνση 120%.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να μεγεθυνθεί 120% σε χρόνο 0.7 secs.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"  
    Width="64" Height="64"/>
```

VB:

```
ZoomIn(iconStudent, 1.2, TimeSpan.FromSeconds(0.7))
```

ZoomIn(element, zoomFactor, beginTime, duration)

Εκτελεί το zoom-in εφέ στο **element**, με συντελεστή μεγέθυνσης **zoomFactor** και σε χρόνο **duration**. Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomIn(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    beginTime As TimeSpan,  
    duration As TimeSpan  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single
Ο συντελεστής μεγέθυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η μεγέθυνση θα ξεκινήσει από το 0 προς το 1.2. Δηλαδή, μεγέθυνση 120%.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να μεγεθυνθεί 120% σε χρόνο 0.7 secs. Το εφέ ξεκινάει με χρονοκαθυστέρηση 1.5 secs.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"  
    Width="64" Height="64"/>
```

VB:

```
ZoomIn(iconStudent, 1.2, TimeSpan.FromSeconds(1.5), TimeSpan.FromSeconds(0.7))
```

ZoomIn(element, zoomFactor, pointOrigin, duration)

Εκτελεί το zoom-in εφέ στο **element**, με συντελεστή μεγέθυνσης **zoomFactor**, με κεντρικό σημείο το **pointOrigin** και σε χρόνο **duration**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomIn(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    pointOrigin As Point,  
    duration As TimeSpan  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single

Ο συντελεστής μεγέθυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η μεγέθυνση θα ξεκινήσει από το 0 προς το 1.2. Δηλαδή, μεγέθυνση 120%.

pointOrigin

Τύπος: System.Windows.Point

Καθορίζει το κεντρικό σημείο από το οποίο γίνεται η μεγέθυνση (pivot). Η τιμή εκτείνεται στο διάστημα από το [0,0] έως το [1,1], όπου το [0,0] υποδηλώνει το πάνω αριστερό μέρος του element ενώ το [1,1] υποδηλώνει το κάτω δεξί. Για παράδειγμα, η τιμή (0.5, 1) υποδηλώνει το κάτω μεσαίο μέρος. Η default τιμή είναι (0.5, 0.5), δηλαδή το μέσον του.

duration

Τύπος: System.TimeSpan

Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να μεγεθυνθεί 120% σε χρόνο 0.7 secs. Ορίζουμε ως κεντρικό σημείο μεγέθυνσης το πάνω μεσαίο μέρος (θέση (0.5, 0)).

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"  
      Width="64" Height="64"/>
```

VB:

```
ZoomIn(iconStudent, 1.2, new Point(0.5, 0), TimeSpan.FromSeconds(0.7))
```


ZoomIn(element, zoomFactor, pointOrigin, beginTime, duration)

Εκτελεί το zoom-in εφέ στο **element**, με συντελεστή μεγέθυνσης **zoomFactor**, με κεντρικό σημείο το **pointOrigin** και σε χρόνο **duration**. Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomIn(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    pointOrigin As Point,  
    beginTime As TimeSpan,  
    duration As TimeSpan  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single
Ο συντελεστής μεγέθυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η μεγέθυνση θα ξεκινήσει από το 0 προς το 1.2. Δηλαδή, μεγέθυνση 120%.

pointOrigin

Τύπος: System.Windows.Point
Καθορίζει το κεντρικό σημείο από το οποίο γίνεται η μεγέθυνση (pivot). Η τιμή εκτείνεται στο διάστημα από το [0,0] έως το [1,1], όπου το [0,0] υποδηλώνει το πάνω αριστερό μέρος του element ενώ το [1,1] υποδηλώνει το κάτω δεξί. Για παράδειγμα, η τιμή (0.5, 1) υποδηλώνει το κάτω μεσαίο μέρος. Η default τιμή είναι (0.5, 0.5), δηλαδή το μέσον του.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να μεγεθυνθεί 120% σε χρόνο 0.7 secs. Ορίζουμε ως κεντρικό σημείο μεγέθυνσης το πάνω μεσαίο μέρος (θέση (0.5, 0)). Το εφέ ξεκινάει με χρονοκαθυστέρηση 2 secs.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"
      Width="64" Height="64"/>
```

VB:

```
ZoomIn(iconStudent, 1.2, new Point(0.5, 0), TimeSpan.FromSeconds(2), _
      TimeSpan.FromSeconds(0.7))
```

ZoomInAndOut – Μέθοδοι

Όνομα	Περιγραφή
ZoomInAndOut (element As FrameworkElement, zoomFactor As Single, animationSpeed As AnimationSpeedEnum, repeatTimes As Integer, Optional repeatForever As Boolean = False)	<p>Εκτελεί το zoom-in and out εφέ στο element, με συντελεστή μεγέθυνσης zoomFactor και με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum.</p> <p>Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την repeatTimes εκτός και αν θέσουμε σαφώς την repeatForever, οπότε θα επαναλαμβάνεται συνεχώς.</p>
ZoomInAndOut (element As FrameworkElement, zoomFactor As Single, beginTime As TimeSpan, animationSpeed As AnimationSpeedEnum, repeatTimes As Integer, Optional repeatForever As Boolean = False)	<p>Εκτελεί το zoom-in and out εφέ στο element, με συντελεστή μεγέθυνσης zoomFactor και με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum.</p> <p>Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την repeatTimes εκτός και αν θέσουμε σαφώς την repeatForever, οπότε θα επαναλαμβάνεται συνεχώς.</p> <p>Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime.</p>
ZoomInAndOut (element As FrameworkElement, zoomFactor As Single, pointOrigin As Point, animationSpeed As AnimationSpeedEnum, repeatTimes As Integer, Optional repeatForever As Boolean = False)	<p>Εκτελεί το zoom-in and out εφέ στο element, με συντελεστή μεγέθυνσης zoomFactor, με κεντρικό σημείο το pointOrigin και με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum.</p> <p>Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την repeatTimes εκτός και αν θέσουμε σαφώς την repeatForever, οπότε θα επαναλαμβάνεται συνεχώς.</p>

<p>ZoomInAndOut(element As FrameworkElement, zoomFactor As Single, pointOrigin As Point, beginTime As TimeSpan, animationSpeed As AnimationSpeedEnum, repeatTimes As Integer, Optional repeatForever As Boolean = False)</p>	<p>Εκτελεί το zoom-in and out εφέ στο element, με συντελεστή μεγέθυνσης zoomFactor, με κεντρικό σημείο το pointOrigin και με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum.</p> <p>Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την repeatTimes εκτός και αν θέσουμε σαφώς την repeatForever, οπότε θα επαναλαμβάνεται συνεχώς.</p> <p>Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime.</p>
<p>ZoomInAndOut(element As FrameworkElement, zoomFactor As Single, duration As TimeSpan, repeatTimes As Integer, Optional repeatForever As Boolean = False)</p>	<p>Εκτελεί το zoom-in and out εφέ στο element, με συντελεστή μεγέθυνσης zoomFactor και σε χρόνο duration.</p> <p>Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την repeatTimes εκτός και αν θέσουμε σαφώς την repeatForever, οπότε θα επαναλαμβάνεται συνεχώς.</p>
<p>ZoomInAndOut(element As FrameworkElement, zoomFactor As Single, beginTime As TimeSpan, duration As TimeSpan, repeatTimes As Integer, Optional repeatForever As Boolean = False)</p>	<p>Εκτελεί το zoom-in and out εφέ στο element, με συντελεστή μεγέθυνσης zoomFactor και σε χρόνο duration.</p> <p>Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την repeatTimes εκτός και αν θέσουμε σαφώς την repeatForever, οπότε θα επαναλαμβάνεται συνεχώς.</p> <p>Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime.</p>
<p>ZoomInAndOut(element As FrameworkElement, zoomFactor As Single, pointOrigin As Point, duration As TimeSpan, repeatTimes As Integer, Optional repeatForever As Boolean = False)</p>	<p>Εκτελεί το zoom-in and out εφέ στο element, με συντελεστή μεγέθυνσης zoomFactor, με κεντρικό σημείο το pointOrigin και σε χρόνο duration.</p> <p>Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την repeatTimes εκτός και αν θέσουμε σαφώς την repeatForever, οπότε θα επαναλαμβάνεται συνεχώς.</p>

<p>ZoomInAndOut(element As FrameworkElement, zoomFactor As Single, pointOrigin As Point, beginTime As TimeSpan, duration As TimeSpan, repeatTimes As Integer, Optional repeatForever As Boolean = False)</p>	<p>Εκτελεί το zoom-in and out εφέ στο element, με συντελεστή μεγέθυνσης zoomFactor, με κεντρικό σημείο το pointOrigin και σε χρόνο duration.</p> <p>Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την repeatTimes εκτός και αν θέσουμε σαφώς την repeatForever, οπότε θα επαναλαμβάνεται συνεχώς.</p> <p>Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime.</p>
<p>StopZoomInAndOutAnimation(element As FrameworkElement)</p>	<p>Σταματάει το zoom-in and out animation στο element.</p>

ZoomInAndOut(element, zoomFactor, animationSpeed, repeatTimes, Optional repeatForever)

Εκτελεί το zoom-in and out εφέ στο **element**, με συντελεστή μεγέθυνσης **zoomFactor** και με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**. Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την **repeatTimes** εκτός και αν θέσουμε σαφώς την **repeatForever**, οπότε θα επαναλαμβάνεται συνεχώς.

Σύνταξη:

VB :

```
Public Shared Sub ZoomInAndOut(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    animationSpeed As AnimationSpeedEnum,  
    repeatTimes As Integer,  
    Optional repeatForever As Boolean  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single
Ο συντελεστής μεγέθυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η μεγέθυνση θα ξεκινήσει από το κανονικό μέγεθος (1.0) προς το 1.2. Δηλαδή, μεγέθυνση 20%. Αν θέσουμε την τιμή 0.4 τότε θα ξεκινήσει από το κανονικό μέγεθος (1.0) προς το 0.4, δηλαδή σμίκρυνση κατά 60%.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

repeatTimes

Τύπος: System.Integer
Καθορίζει πόσες φορές θα επαναλαμβάνεται το εφέ (δηλαδή, πόσες φορές θα κάνει την μεγέθυνση/σμίκρυνση) . Για παράδειγμα, η τιμή 4 σημαίνει 4 φορές θα μεγαλώνει/μικραίνει. Τεχνικά μιλώντας, πόσες φορές θα κάνει τον κύκλο zoom in->out->in.

Optional repeatForever

Τύπος: System.Boolean
Αν είναι True, τότε το εφέ θα εκτελείται συνεχώς. Σε αυτήν την περίπτωση αγνοείται η τιμή της παραμέτρου repeatTimes. Η default τιμή είναι False.

Ο μόνος τρόπος να σταματήσει είναι να καλέσουμε τη μέθοδο `StopZoomInAndOutAnimation`.

Παράδειγμα 1:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα `iconStudent`. Στον VB κώδικα, κάνουμε την εικόνα να μεγεθύνεται/σμικραίνει 4 φορές κατά 20%.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"
        Width="64" Height="64"/>
```

VB:

```
ZoomInAndOut(iconStudent, 1.2, AnimationSpeedEnum.Normal, 4)
```

Παράδειγμα 2:

Στο ίδιο παράδειγμα με το παραπάνω, αλλά κάνοντας το εφέ να επαναλαμβάνεται συνεχώς:

VB:

```
ZoomInAndOut(iconStudent, 1.2, AnimationSpeedEnum.Normal, 0, repeatForever:=True)
```

ZoomInAndOut(element, zoomFactor, beginTime, animationSpeed, repeatTimes, Optional repeatForever)

Εκτελεί το zoom-in and out εφέ στο **element**, με συντελεστή μεγέθυνσης **zoomFactor** και με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**. Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την **repeatTimes** εκτός και αν θέσουμε σαφώς την **repeatForever**, οπότε θα επαναλαμβάνεται συνεχώς. Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomInAndOut(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    beginTime As TimeSpan,  
    animationSpeed As AnimationSpeedEnum,  
    repeatTimes As Integer,  
    Optional repeatForever As Boolean  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single
Ο συντελεστής μεγέθυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η μεγέθυνση θα ξεκινήσει από το κανονικό μέγεθος (1.0) προς το 1.2. Δηλαδή, μεγέθυνση 20%. Αν θέσουμε την τιμή 0.4 τότε θα ξεκινήσει από το κανονικό μέγεθος (1.0) προς το 0.4, δηλαδή σμίκρυνση κατά 60%.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

repeatTimes

Τύπος: System.Integer
Καθορίζει πόσες φορές θα επαναλαμβάνεται το εφέ (δηλαδή, πόσες φορές θα κάνει την μεγέθυνση/σμίκρυνση). Για παράδειγμα, η τιμή 4 σημαίνει 4 φορές θα

μεγαλώνει/μικραίνει. Τεχνικά μιλώντας, πόσες φορές θα κάνει τον κύκλο zoom in->out->in.

Optional repeatForever

Τύπος: System.Boolean

Αν είναι True, τότε το εφέ θα εκτελείται συνεχώς. Σε αυτήν την περίπτωση αγνοείται η τιμή της παραμέτρου repeatTimes. Η default τιμή είναι False.

Ο μόνος τρόπος να σταματήσει είναι να καλέσουμε τη μέθοδο StopZoomInAndOutAnimation.

Παράδειγμα 1:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να μεγεθύνεται/σμικραίνει 4 φορές κατά 20%. Το εφέ ξεκινάει με χρονοκαθυστέρηση 1.5 secs.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"
        Width="64" Height="64"/>
```

VB:

```
ZoomInAndOut(iconStudent, 1.2, TimeSpan.FromSeconds(1.5), _
              AnimationSpeedEnum.Normal, 4)
```

Παράδειγμα 2:

Στο ίδιο παράδειγμα με το παραπάνω, αλλά κάνοντας το εφέ να επαναλαμβάνεται συνεχώς:

VB:

```
ZoomInAndOut(iconStudent, 1.2, TimeSpan.FromSeconds(1.5), _
              AnimationSpeedEnum.Normal, 0, repeatForever:=True)
```

ZoomInAndOut(element, zoomFactor, pointOrigin, animationSpeed, repeatTimes, Optional repeatForever)

Εκτελεί το zoom-in and out εφέ στο **element**, με συντελεστή μεγέθυνσης **zoomFactor**, με κεντρικό σημείο το **pointOrigin** και με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**. Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την **repeatTimes** εκτός και αν θέσουμε σαφώς την **repeatForever**, οπότε θα επαναλαμβάνεται συνεχώς.

Σύνταξη:

VB :

```
Public Shared Sub ZoomInAndOut(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    pointOrigin As Point,  
    animationSpeed As AnimationSpeedEnum,  
    repeatTimes As Integer,  
    Optional repeatForever As Boolean  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single
Ο συντελεστής μεγέθυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η μεγέθυνση θα ξεκινήσει από το κανονικό μέγεθος (1.0) προς το 1.2. Δηλαδή, μεγέθυνση 20%. Αν θέσουμε την τιμή 0.4 τότε θα ξεκινήσει από το κανονικό μέγεθος (1.0) προς το 0.4, δηλαδή σμίκρυνση κατά 60%.

pointOrigin

Τύπος: System.Windows.Point
Καθορίζει το κεντρικό σημείο από το οποίο γίνεται η μεγέθυνση (pivot). Η τιμή εκτείνεται στο διάστημα από το [0,0] έως το [1,1], όπου το [0,0] υποδηλώνει το πάνω αριστερό μέρος του element ενώ το [1,1] υποδηλώνει το κάτω δεξί. Για παράδειγμα, η τιμή (0.5, 1) υποδηλώνει το κάτω μεσαίο μέρος. Η default τιμή είναι (0.5, 0.5), δηλαδή το μέσον του.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

repeatTimes

Τύπος: System.Integer

Καθορίζει πόσες φορές θα επαναλαμβάνεται το εφέ (δηλαδή, πόσες φορές θα κάνει την μεγέθυνση/σμίκρυνση) . Για παράδειγμα, η τιμή 4 σημαίνει 4 φορές θα μεγαλώνει/μικραίνει. Τεχνικά μιλώντας, πόσες φορές θα κάνει τον κύκλο zoom in->out->in.

Optional repeatForever

Τύπος: System.Boolean

Αν είναι True, τότε το εφέ θα εκτελείται συνεχώς. Σε αυτήν την περίπτωση αγνοείται η τιμή της παραμέτρου repeatTimes. Η default τιμή είναι False.

Ο μόνος τρόπος να σταματήσει είναι να καλέσουμε τη μέθοδο StopZoomInAndOutAnimation.

Παράδειγμα 1:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να μεγεθύνεται/σμικραίνει 4 φορές κατά 20%. Ορίζουμε ως κεντρικό σημείο μεγέθυνσης το κάτω μεσαίο μέρος (θέση (0.5, 1)).

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"
      Width="64" Height="64"/>
```

VB:

```
ZoomInAndOut(iconStudent, 1.2, new Point(0.5, 1), AnimationSpeedEnum.Normal, 4)
```

Παράδειγμα 2:

Στο ίδιο παράδειγμα με το παραπάνω, αλλά κάνοντας το εφέ να επαναλαμβάνεται συνεχώς:

VB:

```
ZoomInAndOut(iconStudent, 1.2, new Point(0.5, 1), _
              AnimationSpeedEnum.Normal, 0, repeatForever:=True)
```

ZoomInAndOut(element, zoomFactor, pointOrigin, beginTime, animationSpeed, repeatTimes, Optional repeatForever)

Εκτελεί το zoom-in and out εφέ στο **element**, με συντελεστή μεγέθυνσης **zoomFactor**, με κεντρικό σημείο το **pointOrigin** και με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**. Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την **repeatTimes** εκτός και αν θέσουμε σαφώς την **repeatForever**, οπότε θα επαναλαμβάνεται συνεχώς.

Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomInAndOut(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    pointOrigin As Point,  
    beginTime As TimeSpan,  
    animationSpeed As AnimationSpeedEnum,  
    repeatTimes As Integer,  
    Optional repeatForever As Boolean  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single
Ο συντελεστής μεγέθυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η μεγέθυνση θα ξεκινήσει από το κανονικό μέγεθος (1.0) προς το 1.2. Δηλαδή, μεγέθυνση 20%. Αν θέσουμε την τιμή 0.4 τότε θα ξεκινήσει από το κανονικό μέγεθος (1.0) προς το 0.4, δηλαδή σμίκρυνση κατά 60%.

pointOrigin

Τύπος: System.Windows.Point
Καθορίζει το κεντρικό σημείο από το οποίο γίνεται η μεγέθυνση (pivot). Η τιμή εκτείνεται στο διάστημα από το [0,0] έως το [1,1], όπου το [0,0] υποδηλώνει το πάνω αριστερό μέρος του element ενώ το [1,1] υποδηλώνει το κάτω δεξί. Για παράδειγμα, η τιμή (0.5, 1) υποδηλώνει το κάτω μεσαίο μέρος. Η default τιμή είναι (0.5, 0.5), δηλαδή το μέσον του.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

animationSpeed

Τύπος: `AnimationUtilities.AnimationSpeedEnum`

Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης `AnimationSpeedEnum`. Για παράδειγμα, `AnimationSpeedEnum.Fast`.

repeatTimes

Τύπος: `System.Integer`

Καθορίζει πόσες φορές θα επαναλαμβάνεται το εφέ (δηλαδή, πόσες φορές θα κάνει την μεγέθυνση/σμίκρυνση) . Για παράδειγμα, η τιμή 4 σημαίνει 4 φορές θα μεγαλώνει/μικραίνει. Τεχνικά μιλώντας, πόσες φορές θα κάνει τον κύκλο `zoom in->out->in`.

Optional repeatForever

Τύπος: `System.Boolean`

Αν είναι `True`, τότε το εφέ θα εκτελείται συνεχώς. Σε αυτήν την περίπτωση αγνοείται η τιμή της παραμέτρου `repeatTimes`. Η default τιμή είναι `False`.

Ο μόνος τρόπος να σταματήσει είναι να καλέσουμε τη μέθοδο `StopZoomInAndOutAnimation`.

Παράδειγμα 1:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα `Image` element, με όνομα `iconStudent`. Στον VB κώδικα, κάνουμε την εικόνα να μεγεθύνεται/σμικραίνει 4 φορές κατά 20%. Ορίζουμε ως κεντρικό σημείο μεγέθυνσης το κάτω μεσαίο μέρος (θέση (0.5, 1)). Το εφέ ξεκινάει με χρονοκαθυστέρηση 1.5 secs.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"
        Width="64" Height="64"/>
```

VB:

```
ZoomInAndOut(iconStudent, 1.2, new Point(0.5, 1), TimeSpan.FromSeconds(1.5), _
              AnimationSpeedEnum.Normal, 4)
```

Παράδειγμα 2:

Στο ίδιο παράδειγμα με το παραπάνω, αλλά κάνοντας το εφέ να επαναλαμβάνεται συνεχώς:

VB:

```
ZoomInAndOut(iconStudent, 1.2, new Point(0.5, 1), _
              TimeSpan.FromSeconds(1.5), AnimationSpeedEnum.Normal, 0, _
              repeatForever:= True)
```

ZoomInAndOut(element, zoomFactor, duration, repeatTimes, Optional repeatForever)

Εκτελεί το zoom-in and out εφέ στο **element**, με συντελεστή μεγέθυνσης **zoomFactor** και σε χρόνο **duration**. Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την **repeatTimes** εκτός και αν θέσουμε σαφώς την **repeatForever**, οπότε θα επαναλαμβάνεται συνεχώς.

Σύνταξη:

VB :

```
Public Shared Sub ZoomInAndOut(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    duration As TimeSpan,  
    repeatTimes As Integer,  
    Optional repeatForever As Boolean  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single
Ο συντελεστής μεγέθυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η μεγέθυνση θα ξεκινήσει από το κανονικό μέγεθος (1.0) προς το 1.2. Δηλαδή, μεγέθυνση 20%. Αν θέσουμε την τιμή 0.4 τότε θα ξεκινήσει από το κανονικό μέγεθος (1.0) προς το 0.4, δηλαδή σμίκρυνση κατά 60%.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

repeatTimes

Τύπος: System.Integer
Καθορίζει πόσες φορές θα επαναλαμβάνεται το εφέ (δηλαδή, πόσες φορές θα κάνει την μεγέθυνση/σμίκρυνση). Για παράδειγμα, η τιμή 4 σημαίνει 4 φορές θα μεγαλώνει/μικραίνει. Τεχνικά μιλώντας, πόσες φορές θα κάνει τον κύκλο zoom in->out->in.

Optional repeatForever

Τύπος: System.Boolean
Αν είναι True, τότε το εφέ θα εκτελείται συνεχώς. Σε αυτήν την περίπτωση αγνοείται η τιμή της παραμέτρου repeatTimes. Η default τιμή είναι False.
Ο μόνος τρόπος να σταματήσει είναι να καλέσουμε τη μέθοδο StopZoomInAndOutAnimation.

Παράδειγμα 1:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να μεγεθύνεται/σμικραίνει 4 φορές κατά 20% σε χρόνο 0.7 secs.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"
        Width="64" Height="64"/>
```

VB:

```
ZoomInAndOut(iconStudent, 1.2, TimeSpan.FromSeconds(0.7), 4)
```

Παράδειγμα 2:

Στο ίδιο παράδειγμα με το παραπάνω, αλλά κάνοντας το εφέ να επαναλαμβάνεται συνεχώς:

VB:

```
ZoomInAndOut(iconStudent, 1.2, TimeSpan.FromSeconds(0.7), 0, repeatForever:=True)
```

ZoomInAndOut(element, zoomFactor, beginTime, duration, repeatTimes, Optional repeatForever)

Εκτελεί το zoom-in and out εφέ στο **element**, με συντελεστή μεγέθυνσης **zoomFactor** και σε χρόνο **duration**. Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την **repeatTimes** εκτός και αν θέσουμε σαφώς την **repeatForever**, οπότε θα επαναλαμβάνεται συνεχώς. Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomInAndOut(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    beginTime As TimeSpan,  
    duration As TimeSpan,  
    repeatTimes As Integer,  
    Optional repeatForever As Boolean  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single

Ο συντελεστής μεγέθυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η μεγέθυνση θα ξεκινήσει από το κανονικό μέγεθος (1.0) προς το 1.2. Δηλαδή, μεγέθυνση 20%. Αν θέσουμε την τιμή 0.4 τότε θα ξεκινήσει από το κανονικό μέγεθος (1.0) προς το 0.4, δηλαδή σμίκρυνση κατά 60%.

beginTime

Τύπος: System.TimeSpan

Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή `TimeSpan.FromSeconds(1.5)` σημαίνει καθυστέρηση έναρξης 1.5 secs.

duration

Τύπος: System.TimeSpan

Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή `TimeSpan`. Για παράδειγμα, η τιμή `TimeSpan.FromSeconds(0.7)` σημαίνει 0.7 secs.

repeatTimes

Τύπος: System.Integer

Καθορίζει πόσες φορές θα επαναλαμβάνεται το εφέ (δηλαδή, πόσες φορές θα κάνει την μεγέθυνση/σμίκρυνση). Για παράδειγμα, η τιμή 4 σημαίνει 4 φορές θα μεγαλώνει/μικραίνει. Τεχνικά μιλώντας, πόσες φορές θα κάνει τον κύκλο zoom in->out->in.

Optional repeatForever

Τύπος: System.Boolean

Αν είναι True, τότε το εφέ θα εκτελείται συνεχώς. Σε αυτήν την περίπτωση αγνοείται η τιμή της παραμέτρου repeatTimes. Η default τιμή είναι False.

Ο μόνος τρόπος να σταματήσει είναι να καλέσουμε τη μέθοδο StopZoomInAndOutAnimation.

Παράδειγμα 1:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να μεγεθύνεται/σμικραίνει 4 φορές κατά 20% σε χρόνο 0.7 secs. Το εφέ ξεκινάει με χρονοκαθυστέρηση 1.5 secs.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"
        Width="64" Height="64"/>
```

VB:

```
ZoomInAndOut(iconStudent, 1.2, TimeSpan.FromSeconds(1.5), _
              TimeSpan.FromSeconds(0.7), 4)
```

Παράδειγμα 2:

Στο ίδιο παράδειγμα με το παραπάνω, αλλά κάνοντας το εφέ να επαναλαμβάνεται συνεχώς:

VB:

```
ZoomInAndOut(iconStudent, 1.2, TimeSpan.FromSeconds(1.5), _
              TimeSpan.FromSeconds(0.7), 0, repeatForever:=True)
```

ZoomInAndOut(element, zoomFactor, pointOrigin, duration, repeatTimes, Optional repeatForever)

Εκτελεί το zoom-in and out εφέ στο **element**, με συντελεστή μεγέθυνσης **zoomFactor**, με κεντρικό σημείο το **pointOrigin** και σε χρόνο **duration**. Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορές σύμφωνα με την **repeatTimes** εκτός και αν θέσουμε σαφώς την **repeatForever**, οπότε θα επαναλαμβάνεται συνεχώς.

Σύνταξη:

VB :

```
Public Shared Sub ZoomInAndOut(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    pointOrigin As Point,  
    duration As TimeSpan,  
    repeatTimes As Integer,  
    Optional repeatForever As Boolean  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single
Ο συντελεστής μεγέθυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η μεγέθυνση θα ξεκινήσει από το κανονικό μέγεθος (1.0) προς το 1.2. Δηλαδή, μεγέθυνση 20%. Αν θέσουμε την τιμή 0.4 τότε θα ξεκινήσει από το κανονικό μέγεθος (1.0) προς το 0.4, δηλαδή σμίκρυνση κατά 60%.

pointOrigin

Τύπος: System.Windows.Point
Καθορίζει το κεντρικό σημείο από το οποίο γίνεται η μεγέθυνση (pivot). Η τιμή εκτείνεται στο διάστημα από το [0,0] έως το [1,1], όπου το [0,0] υποδηλώνει το πάνω αριστερό μέρος του element ενώ το [1,1] υποδηλώνει το κάτω δεξί. Για παράδειγμα, η τιμή (0.5, 1) υποδηλώνει το κάτω μεσαίο μέρος. Η default τιμή είναι (0.5, 0.5), δηλαδή το μέσον του.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

repeatTimes

Τύπος: System.Integer

Καθορίζει πόσες φορές θα επαναλαμβάνεται το εφέ (δηλαδή, πόσες φορές θα κάνει την μεγέθυνση/σμίκρυνση) . Για παράδειγμα, η τιμή 4 σημαίνει 4 φορές θα μεγαλώνει/μικραίνει. Τεχνικά μιλώντας, πόσες φορές θα κάνει τον κύκλο zoom in->out->in.

Optional repeatForever

Τύπος: System.Boolean

Αν είναι True, τότε το εφέ θα εκτελείται συνεχώς. Σε αυτήν την περίπτωση αγνοείται η τιμή της παραμέτρου repeatTimes. Η default τιμή είναι False.

Ο μόνος τρόπος να σταματήσει είναι να καλέσουμε τη μέθοδο StopZoomInAndOutAnimation.

Παράδειγμα 1:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να μεγεθύνεται/σμικραίνει 4 φορές κατά 20% σε χρόνο 0.7 secs. Ορίζουμε ως κεντρικό σημείο μεγέθυνσης το κάτω μεσαίο μέρος (θέση (0.5, 1)).

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"
        Width="64" Height="64"/>
```

VB:

```
ZoomInAndOut(iconStudent, 1.2, new Point(0.5, 1), TimeSpan.FromSeconds(0.7), 4)
```

Παράδειγμα 2:

Στο ίδιο παράδειγμα με το παραπάνω, αλλά κάνοντας το εφέ να επαναλαμβάνεται συνεχώς:

VB:

```
ZoomInAndOut(iconStudent, 1.2, new Point(0.5, 1), _
              TimeSpan.FromSeconds(0.7), 0, repeatForever:=True)
```

ZoomInAndOut(element, zoomFactor, pointOrigin, beginTime, duration, repeatTimes, Optional repeatForever)

Εκτελεί το zoom-in and out εφέ στο **element**, με συντελεστή μεγέθυνσης **zoomFactor**, με κεντρικό σημείο το **pointOrigin** και σε χρόνο **duration**. Το εφέ επαναλαμβάνεται συγκεκριμένο αριθμό φορών σύμφωνα με την **repeatTimes** εκτός και αν θέσουμε σαφώς την **repeatForever**, οπότε θα επαναλαμβάνεται συνεχώς.

Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomInAndOut(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    pointOrigin As Point,  
    beginTime As TimeSpan,  
    duration As TimeSpan,  
    repeatTimes As Integer,  
    Optional repeatForever As Boolean  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single

Ο συντελεστής μεγέθυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η μεγέθυνση θα ξεκινήσει από το κανονικό μέγεθος (1.0) προς το 1.2. Δηλαδή, μεγέθυνση 20%. Αν θέσουμε την τιμή 0.4 τότε θα ξεκινήσει από το κανονικό μέγεθος (1.0) προς το 0.4, δηλαδή σμίκρυνση κατά 60%.

pointOrigin

Τύπος: System.Windows.Point

Καθορίζει το κεντρικό σημείο από το οποίο γίνεται η μεγέθυνση (pivot). Η τιμή εκτείνεται στο διάστημα από το [0,0] έως το [1,1], όπου το [0,0] υποδηλώνει το πάνω αριστερό μέρος του element ενώ το [1,1] υποδηλώνει το κάτω δεξί. Για παράδειγμα, η τιμή (0.5, 1) υποδηλώνει το κάτω μεσαίο μέρος. Η default τιμή είναι (0.5, 0.5), δηλαδή το μέσον του.

beginTime

Τύπος: System.TimeSpan

Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

duration

Τύπος: `System.TimeSpan`

Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή `TimeSpan`. Για παράδειγμα, η τιμή `TimeSpan.FromSeconds(0.7)` σημαίνει 0.7 secs.

repeatTimes

Τύπος: `System.Integer`

Καθορίζει πόσες φορές θα επαναλαμβάνεται το εφέ (δηλαδή, πόσες φορές θα κάνει την μεγέθυνση/σμίκρυνση). Για παράδειγμα, η τιμή 4 σημαίνει 4 φορές θα μεγαλώνει/μικραίνει. Τεχνικά μιλώντας, πόσες φορές θα κάνει τον κύκλο zoom in->out->in.

Optional repeatForever

Τύπος: `System.Boolean`

Αν είναι `True`, τότε το εφέ θα εκτελείται συνεχώς. Σε αυτήν την περίπτωση αγνοείται η τιμή της παραμέτρου `repeatTimes`. Η default τιμή είναι `False`. Ο μόνος τρόπος να σταματήσει είναι να καλέσουμε τη μέθοδο `StopZoomInAndOutAnimation`.

Παράδειγμα 1:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα `Image` element, με όνομα `iconStudent`. Στον VB κώδικα, κάνουμε την εικόνα να μεγεθύνεται/σμικραίνει 4 φορές κατά 20% σε χρόνο 0.7 secs. Ορίζουμε ως κεντρικό σημείο μεγέθυνσης το κάτω μεσαίο μέρος (θέση (0.5, 1)). Το εφέ ξεκινάει με χρονοκαθυστέρηση 1.5 secs.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"
        Width="64" Height="64"/>
```

VB:

```
ZoomInAndOut(iconStudent, 1.2, new Point(0.5, 1), _
              TimeSpan.FromSeconds(1.5), TimeSpan.FromSeconds(0.7), 4)
```

Παράδειγμα 2:

Στο ίδιο παράδειγμα με το παραπάνω, αλλά κάνοντας το εφέ να επαναλαμβάνεται συνεχώς:

VB:

```
ZoomInAndOut(iconStudent, 1.2, new Point(0.5, 1), _
              TimeSpan.FromSeconds(1.5), TimeSpan.FromSeconds(0.7), 0, _
              repeatForever:=True)
```

StopZoomInAndOutAnimation(element)

Σταματάει το zoom-in and out animation στο **element**.

Σύνταξη:

VB :

```
Public Shared Sub StopZoomInAndOutAnimation(  
    element As FrameworkElement  
)
```

Παράμετροι:

eLement

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο ήδη τρέχει το εφέ.

Παρατηρήσεις:

Δεν πρόκειται να προκληθεί σφάλμα αν καλέσουμε τη μέθοδο χωρίς να τρέχει σε αυτό το zoom-in and out animation.

Παράδειγμα:

VB:

```
StopZoomInAndOutAnimation(iconStudent)
```

ZoomOut – Μέθοδοι

Όνομα	Περιγραφή
ZoomOut (element As FrameworkElement, zoomFactor As Single, animationSpeed As AnimationSpeedEnum)	Εκτελεί το zoom-out εφέ στο element , με συντελεστή σμίκρυνσης zoomFactor και με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum .
ZoomOut (element As FrameworkElement, zoomFactor As Single, beginTime As TimeSpan, animationSpeed As AnimationSpeedEnum)	Εκτελεί το zoom-out εφέ στο element , με συντελεστή σμίκρυνσης zoomFactor και με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum . Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime .
ZoomOut (element As FrameworkElement, zoomFactor As Single, pointOrigin As Point, animationSpeed As AnimationSpeedEnum)	Εκτελεί το zoom-out εφέ στο element , με συντελεστή σμίκρυνσης zoomFactor , με κεντρικό σημείο το pointOrigin και με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum .
ZoomOut (element As FrameworkElement, zoomFactor As Single, pointOrigin As Point, beginTime As TimeSpan, animationSpeed As AnimationSpeedEnum)	Εκτελεί το zoom-out εφέ στο element , με συντελεστή σμίκρυνσης zoomFactor , με κεντρικό σημείο το pointOrigin και με ταχύτητα που προσδιορίζεται από την απαρίθμηση AnimationSpeedEnum . Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime .
ZoomOut (element As FrameworkElement, zoomFactor As Single, duration As TimeSpan)	Εκτελεί το zoom-out εφέ στο element , με συντελεστή σμίκρυνσης zoomFactor και σε χρόνο duration .
ZoomOut (element As FrameworkElement, zoomFactor As Single, beginTime As TimeSpan, duration As TimeSpan)	Εκτελεί το zoom-out εφέ στο element , με συντελεστή σμίκρυνσης zoomFactor και σε χρόνο duration . Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime .

ZoomOut (element As FrameworkElement, zoomFactor As Single, pointOrigin As Point, duration As TimeSpan)	Εκτελεί το zoom-out εφέ στο element , με συντελεστή σμίκρυνσης zoomFactor , με κεντρικό σημείο το pointOrigin και σε χρόνο duration .
ZoomOut (element As FrameworkElement, zoomFactor As Single, pointOrigin As Point, beginTime As TimeSpan, duration As TimeSpan)	Εκτελεί το zoom-out εφέ στο element , με συντελεστή σμίκρυνσης zoomFactor , με κεντρικό σημείο το pointOrigin και σε χρόνο duration . Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την beginTime .

ZoomOut(element, zoomFactor, animationSpeed)

Εκτελεί το zoom-out εφέ στο **element**, με συντελεστή σμίκρυνσης **zoomFactor** και με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomOut(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    animationSpeed As AnimationSpeedEnum  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single
Ο συντελεστής σμίκρυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η σμίκρυνση θα ξεκινήσει από το 1.2 προς το 0 . Δηλαδή, σμίκρυνση 120%.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να σμικρυνθεί 120%.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"  
    Width="64" Height="64"/>
```

VB:

```
ZoomOut(iconStudent, 1.2, AnimationSpeedEnum.Normal)
```

ZoomOut(element, zoomFactor, beginTime, animationSpeed)

Εκτελεί το zoom-out εφέ στο **element**, με συντελεστή σμίκρυνσης **zoomFactor** και με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**. Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomOut(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    beginTime As TimeSpan,  
    animationSpeed As AnimationSpeedEnum  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single
Ο συντελεστής σμίκρυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η σμίκρυνση θα ξεκινήσει από το 1.2 προς το 0 . Δηλαδή, σμίκρυνση 120%.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να σμικρυνθεί 120%. Το εφέ ξεκινάει με χρονοκαθυστέρηση 1.5 secs.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"  
    Width="64" Height="64"/>
```

VB:

```
ZoomOut(iconStudent, 1.2, TimeSpan.FromSeconds(1.5), AnimationSpeedEnum.Normal)
```

ZoomOut(element, zoomFactor, pointOrigin, animationSpeed)

Εκτελεί το zoom-out εφέ στο **element**, με συντελεστή σμίκρυνσης **zoomFactor**, με κεντρικό σημείο το **pointOrigin** και με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomOut(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    pointOrigin As Point,  
    animationSpeed As AnimationSpeedEnum  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single
Ο συντελεστής σμίκρυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η σμίκρυνση θα ξεκινήσει από το 1.2 προς το 0. Δηλαδή, σμίκρυνση 120%.

pointOrigin

Τύπος: System.Windows.Point
Καθορίζει το κεντρικό σημείο από το οποίο γίνεται η σμίκρυνση (pivot). Η τιμή εκτείνεται στο διάστημα από το [0,0] έως το [1,1], όπου το [0,0] υποδηλώνει το πάνω αριστερό μέρος του element ενώ το [1,1] υποδηλώνει το κάτω δεξί. Για παράδειγμα, η τιμή (0.5, 1) υποδηλώνει το κάτω μεσαίο μέρος. Η default τιμή είναι (0.5, 0.5), δηλαδή το μέσον του.

animationSpeed

Τύπος: AnimationUtilities.AnimationSpeedEnum
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης AnimationSpeedEnum. Για παράδειγμα, AnimationSpeedEnum.Fast.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να σμικρυνθεί 120%. Ορίζουμε ως κεντρικό σημείο σμίκρυνσης το πάνω μεσαίο μέρος (θέση (0.5, 0)).

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"  
      Width="64" Height="64"/>
```

VB:

```
ZoomOut(iconStudent, 1.2, new Point(0.5, 0), AnimationSpeedEnum.Normal)
```

ZoomOut(element, zoomFactor, pointOrigin, beginTime, animationSpeed)

Εκτελεί το zoom-out εφέ στο **element**, με συντελεστή σμίκρυνσης **zoomFactor**, με κεντρικό σημείο το **pointOrigin** και με ταχύτητα που προσδιορίζεται από την απαρίθμηση **AnimationSpeedEnum**. Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomOut(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    pointOrigin As Point,  
    beginTime As TimeSpan,  
    animationSpeed As AnimationSpeedEnum  
)
```

Παράμετροι:

element

Τύπος: `System.Windows.FrameworkElement`
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: `System.Single`
Ο συντελεστής σμίκρυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η σμίκρυνση θα ξεκινήσει από το 1.2 προς το 0. Δηλαδή, σμίκρυνση 120%.

pointOrigin

Τύπος: `System.Windows.Point`
Καθορίζει το κεντρικό σημείο από το οποίο γίνεται η σμίκρυνση (pivot). Η τιμή εκτείνεται στο διάστημα από το [0,0] έως το [1,1], όπου το [0,0] υποδηλώνει το πάνω αριστερό μέρος του element ενώ το [1,1] υποδηλώνει το κάτω δεξί. Για παράδειγμα, η τιμή (0.5, 1) υποδηλώνει το κάτω μεσαίο μέρος. Η default τιμή είναι (0.5, 0.5), δηλαδή το μέσον του.

beginTime

Τύπος: `System.TimeSpan`
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή `TimeSpan.FromSeconds(1.5)` σημαίνει καθυστέρηση έναρξης 1.5 secs.

animationSpeed

Τύπος: `AnimationUtilities.AnimationSpeedEnum`
Η ταχύτητα του εφέ, σύμφωνα με κάποιες προκαθορισμένες τιμές της απαρίθμησης `AnimationSpeedEnum`. Για παράδειγμα, `AnimationSpeedEnum.Fast`.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να σμικρυνθεί 120%. Ορίζουμε ως κεντρικό σημείο σμίκρυνσης το πάνω μεσαίο μέρος (θέση (0.5, 0)). Το εφέ ξεκινάει με χρονοκαθυστέρηση 1.5 secs.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"
      Width="64" Height="64"/>
```

VB:

```
ZoomOut(iconStudent, 1.2, new Point(0.5, 0), TimeSpan.FromSeconds(1.5), _
      AnimationSpeedEnum.Normal)
```

ZoomOut(element, zoomFactor, duration)

Εκτελεί το zoom-out εφέ στο **element**, με συντελεστή σμίκρυνσης **zoomFactor** και σε χρόνο **duration**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomOut(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    duration As TimeSpan  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single
Ο συντελεστής σμίκρυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η σμίκρυνση θα ξεκινήσει από το 1.2 προς το 0 . Δηλαδή, σμίκρυνση 120%.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να σμικρυνθεί 120% σε χρόνο 0.7 secs..

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"  
    Width="64" Height="64"/>
```

VB:

```
ZoomOut(iconStudent, 1.2, TimeSpan.FromSeconds(0.7))
```


ZoomOut(element, zoomFactor, beginTime, duration)

Εκτελεί το zoom-out εφέ στο **element**, με συντελεστή σμίκρυνσης **zoomFactor** και σε χρόνο **duration**. Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomOut(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    beginTime As TimeSpan,  
    duration As TimeSpan  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single
Ο συντελεστής σμίκρυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η σμίκρυνση θα ξεκινήσει από το 1.2 προς το 0 . Δηλαδή, σμίκρυνση 120%.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να σμικρυνθεί 120% σε χρόνο 0.7 secs. Το εφέ ξεκινάει με χρονοκαθυστέρηση 1.5 secs.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"  
    Width="64" Height="64"/>
```

VB:

```
ZoomOut(iconStudent, 1.2, TimeSpan.FromSeconds(1.5), TimeSpan.FromSeconds(0.7))
```

ZoomOut(element, zoomFactor, pointOrigin, duration)

Εκτελεί το zoom-out εφέ στο **element**, με συντελεστή σμίκρυνσης **zoomFactor**, με κεντρικό σημείο το **pointOrigin** και σε χρόνο **duration**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomOut(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    pointOrigin As Point,  
    duration As TimeSpan  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single
Ο συντελεστής σμίκρυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η σμίκρυνση θα ξεκινήσει από το 1.2 προς το 0 . Δηλαδή, σμίκρυνση 120%.

pointOrigin

Τύπος: System.Windows.Point
Καθορίζει το κεντρικό σημείο από το οποίο γίνεται η σμίκρυνση (pivot). Η τιμή εκτείνεται στο διάστημα από το [0,0] έως το [1,1], όπου το [0,0] υποδηλώνει το πάνω αριστερό μέρος του element ενώ το [1,1] υποδηλώνει το κάτω δεξί. Για παράδειγμα, η τιμή (0.5, 1) υποδηλώνει το κάτω μεσαίο μέρος. Η default τιμή είναι (0.5, 0.5), δηλαδή το μέσον του.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να σμικρυνθεί 120% σε χρόνο 0.7 secs. Ορίζουμε ως κεντρικό σημείο σμίκρυνσης το πάνω μεσαίο μέρος (θέση (0.5, 0)).

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"  
      Width="64" Height="64"/>
```

VB:

```
ZoomOut(iconStudent, 1.2, new Point(0.5, 0), TimeSpan.FromSeconds(0.7))
```

ZoomOut(element, zoomFactor, pointOrigin, beginTime, duration)

Εκτελεί το zoom-out εφέ στο **element**, με συντελεστή σμίκρυνσης **zoomFactor**, με κεντρικό σημείο το **pointOrigin** και σε χρόνο **duration**. Το εφέ ξεκινάει με μία καθυστέρηση που προσδιορίζεται από την **beginTime**.

Σύνταξη:

VB :

```
Public Shared Sub ZoomOut(  
    element As FrameworkElement,  
    zoomFactor As Single,  
    pointOrigin As Point,  
    beginTime As TimeSpan,  
    duration As TimeSpan  
)
```

Παράμετροι:

element

Τύπος: System.Windows.FrameworkElement
Το element στο οποίο θα εφαρμοστεί το εφέ.

zoomFactor

Τύπος: System.Single
Ο συντελεστής σμίκρυνσης. Για παράδειγμα, η τιμή 1.2 σημαίνει ότι η σμίκρυνση θα ξεκινήσει από το 1.2 προς το 0 . Δηλαδή, σμίκρυνση 120%.

pointOrigin

Τύπος: System.Windows.Point
Καθορίζει το κεντρικό σημείο από το οποίο γίνεται η σμίκρυνση (pivot). Η τιμή εκτείνεται στο διάστημα από το [0,0] έως το [1,1], όπου το [0,0] υποδηλώνει το πάνω αριστερό μέρος του element ενώ το [1,1] υποδηλώνει το κάτω δεξί. Για παράδειγμα, η τιμή (0.5, 1) υποδηλώνει το κάτω μεσαίο μέρος. Η default τιμή είναι (0.5, 0.5), δηλαδή το μέσον του.

beginTime

Τύπος: System.TimeSpan
Προσδιορίζει με πόση χρονική καθυστέρηση θα ξεκινήσει το εφέ. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(1.5) σημαίνει καθυστέρηση έναρξης 1.5 secs.

duration

Τύπος: System.TimeSpan
Καθορίζει την χρονική διάρκεια του εφέ (κι εμμέσως την ταχύτητά του), με μία δομή TimeSpan. Για παράδειγμα, η τιμή TimeSpan.FromSeconds(0.7) σημαίνει 0.7 secs.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει ένα Image element, με όνομα iconStudent. Στον VB κώδικα, κάνουμε την εικόνα να σμικρυνθεί 120% σε χρόνο 0.7 secs. Ορίζουμε ως κεντρικό σημείο σμίκρυνσης το πάνω μεσαίο μέρος (θέση (0.5, 0)). Το εφέ ξεκινάει με χρονοκαθυστέρηση 1.5 secs.

XAML:

```
<Image Name="iconStudent" Source="Images/Student64x64.png"
        Width="64" Height="64"/>
```

VB:

```
ZoomOut(iconStudent, 1.2, new Point(0.5, 0), TimeSpan.FromSeconds(1.5), _
        TimeSpan.FromSeconds(0.7))
```



DataUtilities

Περιλαμβάνει μία συλλογή από στατικές μεθόδους σχετικές με **διαχείριση δεδομένων, ιδιαιτέρως για το διάβασμα/γράψιμο από/σε database πεδία.**

Σύνταξη:

VB:

```
Public Class DataUtilities
```

Ο παρακάτω πίνακας περιέχει τις διαθέσιμες **κατηγορίες**:

Κατηγορία	Περιγραφή
GetDBValue μέθοδοι	<p>Περιλαμβάνει στατικές μεθόδους για διάβασμα τιμής από ένα non or nullable database πεδίο.</p> <p>Χρήσιμο, επειδή ιδιαίτερα σε nullable database πεδία, μπορεί να υπάρχει η τιμή DBNull.Value οπότε πρέπει να επιστραφεί η τιμή Nothing. Οι μέθοδοι αναλαμβάνουν τον σχετικό έλεγχο αυτόματα, ανάλογα με τον τύπο του πεδίου.</p>
SetDBValue μέθοδοι	<p>Περιλαμβάνει στατικές μεθόδους για αποθήκευση τιμής σε ένα nullable database πεδίο.</p> <p>Χρήσιμο, επειδή αν η τιμή που πρόκειται να αποθηκευτεί στο database πεδίο είναι Nothing τότε πρέπει να τεθεί σε αυτό η τιμή DBNull.Value. Οι μέθοδοι αναλαμβάνουν τον σχετικό έλεγχο αυτόματα, ανάλογα με τον τύπο του πεδίου.</p>
Έλεγχος σύνδεσης σε Βάση	<p>Περιλαμβάνει στατικές μεθόδους για ένα fade-in and move-in εφέ.</p> <p>Χρήσιμο, αν επιθυμούμε να εμφανίσουμε σταδιακά ένα framework element, το οποίο ταυτόχρονα μετακινείται από μία offset θέση στην κανονική του θέση.</p>

Απαριθμήσεις

Όνομα	Περιγραφή
OleDbConnectionResultEnum	Περιλαμβάνει προκαθορισμένες τιμές για το αποτέλεσμα μίας OLEDB σύνδεσης (OLEDB connection).
SqlServerConnectionResultEnum	Περιλαμβάνει προκαθορισμένες τιμές για το αποτέλεσμα μίας SQL Server σύνδεσης (SQL Server connection).

GetDBValue – Μέθοδοι

Όνομα	Περιγραφή
GetDBValueBoolean (value As Object) As Boolean	Επιστρέφει μία τιμή Boolean από την value . Χρησιμοποιείται για διάβασμα τιμής από non-nullable database πεδίο .
GetDBValueBooleanNullable (value As Object) As Nullable(Of Boolean)	Επιστρέφει μία τιμή Boolean από την value ή Nothing αν η value είναι DBNull.Value . Χρησιμοποιείται για διάβασμα τιμής από nullable database πεδίο .
GetDBValueByte (value As Object) As Byte	Επιστρέφει μία τιμή Byte από την value . Χρησιμοποιείται για διάβασμα τιμής από non-nullable database πεδίο .
GetDBValueByteNullable (value As Object) As Nullable(Of Byte)	Επιστρέφει μία τιμή Byte από την value ή Nothing αν η value είναι DBNull.Value . Χρησιμοποιείται για διάβασμα τιμής από nullable database πεδίο .
GetDBValueShort (value As Object) As Short	Επιστρέφει μία τιμή Short ή Int16 από την value . Χρησιμοποιείται για διάβασμα τιμής από non-nullable database πεδίο .
GetDBValueShortNullable (value As Object) As Nullable(Of Short)	Επιστρέφει μία τιμή Short ή Int16 από την value ή Nothing αν η value είναι DBNull.Value . Χρησιμοποιείται για διάβασμα τιμής από nullable database πεδίο .
GetDBValueInteger (value As Object) As Integer	Επιστρέφει μία τιμή Integer ή Int32 από την value . Χρησιμοποιείται για διάβασμα τιμής από non-nullable database πεδίο .
GetDBValueIntegerNullable (value As Object) As Nullable(Of Integer)	Επιστρέφει μία τιμή Integer ή Int32 από την value ή Nothing αν η value είναι DBNull.Value . Χρησιμοποιείται για διάβασμα τιμής από nullable database πεδίο .
GetDBValueLong (value As Object) As Long	Επιστρέφει μία τιμή Long ή Int64 από την value . Χρησιμοποιείται για διάβασμα τιμής από non-nullable database πεδίο .

GetDBValueLongNullable (value As Object) As Nullable(Of Long)	Επιστρέφει μία τιμή Long ή Int64 από την value ή Nothing αν η value είναι DBNull.Value . Χρησιμοποιείται για διάβασμα τιμής από nullable database πεδίο .
GetDBValueSingle (value As Object) As Single	Επιστρέφει μία τιμή Single από την value . Χρησιμοποιείται για διάβασμα τιμής από non-nullable database πεδίο .
GetDBValueSingleNullable (value As Object) As Nullable(Of Single)	Επιστρέφει μία τιμή Single από την value ή Nothing αν η value είναι DBNull.Value . Χρησιμοποιείται για διάβασμα τιμής από nullable database πεδίο .
GetDBValueDouble (value As Object) As Double	Επιστρέφει μία τιμή Double από την value . Χρησιμοποιείται για διάβασμα τιμής από non-nullable database πεδίο .
GetDBValueDoubleNullable (value As Object) As Nullable(Of Double)	Επιστρέφει μία τιμή Double από την value ή Nothing αν η value είναι DBNull.Value . Χρησιμοποιείται για διάβασμα τιμής από nullable database πεδίο .
GetDBValueDecimal (value As Object) As Decimal	Επιστρέφει μία τιμή Decimal από την value . Χρησιμοποιείται για διάβασμα τιμής από non-nullable database πεδίο .
GetDBValueDecimalNullable (value As Object) As Nullable(Of Decimal)	Επιστρέφει μία τιμή Decimal από την value ή Nothing αν η value είναι DBNull.Value . Χρησιμοποιείται για διάβασμα τιμής από nullable database πεδίο .
GetDBValueDateTime (value As Object) As DateTime	Επιστρέφει μία τιμή DateTime από την value . Χρησιμοποιείται για διάβασμα τιμής από non-nullable database πεδίο .
GetDBValueDateTimeNullable (value As Object) As Nullable(Of DateTime)	Επιστρέφει μία τιμή DateTime από την value ή Nothing αν η value είναι DBNull.Value . Χρησιμοποιείται για διάβασμα τιμής από nullable database πεδίο .
GetDBValueDate (value As Object) As Date	Επιστρέφει μία τιμή Date από την value . Χρησιμοποιείται για διάβασμα τιμής από non-nullable database πεδίο .

GetDBValueDateNullable(value As Object) As Nullable(Of Date)

Επιστρέφει μία τιμή **Date** από την **value** ή **Nothing** αν η **value** είναι **DBNull.Value**.
Χρησιμοποιείται για διάβασμα τιμής από **nullable database πεδίο**.

GetDBValueBoolean(value)

Επιστρέφει μία τιμή **Boolean** από την **value**. Χρησιμοποιείται για διάβασμα τιμής από **non-nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueBoolean(  
    value As Object  
)  
    As Boolean
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε Boolean.

Επιστρεφόμενη τιμή

Τύπος: System.Boolean

Αφού γίνει η μετατροπή, επιστρέφει την τιμή Boolean.

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου non-nullable Boolean. Αλλιώς, θα προκληθεί σφάλμα (exceprtion).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε τα στοιχεία των μαθητών από μία Access database σε μία συλλογή ObservableCollection(Of Student), όπου Student είναι μία data class που κρατάει τα δεδομένα ενός μαθητή. Το **διάβασμα** των μαθητών γίνεται με ένα OleDbDataReader. Σε εκείνο το σημείο, χρησιμοποιούμε τις **GetDBValueXXX μεθόδους**, ανάλογα με τον τύπο του πεδίου.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
Public Shared Function GetStudentsFromAccess(connectionString As String)  
    As ObservableCollection(Of Student)  
  
    Dim obsStudents As New ObservableCollection(Of Student)
```

```

Dim student As Student

Using conn As New OleDbConnection(connectionString)

    conn.Open()

    Using cmd As OleDbCommand = conn.CreateCommand()

        Dim strSQL = "Select STUDENTS.ID, STUDENTS.FIRSTNAME,
STUDENTS.LASTNAME, STUDENTS.[REGISTRATION DATE], STUDENTS.GRADE,
STUDENTS.CLASS_ID, CLASSES.CLASS, STUDENTS.DIVISION_ID,
DIVISIONS.DIVISION, STUDENTS.[IS ACTIVE]
                        From (STUDENTS INNER Join CLASSES ON
STUDENTS.CLASS_ID = CLASSES.ID_CLASS)
                                INNER Join DIVISIONS ON
STUDENTS.DIVISION_ID = DIVISIONS.ID_DIVISION
                                Order by
STUDENTS.LASTNAME,STUDENTS.FIRSTNAME;"

        cmd.CommandText = strSQL
        cmd.CommandType = System.Data.CommandType.Text

        Dim reader As OleDbDataReader = cmd.ExecuteReader

        While reader.Read()

            student = New Student

            'Σε επαγγελματική εφαρμογή πρέπει να τσεκάρουμε αν η τιμή που
            'διαβάζουμε ενδέχεται να είναι null.
            'ΑΥΤΟ ΕΠΙΤΥΓΧΑΝΟΥΝ ΟΙ ΧΡΗΣΙΜΕΣ ΣΤΑΤΙΚΕΣ ΜΕΘΟΔΟΙ GetDBValueXXX
            'ΓΙΑ ΔΙΑΒΑΣΜΑ ΤΙΜΩΝ ΑΠΟ ΠΕΔΙΑ ΤΗΣ ΒΑΣΗΣ.

            With student

                .ID = GetDBValueLong(reader("ID")) 'no null.
                .FirstName = reader("FIRSTNAME").ToString
                .LastName = reader("LASTNAME").ToString
                .RegistrationDate = _
                GetDBValueDateNullable(reader("REGISTRATION DATE")) 'is nullable.
                .Grade = _
                GetDBValueSingleNullable(reader("GRADE")) 'is nullable.
                .ClassID = _
                GetDBValueLongNullable(reader("CLASS_ID")) 'is nullable.
                .ClassName = reader("CLASS").ToString
                .DivisionID = _
                GetDBValueLongNullable(reader("DIVISION_ID")) 'is nullable.
                .DivisionName = reader("DIVISION").ToString
                .IsActive = _
                GetDBValueBoolean(reader("IS ACTIVE")) 'no null.

            End With

            obsStudents.Add(student)

        End While

        reader.Close()

    End Using

```

End Using

Return obsStudents

End Function

GetDBValueBooleanNullable(value)

Επιστρέφει μία τιμή **Boolean** από την **value** ή **Nothing** αν η **value** είναι **DBNull.Value**. Χρησιμοποιείται για διάβασμα τιμής από **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueBooleanNullable(  
    value As Object  
)  
    As Nullable(Of Boolean)
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε Nullable(of Boolean).

Επιστρεφόμενη τιμή

Τύπος: System.Nullable(Of Boolean)

Αφού γίνει η μετατροπή, επιστρέφει την τιμή Nullable(of Boolean).

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου nullable Boolean. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε μία τιμή nullable Boolean από ένα database πεδίο με τη χρήση ενός OleDbDataReader, με όνομα reader . Η τιμή αποθηκεύεται σε μία ιδιότητα της κλάσης Student.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
...  
  
Dim reader As OleDbDataReader = cmd.ExecuteReader  
  
While reader.Read()
```

```
        student = New Student
        With student
            ....
            .IsActive = _
            GetDBValueBooleanNullable(reader("IS ACTIVE")) 'nullable Boolean.
        End With
        ...
    End While
    reader.Close()
    ...
```

- Για ένα πλήρες παράδειγμα χρήσης των μεθόδων GetDBValueXXX, δείτε στο λήμμα [GetDBValueBoolean\(value\)](#).

GetDBValueByte(value)

Επιστρέφει μία τιμή **Byte** από την **value**. Χρησιμοποιείται για διάβασμα τιμής από **non-nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueByte(  
    value As Object  
)  
    As Byte
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε Byte.

Επιστρεφόμενη τιμή

Τύπος: System.Byte

Αφού γίνει η μετατροπή, επιστρέφει την τιμή Byte.

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου non-nullable Byte. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε μία τιμή non-nullable Byte από ένα database πεδίο με τη χρήση ενός OleDbDataReader, με όνομα reader. Η τιμή αποθηκεύεται σε μία ιδιότητα της κλάσης Student.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
...  
  
Dim reader As OleDbDataReader = cmd.ExecuteReader  
  
While reader.Read()  
  
    student = New Student
```

```
        With student
            ....
            .Grade = GetDBValueByte(reader("GRADE")) 'non-nullable Byte.
        End With
        ...
    End While
    reader.Close()
    ...
```

- Για ένα πλήρες παράδειγμα χρήσης των μεθόδων GetDBValueXXX, δείτε στο λήμμα [GetDBValueBoolean\(value\)](#).

GetDBValueByteNullable(value)

Επιστρέφει μία τιμή **Byte** από την **value** ή **Nothing** αν η **value** είναι **DBNull.Value**. Χρησιμοποιείται για διάβασμα τιμής από **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueByteNullable(  
    value As Object  
)  
    As Nullable(Of Byte)
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε Nullable(of Byte).

Επιστρεφόμενη τιμή

Τύπος: System.Nullable(Of Byte)

Αφού γίνει η μετατροπή, επιστρέφει την τιμή Nullable(of Byte).

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου nullable Byte. Αλλιώς, θα προκληθεί σφάλμα (excerption).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε μία τιμή nullable Byte από ένα database πεδίο με τη χρήση ενός OleDbDataReader, με όνομα reader. Η τιμή αποθηκεύεται σε μία ιδιότητα της κλάσης Student.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
...  
  
Dim reader As OleDbDataReader = cmd.ExecuteReader  
  
While reader.Read()
```

```
        student = New Student
        With student
            ....
            .Grade = GetDBValueByteNullable(reader("GRADE")) 'nullable
            Byte.
        End With
        ...
    End While
    reader.Close()
    ...
```

- Για ένα πλήρες παράδειγμα χρήσης των μεθόδων GetDBValueXXX, δείτε στο λήμμα [GetDBValueBoolean\(value\)](#).

GetDBValueShort(value)

Επιστρέφει μία τιμή **Short/Int16** από την **value**. Χρησιμοποιείται για διάβασμα τιμής από **non-nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueShort(  
    value As Object  
)  
    As Short
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε Short/Int16.

Επιστρεφόμενη τιμή

Τύπος: System.Short

Αφού γίνει η μετατροπή, επιστρέφει την τιμή Short/Int16.

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου non-nullable Short/Int16. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε μία τιμή non-nullable Short/Int16 από ένα database πεδίο με τη χρήση ενός OleDbDataReader, με όνομα reader. Η τιμή αποθηκεύεται σε μία ιδιότητα της κλάσης Student.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
...  
  
Dim reader As OleDbDataReader = cmd.ExecuteReader  
  
While reader.Read()
```

```
        student = New Student
        With student
            ....
            .Grade = GetDBValueShort(reader("GRADE")) 'non-nullable
                Short/Int16.
        End With
        ...
    End While
    reader.Close()
    ...
```

- Για ένα πλήρες παράδειγμα χρήσης των μεθόδων GetDBValueXXX, δείτε στο λήμμα [GetDBValueBoolean\(value\)](#).

GetDBValueShortNullable(value)

Επιστρέφει μία τιμή **Short/Int16** από την **value** ή **Nothing** αν η **value** είναι **DBNull.Value**. Χρησιμοποιείται για διάβασμα τιμής από **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueShortNullable(  
    value As Object  
)  
    As Nullable(Of Short)
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε Nullable(of Short).

Επιστρεφόμενη τιμή

Τύπος: System.Nullable(Of Short)

Αφού γίνει η μετατροπή, επιστρέφει την τιμή Nullable(of Short).

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου nullable Short/Int16. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε μία τιμή nullable Short/Int16 από ένα database πεδίο με τη χρήση ενός OleDbDataReader, με όνομα reader. Η τιμή αποθηκεύεται σε μία ιδιότητα της κλάσης Student.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
...  
  
Dim reader As OleDbDataReader = cmd.ExecuteReader  
  
While reader.Read()
```

```
        student = New Student
        With student
            ....
            .Grade = GetDBValueShortNullable(reader("GRADE")) 'nullable
                       Short/Int16.
        End With
        ...
    End While
    reader.Close()
    ...
```

- Για ένα πλήρες παράδειγμα χρήσης των μεθόδων GetDBValueXXX, δείτε στο λήμμα [GetDBValueBoolean\(value\)](#).

GetDBValueInteger(value)

Επιστρέφει μία τιμή **Integer/Int32** από την **value**. Χρησιμοποιείται για διάβασμα τιμής από **non-nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueInteger(  
    value As Object  
)  
    As Integer
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε Integer/Int32.

Επιστρεφόμενη τιμή

Τύπος: System.Integer

Αφού γίνει η μετατροπή, επιστρέφει την τιμή Integer/Int32.

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου non-nullable Integer/Int32. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε μία τιμή non-nullable Integer/Int32 από ένα database πεδίο με τη χρήση ενός OleDbDataReader, με όνομα reader. Η τιμή αποθηκεύεται σε μία ιδιότητα της κλάσης Student.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
...  
  
Dim reader As OleDbDataReader = cmd.ExecuteReader  
  
While reader.Read()
```

```
        student = New Student
        With student
            ....
            .Absences = _
            GetDBValueInteger(reader("ABSENCES")) 'non-nullable Integer/Int32.
        End With
        ...
    End While
    reader.Close()
    ...
```

- Για ένα πλήρες παράδειγμα χρήσης των μεθόδων GetDBValueXXX, δείτε στο λήμμα [GetDBValueBoolean\(value\)](#).

GetDBValueIntegerNullable(value)

Επιστρέφει μία τιμή **Integer/Int32** από την **value** ή **Nothing** αν η **value** είναι **DBNull.Value**. Χρησιμοποιείται για διάβασμα τιμής από **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueIntegerNullable(  
    value As Object  
)  
    As Nullable(Of Integer)
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε Nullable(of Integer).

Επιστρεφόμενη τιμή

Τύπος: System.Nullable(Of Integer)

Αφού γίνει η μετατροπή, επιστρέφει την τιμή Nullable(of Integer).

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου nullable Integer/Int32. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε μία τιμή nullable Integer/Int32 από ένα database πεδίο με τη χρήση ενός OleDbDataReader, με όνομα reader. Η τιμή αποθηκεύεται σε μία ιδιότητα της κλάσης Student.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
...  
  
Dim reader As OleDbDataReader = cmd.ExecuteReader  
  
While reader.Read()
```

```
        student = New Student
        With student
            ....
            .Absences = _
GetDBValueIntegerNullable(reader("ABSENCES")) 'nullable Integer/Int32.
        End With
        ...
    End While
    reader.Close()
    ...
```

- Για ένα πλήρες παράδειγμα χρήσης των μεθόδων GetDBValueXXX, δείτε στο λήμμα [GetDBValueBoolean\(value\)](#).

GetDBValueLong(value)

Επιστρέφει μία τιμή **Long/Int64** από την **value**. Χρησιμοποιείται για διάβαση τιμής από **non-nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueLong(  
    value As Object  
)  
    As Long
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε Long/Int64.

Επιστρεφόμενη τιμή

Τύπος: System.Long

Αφού γίνει η μετατροπή, επιστρέφει την τιμή Long/Int64.

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου non-nullable Long/Int64. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε μία τιμή non-nullable Long/Int64 από ένα database πεδίο με τη χρήση ενός OleDbDataReader, με όνομα reader. Η τιμή αποθηκεύεται σε μία ιδιότητα της κλάσης Student.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
...  
  
Dim reader As OleDbDataReader = cmd.ExecuteReader  
  
While reader.Read()
```

```
        student = New Student
    With student
        ....
        .ID = GetDBValueLong(reader("StudentID")) 'non-nullable
        Long/Int64.
    End With
    ...
End While
reader.Close()
...
```

- Για ένα πλήρες παράδειγμα χρήσης των μεθόδων GetDBValueXXX, δείτε στο λήμμα [GetDBValueBoolean\(value\)](#).

GetDBValueLongNullable(value)

Επιστρέφει μία τιμή **Long/Int64** από την **value** ή **Nothing** αν η **value** είναι **DBNull.Value**. Χρησιμοποιείται για διάβασμα τιμής από **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueLongNullable(  
    value As Object  
)  
    As Nullable(Of Long)
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε Nullable(of Long).

Επιστρεφόμενη τιμή

Τύπος: System.Nullable(Of Long)

Αφού γίνει η μετατροπή, επιστρέφει την τιμή Nullable(of Long).

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου nullable Long/Int64. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε μία τιμή nullable Long/Int64 από ένα database πεδίο με τη χρήση ενός OleDbDataReader, με όνομα reader. Η τιμή αποθηκεύεται σε μία ιδιότητα της κλάσης Student.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
...  
  
Dim reader As OleDbDataReader = cmd.ExecuteReader  
  
While reader.Read()
```

```
        student = New Student
        With student
            ....
            .Absences = _
            GetDBValueLongNullable(reader("ABSENCES")) 'nullable Long/Int64.
        End With
        ...
    End While
    reader.Close()
    ...
```

- Για ένα πλήρες παράδειγμα χρήσης των μεθόδων GetDBValueXXX, δείτε στο λήμμα [GetDBValueBoolean\(value\)](#).

GetDBValueSingle(value)

Επιστρέφει μία τιμή **Single** από την **value**. Χρησιμοποιείται για διάβασμα τιμής από **non-nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueSingle(  
    value As Object  
)  
    As Single
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε Single.

Επιστρεφόμενη τιμή

Τύπος: System.Single

Αφού γίνει η μετατροπή, επιστρέφει την τιμή Single .

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου non-nullable Single. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε μία τιμή non-nullable Single από ένα database πεδίο με τη χρήση ενός OleDbDataReader, με όνομα reader. Η τιμή αποθηκεύεται σε μία ιδιότητα της κλάσης Student.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
...  
  
Dim reader As OleDbDataReader = cmd.ExecuteReader  
  
While reader.Read()  
  
    student = New Student
```

```
        With student
            ....
            .Grade = GetDBValueSingle(reader("GRADE")) 'non-nullable
            Single.
        End With
        ...
    End While
    reader.Close()
    ...
```

- Για ένα πλήρες παράδειγμα χρήσης των μεθόδων GetDBValueXXX, δείτε στο λήμμα [GetDBValueBoolean\(value\)](#).

GetDBValueSingleNullable(value)

Επιστρέφει μία τιμή **Single** από την **value** ή **Nothing** αν η **value** είναι **DBNull.Value**. Χρησιμοποιείται για διάβασμα τιμής από **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueSingleNullable(  
    value As Object  
)  
    As Nullable(Of Single)
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε Nullable(of Single).

Επιστρεφόμενη τιμή

Τύπος: System.Nullable(Of Single)

Αφού γίνει η μετατροπή, επιστρέφει την τιμή Nullable(of Single).

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου nullable Single. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε μία τιμή nullable Single από ένα database πεδίο με τη χρήση ενός OleDbDataReader, με όνομα reader. Η τιμή αποθηκεύεται σε μία ιδιότητα της κλάσης Student.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
...  
  
Dim reader As OleDbDataReader = cmd.ExecuteReader  
  
While reader.Read()
```

```
        student = New Student
        With student
            ....
            .Grade = GetDBValueSingleNullable(reader("GRADE")) 'nullable
            Single.
        End With
        ...
    End While
    reader.Close()
    ...
```

- Για ένα πλήρες παράδειγμα χρήσης των μεθόδων GetDBValueXXX, δείτε στο λήμμα [GetDBValueBoolean\(value\)](#).

GetDBValueDouble(value)

Επιστρέφει μία τιμή **Double** από την **value**. Χρησιμοποιείται για διάβασμα τιμής από **non-nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueDouble(  
    value As Object  
)  
    As Double
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε Double.

Επιστρεφόμενη τιμή

Τύπος: System.Double

Αφού γίνει η μετατροπή, επιστρέφει την τιμή Double .

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου non-nullable Double. Αλλιώς, θα προκληθεί σφάλμα (excerption).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε μία τιμή non-nullable Double από ένα database πεδίο με τη χρήση ενός OleDbDataReader, με όνομα reader. Η τιμή αποθηκεύεται σε μία ιδιότητα της κλάσης Student.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
...  
  
Dim reader As OleDbDataReader = cmd.ExecuteReader  
  
While reader.Read()
```

```
        student = New Student
        With student
            ....
            .Grade = GetDBValueDouble(reader("GRADE")) 'non-nullable
            Double.
        End With
        ...
    End While
    reader.Close()
    ...
```

- Για ένα πλήρες παράδειγμα χρήσης των μεθόδων GetDBValueXXX, δείτε στο λήμμα [GetDBValueBoolean\(value\)](#).

GetDBValueDoubleNullable(value)

Επιστρέφει μία τιμή **Double** από την **value** ή **Nothing** αν η **value** είναι **DBNull.Value**. Χρησιμοποιείται για διάβασμα τιμής από **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueDoubleNullable(  
    value As Object  
)  
    As Nullable(Of Double)
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε Nullable(of Double).

Επιστρεφόμενη τιμή

Τύπος: System.Nullable(Of Double)

Αφού γίνει η μετατροπή, επιστρέφει την τιμή Nullable(of Double).

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου nullable Double. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε μία τιμή nullable Double από ένα database πεδίο με τη χρήση ενός OleDbDataReader, με όνομα reader. Η τιμή αποθηκεύεται σε μία ιδιότητα της κλάσης Student.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
...  
  
Dim reader As OleDbDataReader = cmd.ExecuteReader  
  
While reader.Read()
```

```
        student = New Student
        With student
            ....
            .Grade = GetDBValueDoubleNullable(reader("GRADE")) 'nullable
            Double.
        End With
        ...
    End While
    reader.Close()
    ...
```

- Για ένα πλήρες παράδειγμα χρήσης των μεθόδων GetDBValueXXX, δείτε στο λήμμα [GetDBValueBoolean\(value\)](#).

GetDBValueDecimal(value)

Επιστρέφει μία τιμή **Decimal** από την **value**. Χρησιμοποιείται για διάβασμα τιμής από **non-nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueDecimal(  
    value As Object  
)  
    As Decimal
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε Decimal.

Επιστρεφόμενη τιμή

Τύπος: System.Decimal

Αφού γίνει η μετατροπή, επιστρέφει την τιμή Decimal.

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου non-nullable Decimal. Αλλιώς, θα προκληθεί σφάλμα (exceprtion).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε μία τιμή non-nullable Decimal από ένα database πεδίο με τη χρήση ενός OleDbDataReader, με όνομα reader. Η τιμή αποθηκεύεται σε μία ιδιότητα της κλάσης Student.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
...  
  
Dim reader As OleDbDataReader = cmd.ExecuteReader  
  
While reader.Read()
```

```
        student = New Student
        With student
            ....
            .Fees = GetDBValueDecimal(reader("FEES")) 'non-nullable
                Decimal.
        End With
        ...
    End While
    reader.Close()
    ...
```

- Για ένα πλήρες παράδειγμα χρήσης των μεθόδων GetDBValueXXX, δείτε στο λήμμα [GetDBValueBoolean\(value\)](#).

GetDBValueDecimalNullable(value)

Επιστρέφει μία τιμή **Decimal** από την **value** ή **Nothing** αν η **value** είναι **DBNull.Value**. Χρησιμοποιείται για διάβασμα τιμής από **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueDecimalNullable(  
    value As Object  
)  
    As Nullable(Of Decimal)
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε Nullable(of Decimal).

Επιστρεφόμενη τιμή

Τύπος: System.Nullable(Of Decimal)

Αφού γίνει η μετατροπή, επιστρέφει την τιμή Nullable(of Decimal).

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου nullable Decimal. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε μία τιμή nullable Decimal από ένα database πεδίο με τη χρήση ενός OleDbDataReader, με όνομα reader. Η τιμή αποθηκεύεται σε μία ιδιότητα της κλάσης Student.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
...  
  
Dim reader As OleDbDataReader = cmd.ExecuteReader  
  
While reader.Read()
```

```
        student = New Student
        With student
            ....
            .Fees = GetDBValueDecimalNullable(reader("FEES")) 'nullable
            Decimal.
        End With
        ...
    End While
    reader.Close()
    ...
```

- Για ένα πλήρες παράδειγμα χρήσης των μεθόδων GetDBValueXXX, δείτε στο λήμμα [GetDBValueBoolean\(value\)](#).

GetDBValueDateTime(value)

Επιστρέφει μία τιμή **DateTime** από την **value**. Χρησιμοποιείται για διάβασμα τιμής από **non-nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueDateTime(  
    value As Object  
)  
    As DateTime
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε DateTime.

Επιστρεφόμενη τιμή

Τύπος: System.DateTime

Αφού γίνει η μετατροπή, επιστρέφει την τιμή DateTime.

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου non-nullable DateTime. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε μία τιμή non-nullable DateTime από ένα database πεδίο με τη χρήση ενός OleDbDataReader, με όνομα reader. Η τιμή αποθηκεύεται σε μία ιδιότητα της κλάσης Student.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
...  
  
Dim reader As OleDbDataReader = cmd.ExecuteReader  
  
While reader.Read()
```

```
        student = New Student
        With student
            ....
            .RegistrationDate = _
GetDBValueDateTime(reader("REGISTRATION DATE")) 'non-nullable DateTime.
        End With
        ...
    End While
    reader.Close()
    ...
```

- Για ένα πλήρες παράδειγμα χρήσης των μεθόδων GetDBValueXXX, δείτε στο λήμμα [GetDBValueBoolean\(value\)](#).

GetDBValueDateTimeNullable(value)

Επιστρέφει μία τιμή **DateTime** από την **value** ή **Nothing** αν η **value** είναι **DBNull.Value**. Χρησιμοποιείται για διάβασμα τιμής από **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueDateTimeNullable(  
    value As Object  
)  
    As Nullable(Of DateTime)
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε Nullable(of DateTime).

Επιστρεφόμενη τιμή

Τύπος: System.Nullable(Of DateTime)

Αφού γίνει η μετατροπή, επιστρέφει την τιμή Nullable(of DateTime).

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου nullable DateTime. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε μία τιμή nullable DateTime από ένα database πεδίο με τη χρήση ενός OleDbDataReader, με όνομα reader. Η τιμή αποθηκεύεται σε μία ιδιότητα της κλάσης Student.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
...  
  
Dim reader As OleDbDataReader = cmd.ExecuteReader  
  
While reader.Read()
```

```
        student = New Student
        With student
            ....
            .RegistrationDate = _
GetDBValueDateTimeNullable(reader("REGISTRATION DATE")) 'nullable DateTime.
        End With
        ...
    End While
    reader.Close()
    ...
```

- Για ένα πλήρες παράδειγμα χρήσης των μεθόδων GetDBValueXXX, δείτε στο λήμμα [GetDBValueBoolean\(value\)](#).

GetDBValueDate(value)

Επιστρέφει μία τιμή **Date** από την **value**. Χρησιμοποιείται για διάβασμα τιμής από **non-nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueDate(  
    value As Object  
)  
    As Date
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε Date.

Επιστρεφόμενη τιμή

Τύπος: System.Date

Αφού γίνει η μετατροπή, επιστρέφει την τιμή Date.

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου non-nullable Date. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε μία τιμή non-nullable Date από ένα database πεδίο με τη χρήση ενός OleDbDataReader, με όνομα reader. Η τιμή αποθηκεύεται σε μία ιδιότητα της κλάσης Student.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
...  
  
Dim reader As OleDbDataReader = cmd.ExecuteReader  
  
While reader.Read()  
  
    student = New Student
```

```
        With student
            ....
            .RegistrationDate = _
                GetDBValueDate(reader("REGISTRATION DATE")) 'non-nullable Date.
        End With
        ...
    End While
    reader.Close()
    ...
```

- Για ένα πλήρες παράδειγμα χρήσης των μεθόδων GetDBValueXXX, δείτε στο λήμμα [GetDBValueBoolean\(value\)](#).

GetDBValueDateNullable(value)

Επιστρέφει μία τιμή **Date** από την **value** ή **Nothing** αν η **value** είναι **DBNull.Value**. Χρησιμοποιείται για διάβασμα τιμής από **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function GetDBValueDateNullable(  
    value As Object  
)  
    As Nullable(Of Date)
```

Παράμετροι:

value

Τύπος: System.Object

Το αντικείμενο που περιέχει την τιμή από το database πεδίο και θα μετατραπεί σε Nullable(of Date).

Επιστρεφόμενη τιμή

Τύπος: System.Nullable(Of Date)

Αφού γίνει η μετατροπή, επιστρέφει την τιμή Nullable(of Date).

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο από το οποίο διαβάζεται η τιμή είναι τύπου nullable Datee. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Στο παρακάτω παράδειγμα, διαβάζουμε μία τιμή nullable Date από ένα database πεδίο με τη χρήση ενός OleDbDataReader, με όνομα reader. Η τιμή αποθηκεύεται σε μία ιδιότητα της κλάσης Student.

VB:

```
Imports System.Collections.ObjectModel  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities  
  
...  
  
Dim reader As OleDbDataReader = cmd.ExecuteReader  
  
While reader.Read()
```

```
        student = New Student
        With student
            ....
            .RegistrationDate = _
GetDBValueDateNullable(reader("REGISTRATION DATE")) 'nullable Date.
        End With
        ...
    End While
    reader.Close()
    ...
```

- Για ένα πλήρες παράδειγμα χρήσης των μεθόδων GetDBValueXXX, δείτε στο λήμμα [GetDBValueBoolean\(value\)](#).

SetDBValue - Μέθοδοι

Όνομα	Περιγραφή
SetDBValueBooleanNullable (value As Nullable(Of Boolean)) As Object	Επιστρέφει μία τιμή Boolean από την value ή DBNull.Value αν η value είναι Nothing . Χρησιμοποιείται για αποθήκευση τιμής σε nullable database πεδίο .
SetDBValueByteNullable (value As Nullable(Of Byte)) As Object	Επιστρέφει μία τιμή Byte από την value ή DBNull.Value αν η value είναι Nothing . Χρησιμοποιείται για αποθήκευση τιμής σε nullable database πεδίο .
SetDBValueShortNullable (value As Nullable(Of Short)) As Object	Επιστρέφει μία τιμή Short/Int16 από την value ή DBNull.Value αν η value είναι Nothing . Χρησιμοποιείται για αποθήκευση τιμής σε nullable database πεδίο .
SetDBValueIntegerNullable (value As Nullable(Of Integer)) As Object	Επιστρέφει μία τιμή Integer/Int32 από την value ή DBNull.Value αν η value είναι Nothing . Χρησιμοποιείται για αποθήκευση τιμής σε nullable database πεδίο .
SetDBValueLongNullable (value As Nullable(Of Long)) As Object	Επιστρέφει μία τιμή Long/Int64 από την value ή DBNull.Value αν η value είναι Nothing . Χρησιμοποιείται για αποθήκευση τιμής σε nullable database πεδίο .
SetDBValueSingleNullable (value As Nullable(Of Single)) As Object	Επιστρέφει μία τιμή Single από την value ή DBNull.Value αν η value είναι Nothing . Χρησιμοποιείται για αποθήκευση τιμής σε nullable database πεδίο .
SetDBValueDoubleNullable (value As Nullable(Of Double)) As Object	Επιστρέφει μία τιμή Double από την value ή DBNull.Value αν η value είναι Nothing . Χρησιμοποιείται για αποθήκευση τιμής σε nullable database πεδίο .
SetDBValueDecimalNullable (value As Nullable(Of Decimal)) As Object	Επιστρέφει μία τιμή Decimal από την value ή DBNull.Value αν η value είναι Nothing . Χρησιμοποιείται για αποθήκευση τιμής σε nullable database πεδίο .

SetDBValueDateTimeNullable (value As Nullable(Of DateTime)) As Object	Επιστρέφει μία τιμή DateTime από την value ή DBNull.Value αν η value είναι Nothing . Χρησιμοποιείται για αποθήκευση τιμής σε nullable database πεδίο .
SetDBValueDateNullable (value As Nullable(Of Date)) As Object	Επιστρέφει μία τιμή Date από την value ή DBNull.Value αν η value είναι Nothing . Χρησιμοποιείται για αποθήκευση τιμής σε nullable database πεδίο .

SetDBValueBooleanNullable(value)

Επιστρέφει μία τιμή **Boolean** από την **value** ή **DBNull.Value** αν η **value** είναι **Nothing**. Χρησιμοποιείται για αποθήκευση τιμής σε **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function SetDBValueBooleanNullable(  
    value As Nullable(Of Boolean)  
)  
    As Object
```

Παράμετροι:

value

Τύπος: System.Nullable(Of Boolean)
Η τιμή προς αποθήκευση στο database πεδίο.

Επιστρεφόμενη τιμή

Τύπος: System.Object
Αφού γίνει ο έλεγχος της *value*, επιστρέφει την τιμή Boolean ή DBNull.Value.

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο στο οποίο θα αποθηκευτεί η τιμή είναι τύπου nullable Boolean. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Στο παρακάτω παράδειγμα, δημιουργούμε ένα αντικείμενο Student που περιέχει τα στοιχεία ενός νέου μαθητή. Στη συνέχεια, καλούμε την ρουτίνα AddStudentInAccess όπου αποθηκεύει, με μία SQL Insert query, τον νέο μαθητή στη Βάση. Μάλιστα, βλέπουμε και πώς επιστρέφουμε το νέο (auto) ID που δημιουργεί ο μηχανισμός της Access για αυτόν τον νεοεισερχόμενο μαθητή. Στην αποθήκευση χρησιμοποιούμε τις **SetDBValueXXXNullable μεθόδους**, ανάλογα με τον τύπο του πεδίου.

VB:

```
Imports System.Data  
Imports System.Data.OleDb  
Imports Zeus.WPF.Classes.Utilities.DataUtilities
```

...

```
Dim newStudent As New Student  
Dim newStudentID As Long = -1
```

```

With newStudent
    .FirstName = "ΕΥΦΡΟΣΥΝΗ"
    .LastName = "ΔΙΑΜΑΝΤΟΠΟΥΛΟΥ"
    .RegistrationDate = Nothing
    .Grade = Nothing
    ...
    .IsActive = False
End With

AddStudentInAccess(connectionString, newStudent, newStudentID)

...

'Αποθήκευσε έναν νέο μαθητή στην Access database.
Public Shared Function AddStudentInAccess(connectionString As String, student
As Student, ByRef newStudentID As Long) As Boolean

    Dim strSQL As String = ""
    Dim cntAffected As Integer = 0

    Using conn As New OleDbConnection(connectionString)

        conn.Open()

        strSQL = "INSERT INTO Students (FIRSTNAME, LASTNAME, [REGISTRATION DATE],
GRADE, CLASS_ID, DIVISION_ID, [IS ACTIVE]) VALUES(?,?,?, ?, ?, ?, ?)"
        Dim strSQLIdentity As String = "Select @@Identity"

        Using cmdInsert As OleDbCommand = conn.CreateCommand

            With cmdInsert
                .CommandType = CommandType.Text
                .CommandText = strSQL
                .Parameters.AddWithValue("FIRSTNAME", student.FirstName)
                .Parameters.AddWithValue("LASTNAME", student.LastName)
                .Parameters.AddWithValue("REGISTRATION DATE", _
SetDBValueDateNullable(student.RegistrationDate)) 'nullable Date.
                .Parameters.AddWithValue("GRADE", _
SetDBValueSingleNullable(student.Grade)) 'nullable Single.
                .Parameters.AddWithValue("CLASS_ID", student.ClassID) 'no
null.
                .Parameters.AddWithValue("DIVISION_ID", student.DivisionID)
'no null.
                .Parameters.AddWithValue("IS ACTIVE", student.IsActive) 'no
null.
                Try
                    cntAffected = .ExecuteNonQuery()
                    If cntAffected = 0 Then Return False
                    'Πάρε το νέο ID.
                    .CommandText = strSQLIdentity
                    newStudentID = .ExecuteScalar
                Catch
                    Return False
                End Try
            End With

        End Using

    End Using
End Function

```


End Using

Return True

End Function

SetDBValueByteNullable(value)

Επιστρέφει μία τιμή **Byte** από την **value** ή **DBNull.Value** αν η **value** είναι **Nothing**. Χρησιμοποιείται για αποθήκευση τιμής σε **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function SetDBValueByteNullable(  
    value As Nullable(Of Byte)  
)  
    As Object
```

Παράμετροι:

value

Τύπος: System.Nullable(Of Byte)
Η τιμή προς αποθήκευση στο database πεδίο.

Επιστρεφόμενη τιμή

Τύπος: System.Object
Αφού γίνει ο έλεγχος της **value**, επιστρέφει την τιμή **Byte** ή **DBNull.Value**.

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο στο οποίο θα αποθηκευτεί η τιμή είναι τύπου nullable Byte. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Για ένα πλήρες παράδειγμα χρήσης των **SetDBValueXXXNullable μεθόδων** δείτε στο λήμμα [SetDBValueBooleanNullable](#).

SetDBValueShortNullable(value)

Επιστρέφει μία τιμή **Short/Int16** από την **value** ή **DBNull.Value** αν η **value** είναι **Nothing**. Χρησιμοποιείται για αποθήκευση τιμής σε **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function SetDBValueShortNullable(  
    value As Nullable(Of Short)  
)  
    As Object
```

Παράμετροι:

value

Τύπος: System.Nullable(Of Short)
Η τιμή προς αποθήκευση στο database πεδίο.

Επιστρεφόμενη τιμή

Τύπος: System.Object
Αφού γίνει ο έλεγχος της *value*, επιστρέφει την τιμή Short ή DBNull.Value.

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο στο οποίο θα αποθηκευτεί η τιμή είναι τύπου nullable Short/Int16. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Για ένα πλήρες παράδειγμα χρήσης των **SetDBValueXXXNullable μεθόδων** δείτε στο λήμμα [SetDBValueBooleanNullable](#).

SetDBValueIntegerNullable(value)

Επιστρέφει μία τιμή **Integer/Int32** από την **value** ή **DBNull.Value** αν η **value** είναι **Nothing**. Χρησιμοποιείται για αποθήκευση τιμής σε **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function SetDBValueIntegerNullable(  
    value As Nullable(Of Integer)  
)  
    As Object
```

Παράμετροι:

value

Τύπος: System.Nullable(Of Integer)
Η τιμή προς αποθήκευση στο database πεδίο.

Επιστρεφόμενη τιμή

Τύπος: System.Object
Αφού γίνει ο έλεγχος της *value*, επιστρέφει την τιμή Integer ή DBNull.Value.

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο στο οποίο θα αποθηκευτεί η τιμή είναι τύπου nullable Integer/Int32. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Για ένα πλήρες παράδειγμα χρήσης των **SetDBValueXXXNullable μεθόδων** δείτε στο λήμμα [SetDBValueBooleanNullable](#).

SetDBValueLongrNullable(value)

Επιστρέφει μία τιμή **Long/Int64** από την **value** ή **DBNull.Value** αν η **value** είναι **Nothing**. Χρησιμοποιείται για αποθήκευση τιμής σε **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function SetDBValueLongNullable(  
    value As Nullable(Of Long)  
)  
    As Object
```

Παράμετροι:

value

Τύπος: System.Nullable(Of Long)
Η τιμή προς αποθήκευση στο database πεδίο.

Επιστρεφόμενη τιμή

Τύπος: System.Object
Αφού γίνει ο έλεγχος της *value*, επιστρέφει την τιμή Long ή DBNull.Value.

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο στο οποίο θα αποθηκευτεί η τιμή είναι τύπου nullable Long/Int64. Αλλιώς, θα προκληθεί σφάλμα (excerption).

Παράδειγμα:

Για ένα πλήρες παράδειγμα χρήσης των **SetDBValueXXXNullable μεθόδων** δείτε στο λήμμα [SetDBValueBooleanNullable](#).

SetDBValueSingleNullable(value)

Επιστρέφει μία τιμή **Single** από την **value** ή **DBNull.Value** αν η **value** είναι **Nothing**. Χρησιμοποιείται για αποθήκευση τιμής σε **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function SetDBValueSingleNullable(  
    value As Nullable(Of Single)  
) As Object
```

Παράμετροι:

value

Τύπος: System.Nullable(Of Single)
Η τιμή προς αποθήκευση στο database πεδίο.

Επιστρεφόμενη τιμή

Τύπος: System.Object
Αφού γίνει ο έλεγχος της **value**, επιστρέφει την τιμή **Single** ή **DBNull.Value**.

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο στο οποίο θα αποθηκευτεί η τιμή είναι τύπου nullable Single. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Για ένα πλήρες παράδειγμα χρήσης των **SetDBValueXXXNullable μεθόδων** δείτε στο λήμμα [SetDBValueBooleanNullable](#).

SetDBValueDoubleNullable(value)

Επιστρέφει μία τιμή **Double** από την **value** ή **DBNull.Value** αν η **value** είναι **Nothing**. Χρησιμοποιείται για αποθήκευση τιμής σε **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function SetDBValueDoubleNullable(  
    value As Nullable(Of Double)  
)  
    As Object
```

Παράμετροι:

value

Τύπος: System.Nullable(Of Double)
Η τιμή προς αποθήκευση στο database πεδίο.

Επιστρεφόμενη τιμή

Τύπος: System.Object
Αφού γίνει ο έλεγχος της *value*, επιστρέφει την τιμή Double ή DBNull.Value.

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο στο οποίο θα αποθηκευτεί η τιμή είναι τύπου nullable Double. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Για ένα πλήρες παράδειγμα χρήσης των **SetDBValueXXXNullable μεθόδων** δείτε στο λήμμα [SetDBValueBooleanNullable](#).

SetDBValueDecimalNullable(value)

Επιστρέφει μία τιμή **Decimal** από την **value** ή **DBNull.Value** αν η **value** είναι **Nothing**. Χρησιμοποιείται για αποθήκευση τιμής σε **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function SetDBValueDecimalNullable(  
    value As Nullable(Of Decimal)  
)  
    As Object
```

Παράμετροι:

value

Τύπος: System.Nullable(Of Decimal)
Η τιμή προς αποθήκευση στο database πεδίο.

Επιστρεφόμενη τιμή

Τύπος: System.Object
Αφού γίνει ο έλεγχος της *value*, επιστρέφει την τιμή Decimal ή DBNull.Value.

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο στο οποίο θα αποθηκευτεί η τιμή είναι τύπου nullable Decimal. Αλλιώς, θα προκληθεί σφάλμα (excerption).

Παράδειγμα:

Για ένα πλήρες παράδειγμα χρήσης των **SetDBValueXXXNullable μεθόδων** δείτε στο λήμμα [SetDBValueBooleanNullable](#).

SetDBValueDateTimeNullable(value)

Επιστρέφει μία τιμή **DateTime** από την **value** ή **DBNull.Value** αν η **value** είναι **Nothing**. Χρησιμοποιείται για αποθήκευση τιμής σε **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function SetDBValueDateTimeNullable(  
    value As Nullable(Of DateTime)  
)  
    As Object
```

Παράμετροι:

value

Τύπος: System.Nullable(Of DateTime)
Η τιμή προς αποθήκευση στο database πεδίο.

Επιστρεφόμενη τιμή

Τύπος: System.Object
Αφού γίνει ο έλεγχος της **value**, επιστρέφει την τιμή **DateTime** ή **DBNull.Value**.

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο στο οποίο θα αποθηκευτεί η τιμή είναι τύπου nullable DateTime. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Για ένα πλήρες παράδειγμα χρήσης των **SetDBValueXXXNullable μεθόδων** δείτε στο λήμμα [SetDBValueBooleanNullable](#).

SetDBValueDateNullable(value)

Επιστρέφει μία τιμή **Date** από την **value** ή **DBNull.Value** αν η **value** είναι **Nothing**. Χρησιμοποιείται για αποθήκευση τιμής σε **nullable database πεδίο**.

Σύνταξη:

VB :

```
Public Shared Function SetDBValueDateNullable(  
    value As Nullable(Of Date)  
)  
    As Object
```

Παράμετροι:

value

Τύπος: System.Nullable(Of Date)
Η τιμή προς αποθήκευση στο database πεδίο.

Επιστρεφόμενη τιμή

Τύπος: System.Object
Αφού γίνει ο έλεγχος της *value*, επιστρέφει την τιμή Date ή DBNull.Value.

Παρατηρήσεις:

Ο προγραμματιστής πρέπει να γνωρίζει ότι το database πεδίο στο οποίο θα αποθηκευτεί η τιμή είναι τύπου nullable Date. Αλλιώς, θα προκληθεί σφάλμα (exception).

Παράδειγμα:

Για ένα πλήρες παράδειγμα χρήσης των **SetDBValueXXXNullable μεθόδων** δείτε στο λήμμα [SetDBValueBooleanNullable](#).

Έλεγχος σύνδεσης σε Βάση – Μέθοδοι

Όνομα	Περιγραφή
CheckOLEDBConnection (connectionString As String, pathDB As String) As OLEDBConnectionResultEnum	<p>Επιστρέφει μία τιμή κατάστασης από την απαρίθμηση OLEDBConnectionResultEnum όταν προσπαθεί να συνδεθεί στην Βάση με βάση το connectionString.</p> <p>Αφορά Βάσεις που συνδέονται με OLEDB driver (π.χ. Access).</p>
CheckSqlServerConnection (connectionString As String, pathDB As String) As SqlServerConnectionResultEnum	<p>Επιστρέφει μία τιμή κατάστασης από την απαρίθμηση SqlServerConnectionResultEnum όταν προσπαθεί να συνδεθεί στην Βάση με βάση το connectionString.</p> <p>Αφορά SQL Server database.</p>

CheckOLEDBConnection(connectionString, pathDB)

Επιστρέφει μία τιμή κατάστασης από την απαρίθμηση [OLEDBConnectionResultEnum](#) όταν προσπαθεί να συνδεθεί στην Βάση με βάση το **connectionString**. Αφορά Βάσεις που συνδέονται με OLEDB driver (π.χ. Access).

Σύνταξη:

VB :

```
Public Shared Function CheckOLEDBConnection(  
    connectionString As String,  
    pathDB As String  
)  
    As OLEDBConnectionResultEnum
```

Παράμετροι:

connectionString

Τύπος: System.String

Το αλφαριθμητικό σύνδεσης με τη Βάση.

PathDB

Τύπος: System.String

Το path name του αρχείου της Βάσης.

Επιστρεφόμενη τιμή

Τύπος: Zeus.WPF.Classes.DataUtilities.OLEDBConnectionResultEnum

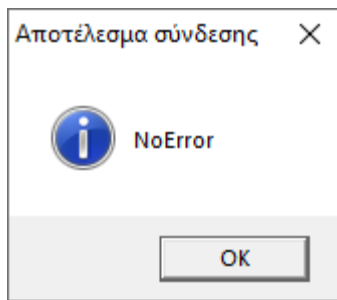
Το αποτέλεσμα της σύνδεσης.

Παράδειγμα:

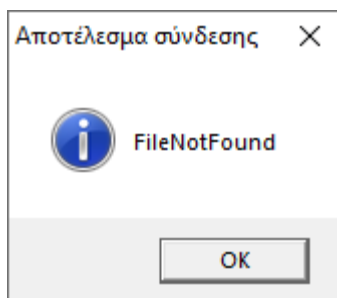
Έλεγχουμε τη σύνδεση του αρχείου Students.accdb που βρίσκεται στον τρέχον φάκελο. Επιστρέφει το αποτέλεσμα και το παρουσιάζουμε ως μήνυμα στον χρήστη:

VB:

```
Dim result As OLEDBConnectionResultEnum = _  
    CheckOLEDBConnection(connectionString, "Students.accdb")  
  
MessageBox.Show([Enum].GetName(GetType(OLEDBConnectionResultEnum), result), _  
    "Αποτέλεσμα σύνδεσης", MessageBoxButton.OK, _  
    MessageBoxImage.Information)
```



Αν το αρχείο δεν υπάρχει στο καθορισμένο path τότε θα εμφανιστεί το παρακάτω μήνυμα:



Φυσικά, τα μηνύματα εμφανίζονται με βάση το string που έχει δοθεί στην απαρίθμηση [OLEDBConnectionResultEnum](#). Μπορούμε, βέβαια, να κατασκευάσουμε τη δική μας μέθοδο που θα επιστρέφει ένα "καλύτερο" μήνυμα.

CheckSqlConnection(connectionString, pathDB)

Επιστρέφει μία τιμή κατάστασης από την απαρίθμηση [SqlConnectionResultEnum](#) όταν προσπαθεί να συνδεθεί στην Βάση με βάση το **connectionString**. Αφορά SQL Server database.

Σύνταξη:

VB :

```
Public Shared Function CheckSqlConnection(  
                                connectionString As String,  
                                pathDB As String  
                                )  
    As SqlConnectionResultEnum
```

Παράμετροι:

connectionString

Τύπος: System.String

Το αλφαριθμητικό σύνδεσης με τη Βάση.

PathDB

Τύπος: System.String

Το path name του αρχείου της Βάσης.

Επιστρεφόμενη τιμή

Τύπος: Zeus.WPF.Classes.DataUtilities.SqlServerConnectionResultEnum

Το αποτέλεσμα της σύνδεσης.

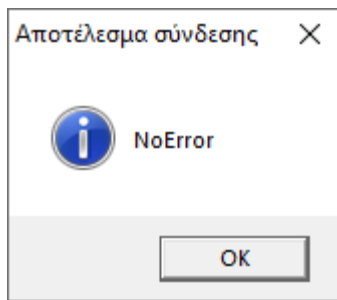
Παράδειγμα:

Έλεγχουμε τη σύνδεση του αρχείου Students.mdf που βρίσκεται στον τρέχον φάκελο.

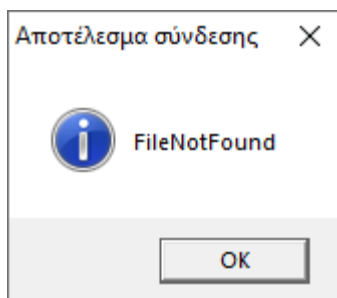
Επιστρέφει το αποτέλεσμα και το παρουσιάζουμε ως μήνυμα στον χρήστη:

VB:

```
Dim result As SqlConnectionResultEnum = _  
    CheckSqlConnection(connectionString, "Students.mdf")  
MessageBox.Show([Enum].GetName(GetType(SqlServerConnectionResultEnum), result), _  
    "Αποτέλεσμα σύνδεσης", MessageBoxButton.OK, _  
    MessageBoxImage.Information)
```



Αν το αρχείο δεν υπάρχει στο καθορισμένο path τότε θα εμφανιστεί το παρακάτω μήνυμα:



Φυσικά, τα μηνύματα εμφανίζονται με βάση το string που έχει δοθεί στην απαρίθμηση [SqlServerConnectionResultEnum](#). Μπορούμε, βέβαια, να κατασκευάσουμε τη δική μας μέθοδο που θα επιστρέφει ένα "καλύτερο" μήνυμα.



FileUtilities

Περιλαμβάνει μία συλλογή από στατικές μεθόδους σχετικές με **διαχείριση αρχείων/φακέλων και assemblies**.

Σύνταξη:

VB:

```
Public Class FileUtilities
```

Ο παρακάτω πίνακας περιέχει τις διαθέσιμες **κατηγορίες**:

Κατηγορία	Περιγραφή
File/Folder μέθοδοι	Περιλαμβάνει στατικές μεθόδους για διαχείριση αρχείων και φακέλων .
Assemblies μέθοδοι	Περιλαμβάνει στατικές μεθόδους για για διαχείριση/ανάγνωση πληροφοριών από assembly αρχεία (.dll και .exe) .

File/Folder - Μέθοδοι

Όνομα	Περιγραφή
GetFileExtension (filename As String) As String	Επιστρέφει την επέκταση ενός αρχείου (συμπεριλαμβανομένου του χαρακτήρα ".").
RemoveFileAttributeReadOnly (filename As String) As Boolean	Απομακρύνει την σημείωση "read-only" από τα attributes ενός αρχείου.

- Σημείωση: Η υποκατηγορία αυτή έχει χώρο για πολλές ακόμα μεθόδους. Ελπίζω, μελλοντικά να προσθέσω περισσότερες.

GetFileExtension(filename)

Επιστρέφει την επέκταση ενός αρχείου (συμπεριλαμβανομένου του χαρακτήρα period (".")).

Σύνταξη:

VB :

```
Public Shared Function GetFileExtension(  
    filename As String  
) As String
```

Παράμετροι:

filename

Τύπος: System.String
Το όνομα του αρχείου (path name).

Επιστρεφόμενη τιμή

Τύπος: System.String
Επιστρέφει την επέκταση (π.χ. ".jpg"). Συμπεριλαμβάνεται και ο χαρακτήρας period. Αν συμβεί σφάλμα, επιστρέφει String.Empty.

Παρατηρήσεις:

Η μέθοδος αποτελεί wrapper της System.IO.Path.GetExtension.

RemoveFileAttributeReadOnly(filename)

Απομακρύνει την σημείωση "read-only" από τα attributes ενός αρχείου.

Σύνταξη:

VB :

```
Public Shared Function RemoveFileAttributeReadOnly(  
    filename As String  
) As Boolean
```

Παράμετροι:

filename

Τύπος: System.String

Το όνομα του αρχείου (path name).

Επιστρεφόμενη τιμή

Τύπος: System.Boolean

Επιστρέφει True, αν η απομάκρυνση της σημείωσης "read-only" στα file attributes ήταν επιτυχής.

Assemblies – Μέθοδοι

Όνομα	Περιγραφή
GetTitleAndFileVersionFromAssembly (whatType As Type) As String	Επιστρέφει τον Τίτλο και την Έκδοση αρχείου από το assembly συγκεκριμένου τύπου (π.χ. Application).
GetAssemblyVersionFromAssembly (whatType As Type) As String	Επιστρέφει την Έκδοση assembly (major και minor) από το assembly συγκεκριμένου τύπου (π.χ. Application).
GetCompanyNameFromAssembly (whatType As Type) As String	Επιστρέφει το όνομα της Εταιρείας από το assembly συγκεκριμένου τύπου.

- Σημείωση: Η υποκατηγορία αυτή έχει χώρο για αρκετές ακόμα μεθόδους. Ελπίζω, μελλοντικά να προσθέσω περισσότερες.

GetTitleAndFileVersionFromAssembly(whatType)

Επιστρέφει τον Τίτλο και την Έκδοση αρχείου από το `assembly` συγκεκριμένου τύπου (π.χ. `Application`).

Σύνταξη:

VB :

```
Public Shared Function GetTitleAndFileVersionFromAssembly(  
    whatType As Type  
) As String
```

Παράμετροι:

whatType

Τύπος: `System.Type`
Ο τύπος του `assembly`.

Επιστρεφόμενη τιμή

Τύπος: `System.String`
Επιστρέφει τον τίτλο και την έκδοση αρχείου. Αν συμβεί σφάλμα, επιστρέφει `String.Empty`.

Παρατηρήσεις:

Η έκδοση του αρχείου εμφανίζεται στις ιδιότητες του αρχείου στον Windows Explorer.

Παράδειγμα:

Στην παρακάτω εικόνα, βλέπουμε τις **ιδιότητες** που έχουμε ορίσει για την εφαρμογής μας (`Utilities_Sample_Project`):

Assembly Information

Title: Utilities Sample Project

Description: Ένα sample project για την χρήση του component ZeusUtilities.

Company: Χρήστος Μουρατιδης - Zeus Software

Product: Utilities_Sample_Project

Copyright: Copyright © 2019

Trademark:

Assembly version: 1 0 0 0

File version: 1 3 0 0

GUID: 45682ede-ce08-457c-8761-88e208ea885a

Neutral language: (None)

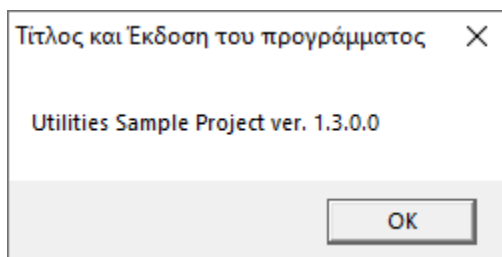
Make assembly COM-Visible

OK Cancel

Με την εντολή:

```
MessageBox.Show(GetTitleAndFileVersionFromAssembly(GetType(Utilities_Sample_Project  
.Application)), "Τίτλος και Έκδοση του προγράμματος")
```

θα εμφανιστεί το παρακάτω παράθυρο:



GetAssemblyVersionFromAssembly(whatType)

Επιστρέφει την Έκδοση assembly (major και minor) από το assembly συγκεκριμένου τύπου (π.χ. Application).

Σύνταξη:

VB :

```
Public Shared Function GetAssemblyVersionFromAssembly(  
    whatType As Type  
    )  
    As String
```

Παράμετροι:

whatType

Τύπος: System.Type
Ο τύπος του assembly.

Επιστρεφόμενη τιμή

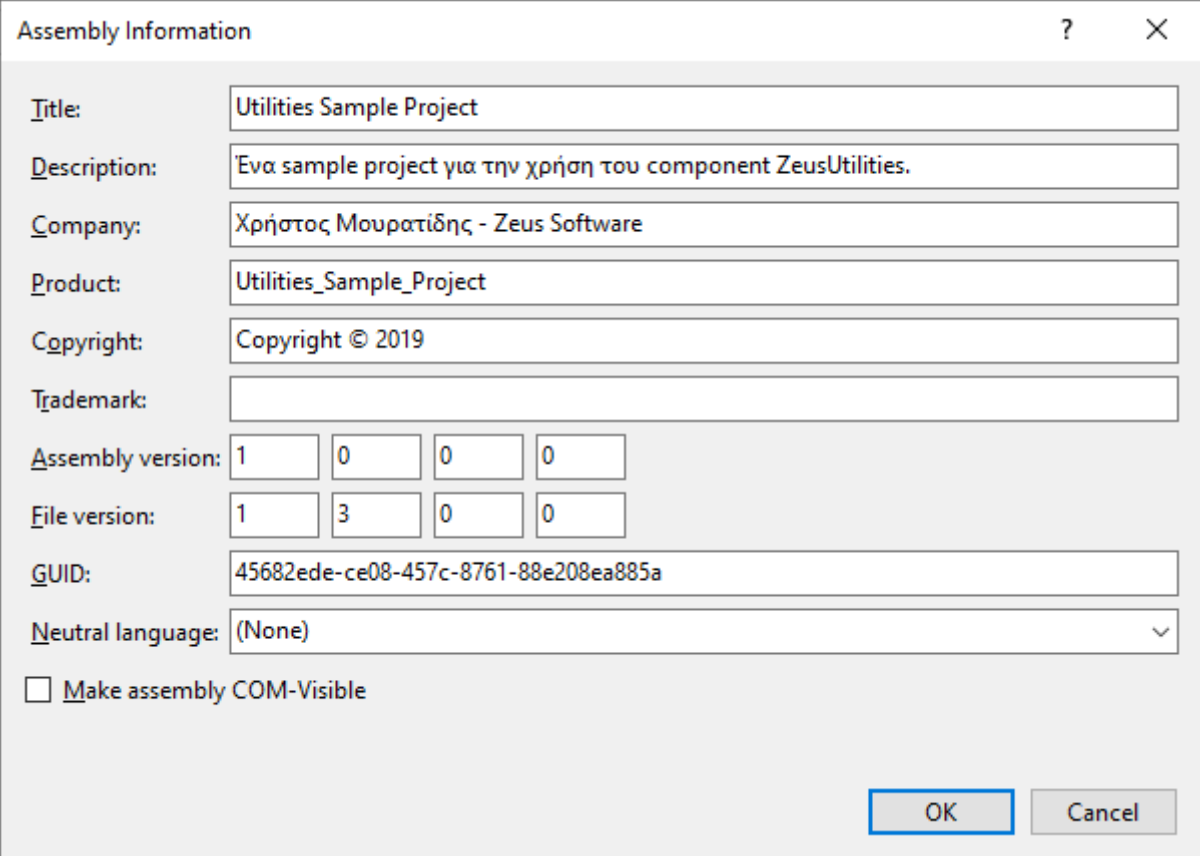
Τύπος: System.String
Επιστρέφει την έκδοση assembly. Αν συμβεί σφάλμα, επιστρέφει String.Empty.

Παρατηρήσεις:

Η έκδοση assembly χρησιμοποιείται από το CLR για να ξεχωρίζει και να βρίσκει τα assemblies με βάση τον αριθμό έκδοσης.

Παράδειγμα:

Στην παρακάτω εικόνα, βλέπουμε τις **ιδιότητες** που έχουμε ορίσει για την εφαρμογής μας (Utilities_Sample_Project):



Assembly Information

Title: Utilities Sample Project

Description: Ένα sample project για την χρήση του component ZeusUtilities.

Company: Χρήστος Μουρατιδης - Zeus Software

Product: Utilities_Sample_Project

Copyright: Copyright © 2019

Trademark:

Assembly version: 1 0 0 0

File version: 1 3 0 0

GUID: 45682ede-ce08-457c-8761-88e208ea885a

Neutral language: (None)

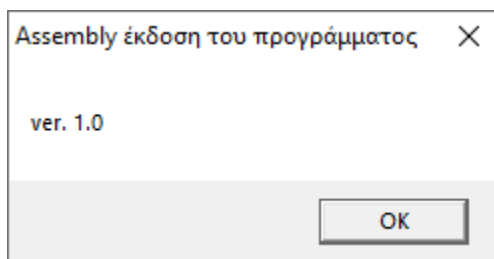
Make assembly COM-Visible

OK Cancel

Με την εντολή:

```
MessageBox.Show(GetAssemblyVersionFromAssembly(GetType(Utilities_Sample_Project.App  
lication)), "Assembly έκδοση του προγράμματος")
```

θα εμφανιστεί το παρακάτω παράθυρο:



GetCompanyNameFromAssembly(whatType)

Επιστρέφει το όνομα της Εταιρείας από το assembly συγκεκριμένου τύπου.

Σύνταξη:

VB :

```
Public Shared Function GetCompanyNameFromAssembly(  
    whatType As Type  
) As String
```

Παράμετροι:

whatType

Τύπος: System.Type
Ο τύπος του assembly.

Επιστρεφόμενη τιμή

Τύπος: System.String
Επιστρέφει το όνομα της Εταιρείας. Αν συμβεί σφάλμα, επιστρέφει String.Empty.

Παράδειγμα:

Στην παρακάτω εικόνα, βλέπουμε τις **ιδιότητες** που έχουμε ορίσει για την εφαρμογής μας (Utilities_Sample_Project):

Assembly Information

Title: Utilities Sample Project

Description: Ένα sample project για την χρήση του component ZeusUtilities.

Company: Χρήστος Μουρατίδης - Zeus Software

Product: Utilities_Sample_Project

Copyright: Copyright © 2019

Trademark:

Assembly version: 1 0 0 0

File version: 1 3 0 0

GUID: 45682ede-ce08-457c-8761-88e208ea885a

Neutral language: (None)

Make assembly COM-Visible

OK Cancel

Με την εντολή:

```
MessageBox.Show(GetCompanyNameFromAssembly(GetType(Utilities_Sample_Project.Application)), "Ανάπτυξη προγράμματος από:");
```

θα εμφανιστεί το παρακάτω παράθυρο:

Ανάπτυξη προγράμματος από: X

Χρήστος Μουρατίδης - Zeus Software

OK



FlowDocumentUtilities

Περιλαμβάνει μία συλλογή από στατικές μεθόδους σχετικές με **διαχείριση FlowDocuments και RichTextBoxes**.

Σύνταξη:

VB:

```
Public Class FlowDocumentUtilities
```

Ο παρακάτω πίνακας περιέχει τις διαθέσιμες **κατηγορίες**:

Κατηγορία	Περιγραφή
FlowDocument μέθοδοι	<p>Περιλαμβάνει στατικές μεθόδους για διαχείριση FlowDocuments (ορισμός κενού FlowDocument, αντιγραφή περιεχομένων σε άλλο, λήψη του μορφοποιημένου κειμένου, λήψη μόνο του plain text, κλπ).</p> <p>Να σημειώσουμε, ότι το περιεχόμενο ενός FlowDocument, μπορεί να <i>αναγνωστεί</i> μέσω ενός FlowDocument container όπως ένα FlowDocumentReader ή ένα FlowDocumentPageViewer.</p>
RichTextBox μέθοδοι	<p>Περιλαμβάνει στατικές μεθόδους για για διαχείριση RichTextBoxes (έλεγχος αν είναι κενό, καθαρισμός του περιεχομένου του, ορισμός του περιεχομένου του, αντιγραφή περιεχομένου του σε άλλο κλπ).</p> <p>Να σημειώσουμε, ότι το περιεχόμενο ενός RichTextBox είναι ένα αντικείμενο FlowDocument, <i>το οποίο μπορούμε να επεξεργαστούμε</i>.</p>

Shared ιδιότητες

Όνομα	Περιγραφή
NewEmptyFlowDocumentString	<p>Τύπος String.</p> <p>Καθορίζει το κενό περιεχόμενο ενός FlowDocument ως ένα xaml string.</p> <p>Παράδειγμα:</p> <pre>FlowDocumentUtilities.NewEmptyFlowDocumentString = "<FlowDocument FontFamily=""Arial"" FontStyle=""Normal"" FontWeight=""Normal"" FontSize=""12"" TextAlignment=""Left"" PagePadding=""5,0,5,0"" AllowDrop=""True"" xmlns=""http://schemas.microsoft.com/winfx/2006/xaml/pres entation"" />"</pre>
FlowDocumentGetPlainTextErrorString	<p>Τύπος String.</p> <p>Καθορίζει το μήνυμα λάθους όταν συμβεί σφάλμα κατά την ανάγνωση του plain text από το FlowDocument.</p> <p>Παράδειγμα:</p> <pre>FlowDocumentUtilities.FlowDocumentGetPlainTextErrorString = "Σφάλμα στην ανάγνωση του plain text"</pre>

FlowDocument - Μέθοδοι

Όνομα	Περιγραφή
FlowDocumentCopy (fromFlowDocument As FlowDocument, ByRef toFlowDocument As FlowDocument) As Boolean	Αντιγράφει το περιεχόμενο ενός FlowDocument σε ένα άλλο. Επιστρέφει True, αν η αντιγραφή ήταν επιτυχής. Το περιεχόμενο ενός FlowDocument είναι ένα μορφοποιημένο κείμενο (μορφής xaml string με root element το <FlowDocument>).
FlowDocumentGetContent (flowDocument As FlowDocument) As String	Επιστρέφει το μορφοποιημένο περιεχόμενο (xaml string) ενός FlowDocument.
FlowDocumentGetPlainText (flowDocument As FlowDocument) As String	Επιστρέφει το αμορφοποίητο κείμενο (plain text) ενός FlowDocument.
FlowDocumentSetContent (ByRef flowDocument As FlowDocument, xamlText As String) As Boolean	Θέτει το μορφοποιημένο κείμενο (xaml string) ως περιεχόμενο του FlowDocument. Επιστρέφει True, αν η απόθεση ήταν επιτυχής.

FlowDocumentCopy(fromFlowDocument, toFlowDocument)

Αντιγράφει το περιεχόμενο ενός FlowDocument σε ένα άλλο. Επιστρέφει True, αν η αντιγραφή ήταν επιτυχής.

Σύνταξη:

VB :

```
Public Shared Function FlowDocumentCopy(  
    fromFlowDocument As FlowDocument,  
    toFlowDocument As FlowDocument  
)  
    As Boolean
```

Παράμετροι:

fromFlowDocument

Τύπος: System.Windows.Documents.FlowDocument

Το αντικείμενο FlowDocument, το περιεχόμενο του οποίου θα αντιγραφεί (source).

toFlowDocument

Τύπος: System.Windows.Documents.FlowDocument

Το αντικείμενο FlowDocument, στο οποίο το περιεχόμενο θα επικολληθεί (target).

Επιστρεφόμενη τιμή

Τύπος: System.Boolean

Επιστρέφει True, αν η αντιγραφή ήταν επιτυχής.

Παρατηρήσεις:

Το περιεχόμενο ενός FlowDocument είναι ένα μορφοποιημένο κείμενο (μορφής xaml string με root element το <FlowDocument>).

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει σε ένα Grid δύο FlowDocuments. Το πρώτο FlowDocument, με όνομα flowDoc1, περιέχει πληροφορίες για τον Σωκράτη. Το δεύτερο FlowDocument, με όνομα flowDoc2, είναι κενό (απλά περιέχει ένα μήνυμα ότι δεν έχει υπάρχει περιεχόμενο). Σε ένα StackPanel, δεξιά, έχουμε τοποθετήσει μία στοίβα από buttons. Το πρώτο κάνει καθάρισμα του flowDoc2, το δεύτερο κάνει αντιγραφή του περιεχομένου του flowDoc1 στο flowDoc2 και το τρίτο παίρνει το αμορφοποιημένο κείμενο (plain text) του flowDoc1.

Να σημειώσουμε ότι καθένα FlowDocument, βρίσκεται μέσα στο πολύ δυναμικό container, το FlowDocumentReader.

XAML:

```

<Grid>

  <Grid.RowDefinitions >
    <RowDefinition Height="Auto"/>
    <RowDefinition />
  </Grid.RowDefinitions>

  <Grid.ColumnDefinitions >
    <ColumnDefinition Width="*"/>
    <ColumnDefinition />
  </Grid.ColumnDefinitions>

  <Border BorderBrush="Black" BorderThickness="1" CornerRadius="5" Margin="5">

    <FlowDocumentReader Name="flowReader1" Margin="5" Background="Yellow"
      IsPrintEnabled="True" IsScrollViewEnabled="True">

      <FlowDocument Name="flowDoc1">

        <Section>

          <Paragraph FontSize="16" FontWeight="ExtraBold"
            Foreground="Blue" >
            Socrates
          </Paragraph>

          <Paragraph FontSize="12" >

            <Figure VerticalAnchor="ParagraphTop"
              HorizontalAnchor="PageLeft"
              VerticalOffset="0"
              HorizontalOffset="2" >

              <Paragraph >
                <InlineUIContainer >
                  <Image Source="Images/Socrates.jpg"
                    Width="100" />
                </InlineUIContainer >
              </Paragraph>

            </Figure>

            Socrates lived during the time of the transition from the
            height of the <Bold>Athenian hegemony</Bold> to its
            decline with the defeat by Sparta and its allies in the
            Peloponnesian War. At a time when Athens sought to
            stabilize and recover from its humiliating defeat, the
            Athenian public may have been entertaining doubts about
            democracy as an efficient form of government.
            Socrates appears to have been a critic of democracy, and
            some scholars[who?] interpret his trial as an expression
            of political infighting.

          </Paragraph>

        </Section>

        <Paragraph FontSize="14" >

```

Xenophon and Plato agree that Socrates had an opportunity to escape, as his followers were able to bribe the prison guards. He chose to stay for several reasons.

```

        </Paragraph>
    </FlowDocument>
</FlowDocumentReader>
</Border>
<Border Grid.Row="1" BorderBrush="Black" BorderThickness="1" CornerRadius="5"
    Margin="5">
    <FlowDocumentReader Name="flowReader2" Margin="5" Background="Yellow"
        IsPrintEnabled="True" IsScrollViewEnabled="True">
        <FlowDocument Name="flowDoc2">
            <Paragraph FontSize="16" Foreground="Orange" >
                Δεν υπάρχει περιεχόμενο.
            </Paragraph>
        </FlowDocument>
    </FlowDocumentReader>
</Border>
<StackPanel Grid.Column="1" VerticalAlignment="Center" Margin="5"
    ButtonBase.Click="btnFlowDocuments_Click" >
    <Button Name="btnClearDocument2"
        Content="Καθάρισμα του 2ου document" Padding="5" />
    <Button Name="btnCopyDocument1ToDocument2"
        Content="Αντιγραφή στο 2ο document"
        Margin="0,10,0,0" Padding="5" />
    <Button Name="btnGetPlainTextOfDocument1"
        Content="Πάρε το plain text από το 1ο document"
        Margin="0,10,0,0" Padding="5" />
</StackPanel>
</Grid>

```

VB:

```

Imports Zeus.WPF.Classes.Utilities.FlowDocumentUtilities

Private Sub btnFlowDocuments_Click(sender As Object, e As RoutedEventArgs)

    Dim btn As Button = TryCast(e.OriginalSource, Button)

    If btn IsNot Nothing Then

        'Καθάρισμα του 2ου FlowDocument.
        If btn Is btnClearDocument2 Then

```



```
        If FlowDocumentSetContent(flowDoc2, GetNoNoContentXamlString) _
            Then flowReader2.Document = flowDoc2

        'Αντιγραφή του flowDoc1 στο flowDoc2.
    ElseIf btn Is btnCopyDocument1ToDocument2 Then _
        If FlowDocumentCopy(flowDoc1, flowDoc2) Then _
            flowReader2.Document = flowDoc2

        'Εμφάνισε μόνο το αμορφοποιητο κείμενο.
    ElseIf btn Is btnGetPlainTextOfDocument1 Then
        Dim plainText As String = FlowDocumentGetPlainText(flowDoc1)
        MessageBox.Show(plainText, "Σωκράτης", _
            MessageBoxButton.OK, MessageBoxImage.Information)
    End If

End If

End Sub

'Δημιουργεί το xaml string για το "δεν υπάρχει περιεχόμενο." στο FlowDocument2.
Private Function GetNoNoContentXamlString() As String


    Dim xamlString As String = "<FlowDocument Name=""flowDoc2""
xmlns=""http://schemas.microsoft.com/winfx/2006/xaml/presentation""><Paragraph
FontSize=""16"" Foreground=""Orange"">Δεν υπάρχει
περιεχόμενο</Paragraph></FlowDocument>"

    Return xamlString

End Function
```


- Αρχικά, τα 2 FlowDocuments εμφανίζονται ως εξής:

Socrates

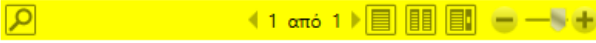


Socrates lived during the time of the transition from the height of the **Athenian hegemony** to its decline with the defeat by Sparta and its allies in the Peloponnesian War. At a time when Athens sought to stabilize and recover from its humiliating defeat, the Athenian public may have been entertaining doubts about democracy as an efficient form of government. Socrates appears to have been a critic of democracy, and some scholars[who?] interpret his trial as an expression of political infighting.

Xenophon and Plato agree that Socrates had an opportunity to escape, as his followers were able to bribe the prison guards. He chose to stay for several reasons.



Δεν υπάρχει περιεχόμενο.




Καθάρισμα του 2ου document

Αντιγραφή στο 2ο document

Πάρε το plain text από το 1ο document


- Το **καθάρισμα** του 2ου FlowDocument θα εμφανίσει την ίδια εικόνα. Αν πατήσουμε το button για την **αντιγραφή** το αποτέλεσμα είναι το εξής:

Socrates



Socrates lived during the time of the transition from the height of the **Athenian hegemony** to its decline with the defeat by Sparta and its allies in the Peloponnesian War. At a time when Athens sought to stabilize and recover from its humiliating defeat, the Athenian public may have been entertaining doubts about democracy as an efficient form of government. Socrates appears to have been a critic of democracy, and some scholars[who?] interpret his trial as an expression of political infighting.

Xenophon and Plato agree that Socrates had an opportunity to escape, as his followers were able to bribe the prison guards. He chose to stay for several reasons.



Καθάρισμα του 2ου document

Αντιγραφή στο 2ο document

Πάρε το plain text από το 1ο document

Socrates




Socrates lived during the time of the transition from the height of the **Athenian hegemony** to its decline with the defeat by Sparta and its allies in the Peloponnesian War. At a time when Athens sought to stabilize and recover from its humiliating defeat, the Athenian public may have been entertaining doubts about democracy as an efficient form



- Αν πατήσουμε το button για το **plain text** του **πρώτου FlowDocument** θα εμφανιστεί το αμορφοποίητο κείμενο σε ένα MessageBox:

Σωκράτης



Socrates

Socrates lived during the time of the transition from the height of the Athenian hegemony to its decline with the defeat by Sparta and its allies in the Peloponnesian War. At a time when Athens sought to stabilize and recover from its humiliating defeat, the Athenian public may have been entertaining doubts about democracy as an efficient form of government. Socrates appears to have been a critic of democracy, and some scholars[who?] interpret his trial as an expression of political infighting.

Xenophon and Plato agree that Socrates had an opportunity to escape, as his followers were able to bribe the prison guards. He chose to stay for several reasons.

OK

Σημείωση: Το απόσπασμα και η εικόνα προέρχονται από το αγγλικό λήμμα της Wikipedia.

FlowDocumentGetContent(flowDocument)

Επιστρέφει το μορφοποιημένο περιεχόμενο (xaml string με root element το <FlowDocument>) ενός FlowDocument.

Σύνταξη:

VB :

```
Public Shared Function FlowDocumentGetContent(  
    flowDocument As FlowDocument  
) As String
```

Παράμετροι:

flowDocument

Τύπος: System.Windows.Documents.FlowDocument

Το αντικείμενο FlowDocument, το περιεχόμενο του οποίου θα επιστραφεί ως xaml string.

Επιστρεφόμενη τιμή

Τύπος: System.String

Επιστρέφει το περιεχόμενο ως xaml string με root element το <FlowDocument>. Αν συμβεί σφάλμα θα επιστραφεί ένα κενό string (String.Empty).

Παρατηρήσεις:

Το περιεχόμενο ενός FlowDocument είναι ένα μορφοποιημένο κείμενο (μορφής xaml string με root element το <FlowDocument>).

Παράδειγμα:

Ένα FlowDocument, με όνομα flowDoc1, περιέχει πληροφορίες για τον Σωκράτη. Με τη μέθοδο **FlowDocumentGetContent** και παράμετρο το flowDoc1, παίρνουμε το περιεχόμενό του, δηλαδή το xaml string και το εμφανίζουμε σε ένα MessageBox:

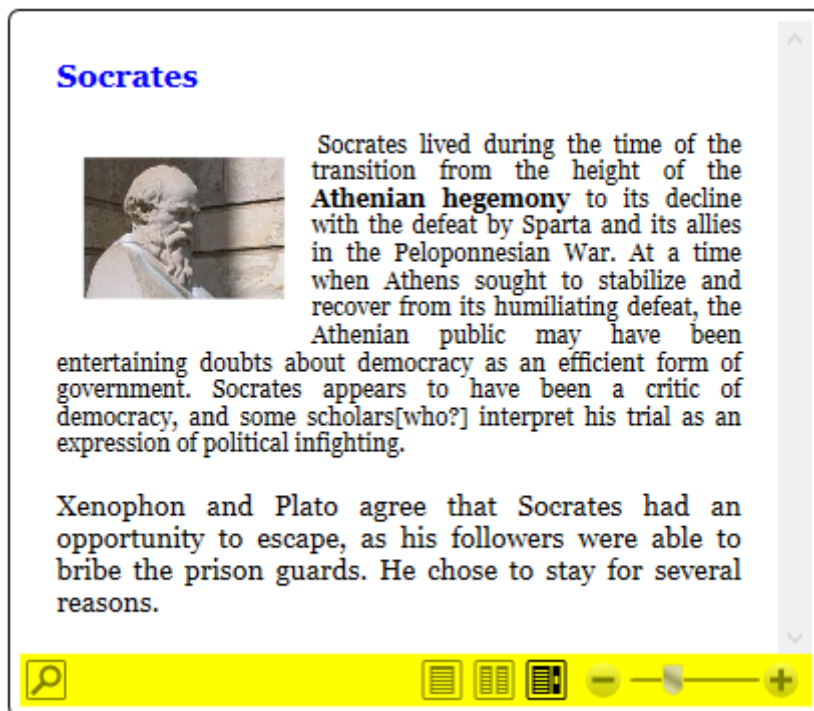
VB:

```
Imports Zeus.WPF.Classes.Utilities.FlowDocumentUtilities
```

```
...
```

```
Dim xamlString As String = FlowDocumentGetContent(flowDoc1)  
MessageBox.Show(xamlString)
```

Το περιεχόμενο του **flowDoc1** φαίνεται παρακάτω:



και το **xaml string** είναι το εξής:

```
xaml String
<FlowDocument Name="flowDoc1"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"><
Section><Paragraph FontWeight="ExtraBold" FontSize="16"
Foreground="#FF0000FF">Socrates</Paragraph><Paragraph
FontSize="12"><Figure HorizontalAnchor="PageLeft"
VerticalAnchor="ParagraphTop" HorizontalOffset="2"
VerticalOffset="0"><Paragraph><Image
Source="pack://application;,,,/Utilities Sample
Project;component/Images/Socrates.jpg" Width="100"
/></Paragraph></Figure> Socrates lived during the time of the
transition from the height of the <Bold>Athenian hegemony</Bold> to
its decline with the defeat by Sparta and its allies in the Peloponnesian
War. At a time when Athens sought to stabilize and recover from its
humiliating defeat, the Athenian public may have been entertaining
doubts about democracy as an efficient form of government. Socrates
appears to have been a critic of democracy, and some scholars[who?]
interpret his trial as an expression of political
infighting.</Paragraph></Section><Paragraph
FontSize="14">Xenophon and Plato agree that Socrates had an
opportunity to escape, as his followers were able to bribe the prison
guards. He chose to stay for several
reasons.</Paragraph></FlowDocument>
```

Αυτό το xaml string μπορεί να αποθηκευτεί σε ένα αρχείο στο δίσκο ή σε ένα database πεδίο τύπου **String/Text**. (π.χ. Σημειώσεις). Αργότερα, μπορεί να διαβαστεί και να τεθεί σε ένα FlowDocument μέσω της μεθόδου **FlowDocumentSetContent**. Με αυτόν τον

μηχανισμό, μπορούμε να παρέχουμε μορφοποιημένες σημειώσεις στους χρήστες των εφαρμογών μας, οι οποίες μπορούν να αποθηκευτούν μέσα στην Βάση Δεδομένων!

- Για ένα πλήρες παράδειγμα, μπορούμε να δούμε στο λήμμα [FlowDocumentCopy](#).

FlowDocumentGetPlainText(flowDocument)

Επιστρέφει το αμορφοποίητο κείμενο (plain text) ενός FlowDocument.

Σύνταξη:

VB :

```
Public Shared Function FlowDocumentGetPlainText(  
    flowDocument As FlowDocument  
)  
    As String
```

Παράμετροι:

flowDocument

Τύπος: System.Windows.Documents.FlowDocument

Το αντικείμενο FlowDocument, το αμορφοποίητο κείμενο του οποίου θα επιστραφεί ως string.

Επιστρεφόμενη τιμή

Τύπος: System.String

Επιστρέφει το αμορφοποίητο κείμενο (plain text). Αν συμβεί σφάλμα, θα επιστραφεί το string λάθους της shared ιδιότητας FlowDocumentGetPlainTextErrorString.

Παρατηρήσεις:

Το περιεχόμενο ενός FlowDocument είναι ένα μορφοποιημένο κείμενο (μορφής xaml string με root element to <FlowDocument>).

Παράδειγμα:

Ένα FlowDocument, με όνομα flowDoc1, περιέχει πληροφορίες για τον Σωκράτη. Με τη μέθοδο **FlowDocumentGetPlainText** και παράμετρο το flowDoc1, παίρνουμε το αμορφοποίητο κείμενο, δηλαδή το plain text και το εμφανίζουμε σε ένα MessageBox:

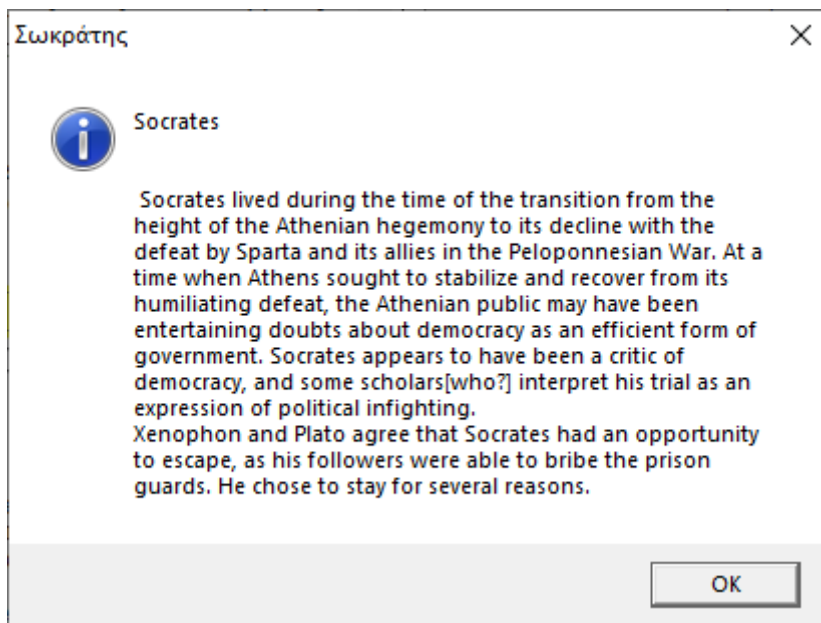
VB:

```
Imports Zeus.WPF.Classes.Utilities.FlowDocumentUtilities
```

...

```
Dim plainText As String = FlowDocumentGetPlainText(flowDoc1)  
MessageBox.Show(plainText, "Σωκράτης", MessageBoxButton.OK,  
MessageBoxImage.Information)
```

Το αποτέλεσμα φαίνεται παρακάτω:



- Για ένα πλήρες παράδειγμα, μπορούμε να δούμε στο λήμμα [FlowDocumentCopy](#).

FlowDocumentSetContent(flowDocument, xamlText)

Θέτει το μορφοποιημένο κείμενο (xaml string) ως περιεχόμενο του FlowDocument. Επιστρέφει True, αν η απόθεση ήταν επιτυχής.

Σύνταξη:

VB :

```
Public Shared Function FlowDocumentSetContent(  
    flowDocument As FlowDocument,  
    xamlText As String  
)  
    As String
```

Παράμετροι:

flowDocument

Τύπος: System.Windows.Documents.FlowDocument

Το αντικείμενο FlowDocument, το περιεχόμενο του οποίου θα αντικατασταθεί από την xamlText.

xamlText

Τύπος: System.String

Το xaml string, με root element το <FlowDocument>.

Επιστρεφόμενη τιμή

Τύπος: System.Boolean

Επιστρέφει True, αν η απόθεση ήταν επιτυχής.

Παρατηρήσεις:

Το περιεχόμενο ενός FlowDocument είναι ένα μορφοποιημένο κείμενο (μορφής xaml string με root element το <FlowDocument>). Συνεπώς η παράμετρος xamlText πρέπει να είναι ένα xaml string με root element το <FlowDocument>.

Παράδειγμα:

Στο παρακάτω παράδειγμα, ορίζουμε μία μεταβλητή, με όνομα xamlString, και θέτουμε το xaml περιεχόμενο με root element το <FlowDocument>. Το document περιέχει μόνο μία παράγραφο με κείμενο "Δεν υπάρχει περιεχόμενο", μεγέθους 16 και χρώματος πορτοκαλί. Με τη μέθοδο **FlowDocumentSetContent** και παράμετρο την flowDoc2 και την xamlString, θέτουμε το περιεχόμενο του flowDoc2 σε αυτό που προσδιορίζει η xamlString. Επιπλέον, για να εμφανιστεί στο FlowDocumentReader, με όνομα flowReader2, θέτουμε την αντίστοιχη ιδιότητα Document.

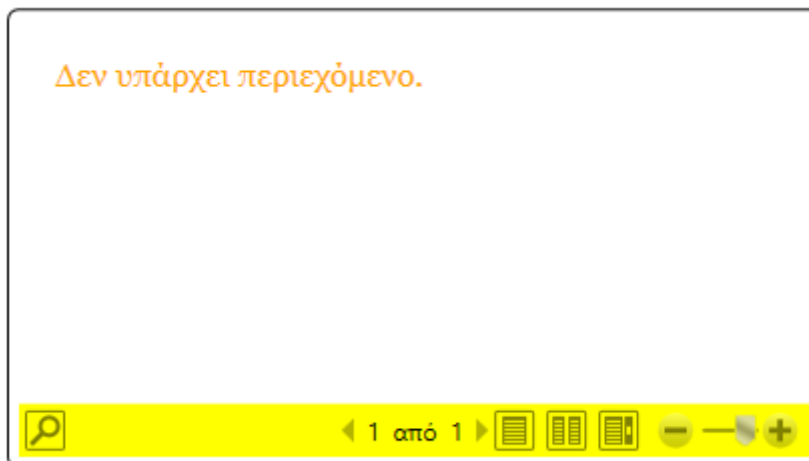
VB:

```
Imports Zeus.WPF.Classes.Utilities.FlowDocumentUtilities
```

```
Dim xamlString As String = "<FlowDocument Name=""flowDoc2""  
xmlns=""http://schemas.microsoft.com/winfx/2006/xaml/presentation""><Paragraph  
FontSize=""16"" Foreground=""Orange"">Δεν υπάρχει  
περιεχόμενο.</Paragraph></FlowDocument>"
```

```
If FlowDocumentSetContent(flowDoc2, xamlString) Then flowReader2.Document = flowDoc2
```

Το αποτέλεσμα φαίνεται παρακάτω:



- Φυσικά, το xaml string θα μπορούσε να περιέχει επιπλέον παραγράφους, εικόνες, λίστες με κουκκίδες, πίνακες κλπ. Αν επιθυμούμε ένα σύντομο **help μήνυμα ή οδηγίες** να εμφανίζονται κάπου, τότε το FlowDocument σε συνδυασμό με το container FlowDocumentReader είναι μία καλή λύση. Επίσης, θα μπορούσε το xaml string να είναι αποθηκευμένο σε κάποιο αρχείο, να διαβάζεται με ένα αντικείμενο **XamlReader** και να φορτώνεται/αποτίθεται με τη μέθοδο **FlowDocumentSetContent** στο αντίστοιχο FlowDocument.
- Για ένα πλήρες παράδειγμα, μπορούμε να δούμε στο λήμμα [FlowDocumentCopy](#).

RichTextBox - Μέθοδοι

Όνομα	Περιγραφή
RichTextBoxClearDocument (ByRef richTextBox As RichTextBox) As Boolean	Καθαρίζει το περιεχόμενο περιεχόμενο ενός RichTextBox. Επιστρέφει True, αν το καθάρισμα ήταν επιτυχές. Σημείωση: Το περιεχόμενο ενός RichTextBox είναι ένα FlowDocument. Έτσι, το καθάρισμα σημαίνει ότι τίθεται ένα κενό FlowDocument.
RichTextBoxCopy (ByRef fromRichTextBox As RichTextBox, ByRef toRichTextBox As RichTextBox) As Boolean	Αντιγράφει το περιεχόμενο ενός RichTextBox σε άλλο. Επιστρέφει True, αν η αντιγραφή ήταν επιτυχής.
RichTextBoxIsEmpty (ByRef richTextBox As RichTextBox) As Boolean	Επιστρέφει True αν το RichTextBox δεν έχει περιεχόμενο.
RichTextBoxSetDocument (ByRef richTextBox As RichTextBox, newDocument As FlowDocument) As Boolean	Θέτει ένα FlowDocument ως περιεχόμενο του RichTextBox. Επιστρέφει True, αν η απόθεση ήταν επιτυχής.
RichTextBoxSetDocument (ByRef richTextBox As RichTextBox, flowDocumentXamlText As String) As Boolean	Θέτει ένα xaml string με root element το <FlowDocument> ως περιεχόμενο του RichTextBox. Επιστρέφει True, αν η απόθεση ήταν επιτυχής.

RichTextBoxClearDocument(richTextBox)

Καθαρίζει το περιεχόμενο περιεχόμενο ενός RichTextBox. Επιστρέφει True, αν το καθάρισμα ήταν επιτυχές.

Σύνταξη:

VB :

```
Public Shared Function RichTextBoxClearDocument(  
    richTextBox As RichTextBox  
)  
    As Boolean
```

Παράμετροι:

richTextBox

Τύπος: System.Windows.Controls.RichTextBox

Το αντικείμενο RichTextBox, το περιεχόμενο του οποίου θα καθαριστεί.

Επιστρεφόμενη τιμή

Τύπος: System.Boolean

Επιστρέφει True, αν το καθάρισμα ήταν επιτυχές. Τεχνικά μιλώντας, τίθεται ένα κενό FlowDocument με βάση την shared ιδιότητα NewEmptyFlowDocumentString.

Παρατηρήσεις:

Το περιεχόμενο ενός RichTextBox είναι ένα FlowDocument.

Παράδειγμα:

Στο παρακάτω παράδειγμα, έχουμε τοποθετήσει σε ένα Grid δύο RichTextBoxes. Το πρώτο RichTextBox, με όνομα richTextBox1, περιέχει πληροφορίες για τον Σωκράτη. Το δεύτερο RichTextBox, με όνομα richTextBox2, είναι κενό (απλά περιέχει ένα μήνυμα ότι δεν έχει υπάρχει περιεχόμενο). Σε ένα StackPanel, δεξιά, έχουμε τοποθετήσει μία στοιβα από buttons. Το πρώτο κάνει καθάρισμα του richTextBox2, το δεύτερο κάνει αντιγραφή του περιεχομένου του richTextBox1 στο richTextBox2 και το τρίτο θέτει ένα xaml string με root element το <FlowDocument> ως περιεχόμενο στο richTextBox2.

Επιπλέον, έχουμε τοποθετήσει δύο επιπλέον buttons για αποθήκευση και διάβασμα του περιεχομένου του richTextBox2 σε/από .rtf αρχείο.

XAML:

```

<Grid>

  <Grid.RowDefinitions >
    <RowDefinition Height="Auto"/>
    <RowDefinition />
  </Grid.RowDefinitions>

  <Grid.ColumnDefinitions >
    <ColumnDefinition Width="*"/>
    <ColumnDefinition />
  </Grid.ColumnDefinitions>

  <Border BorderBrush="Black" BorderThickness="1" CornerRadius="5" Margin="5">

    <RichTextBox Name="richTextBox1" Margin="5" Background="Yellow">

      <FlowDocument Name="richFlowDoc1">

        <Section>

          <Paragraph FontSize="16" FontWeight="ExtraBold"
            Foreground="Blue" >
            Socrates
          </Paragraph>

          <Paragraph FontSize="12" >

            <Figure VerticalAnchor="ParagraphTop"
              HorizontalAnchor="PageLeft"
              VerticalOffset="0"
              HorizontalOffset="2" >

              <Paragraph >
                <InlineUIContainer >
                  <Image Source="Images/Socrates.jpg"
                    Width="100" />
                </InlineUIContainer >
              </Paragraph>

            </Figure>

            Socrates lived during the time of the transition from the
            height of the <Bold>Athenian hegemony</Bold> to its
            decline with the defeat by Sparta and its allies in the
            Peloponnesian War. At a time when Athens sought to
            stabilize and recover from its humiliating defeat, the
            Athenian public may have been entertaining doubts about
            democracy as an efficient form of government.
            Socrates appears to have been a critic of democracy, and
            some scholars[who?] interpret his trial as an expression
            of political infighting.

          </Paragraph>

        </Section>

        <Paragraph FontSize="14" >

```

Xenophon and Plato agree that Socrates had an opportunity to escape, as his followers were able to bribe the prison guards. He chose to stay for several reasons.

```

</Paragraph>

</FlowDocument>

</RichTextBox>

</Border>

<Border Grid.Row="1" BorderBrush="Black" BorderThickness="1" CornerRadius="5"
        Margin="5">

    <RichTextBox Name="richTextBox2" Margin="5" Background="Yellow">

        <FlowDocument Name="richflowDoc2">
            <Paragraph FontSize="16" Foreground="Orange" >
                Δεν υπάρχει περιεχόμενο.
            </Paragraph>
        </FlowDocument>

    </RichTextBox>

</Border>

<StackPanel Grid.Column="1" VerticalAlignment="Center" Margin="5"
            ButtonBase.Click="btnRichTextBoxes_Click" >

    <Button Name="btnClearDocument2"
            Content="Καθάρισμα του 2ου document" Padding="5" />

    <Button Name="btnCopyDocument1ToDocument2"
            Content="Αντιγραφή στο 2ο document"
            Margin="0,10,0,0" Padding="5" />

    <Button Name="btnGetPlainTextOfDocument1"
            Content="Πάρε το plain text από το 1ο document"
            Margin="0,10,0,0" Padding="5" />

    <Button Name="btnSaveToFileRichTextBox2"
            Content="Αποθήκευση του RichTextBox2 σε αρχείο .rtf"
            Margin="0,20,0,0" Padding="5"/>

    <Button Name=" btnLoadFromFileRichTextBox2"
            Content="Διάβασμα απο αρχείο .rtf στο RichTextBo2"
            Margin="0,10,0,0" Padding="5" />

</StackPanel>

</Grid>

VB:

Imports Zeus.WPF.Classes.Utilities.FlowDocumentUtilities
Imports System.IO

```

```
Private Sub btnRichTextBoxes_Click(sender As Object, e As RoutedEventArgs)

    Dim btn As Button = TryCast(e.OriginalSource, Button)

    If btn IsNot Nothing Then

        'Καθάρισμα του 2ου RichTextBox.
        If btn Is btnClearRichTextBox2 Then
            If RichTextBoxClearDocument(richTextBox2) Then _
                FadeIn(richTextBox2, AnimationSpeedEnum.Fast)

        'Αντιγραφή του richTextBox1 στο richTextBox2.
        ElseIf btn Is btnCopyRichTextBox1ToRichTextBox2 Then _
            If RichTextBoxCopy(richTextBox1, richTextBox2) Then _
                FadeIn(richTextBox2, AnimationSpeedEnum.Fast)

        'Βάλε ένα xaml string με root element το <FlowDocument> στο richTextBox2.
        ElseIf btn Is btnSetRichTextBoxContent Then
            Dim xamlString As String = "<FlowDocument Name=""richFlowDoc2""
            xmlns=""http://schemas.microsoft.com/winfx/2006/xaml/presentation"">
            <Paragraph FontSize=""16"" Foreground=""Orange"">Δεν υπάρχει
            περιεχόμενο.</Paragraph></FlowDocument>"

            RichTextBoxSetDocument(richTextBox2, xamlString)
            FadeIn(richTextBox2, AnimationSpeedEnum.Fast)

        'Αποθήκευση του richTextBox2 στο αρχείο Σημειώσεις.rtf στον τρέχον φάκελο
        'της εφαρμογής.
        ElseIf btn Is btnSaveToFileRichTextBox2 Then

            'Λαμβάνουμε την περιοχή που περικλείει όλο το FlowDocument του
            'richTextBox2.
            Dim docTextRange As _
                New TextRange(richTextBox2.Document.ContentStart, _
                    richTextBox2.Document.ContentEnd)

            'Αποθήκευση.
            Try
                Using fs As FileStream = File.Create("Σημειώσεις.rtf")

                    docTextRange.Save(fs, DataFormats.Rtf)
                    FadeIn(richTextBox2, AnimationSpeedEnum.Fast)

                End Using
                MessageBox.Show("Το αρχείο Σημειώσεις.rtf δημιουργήθηκε με
                    επιτυχία.", "Αποθήκευση")

            Catch ex As Exception
                MessageBox.Show("Πρόβλημα στην αποθήκευση" & vbCrLf & _
                    ex.Message, "Αποθήκευση")

            End Try

        'Φόρτωση από το αρχείο Σημειώσεις.rtf στο richTextBox2ς.
        ElseIf btn Is btnLoadFromFileRichTextBox2 Then

            'Λαμβάνουμε την περιοχή που περικλείει όλο το FlowDocument του
            richTextBox2.
            Dim docTextRange As New _
                TextRange(richTextBox2.Document.ContentStart, _
                    richTextBox2.Document.ContentEnd)
```

```

'Φόρτωση.
Try
    Using fs As FileStream = File.Open("Σημειώσεις.rtf",
        FileMode.Open)

        docTextRange.Load(fs, DataFormats.Rtf)
        FadeIn(richTextBox2, AnimationSpeedEnum.Fast)

    End Using

Catch ex As Exception
    MessageBox.Show("Πρόβλημα στην φόρτωση" & vbCrLf _
        & ex.Message, "Φόρτωση")

End Try

End If


End If

End Sub

```

- Αρχικά, τα 2 RichTextBoxes εμφανίζονται ως εξής:

Socrates



Socrates lived during the time of the transition from the height of the **Athenian hegemony** to its decline with the defeat by Sparta and its allies in the Peloponnesian War. At a time when Athens sought to stabilize and recover from its humiliating defeat, the Athenian public may have been entertaining doubts about democracy as an efficient form of government. Socrates appears to have been a critic of democracy, and some scholars[who?] interpret his trial as an expression of political infighting.


Xenophon and Plato agree that Socrates had an opportunity to escape, as his followers were able to bribe the prison guards. He chose to stay for several reasons.

- Καθάρισμα του 2ου RichTextBox
- Αντιγραφή στο 2ο RichTextBox
- Όρισε συγκεκριμένο περιεχόμενο στο RichTextBo2
- Αποθήκευση του RichTextBox2 σε αρχείο .rtf
- Διάβασμα απο αρχείο .rtf στο RichTextBo2

Δεν υπάρχει περιεχόμενο.

- Αν πατήσουμε το button για την **αντιγραφή** το αποτέλεσμα είναι το εξής:

Socrates




Socrates lived during the time of the transition from the height of the **Athenian hegemony** to its decline with the defeat by Sparta and its allies in the Peloponnesian War. At a time when Athens sought to stabilize and recover from its humiliating defeat, the Athenian public may have been entertaining doubts about democracy as an efficient form of government. Socrates appears to have been a critic of democracy, and some scholars[who?] interpret his trial as an expression of political infighting.

Xenophon and Plato agree that Socrates had an opportunity to escape, as his followers were able to bribe the prison guards. He chose to stay for several reasons.

- Καθάρισμα του 2ου RichTextBox
- Αντιγραφή στο 2ο RichTextBox
- Όρισε συγκεκριμένο περιεχόμενο στο RichTextBo2
- Αποθήκευση του RichTextBox2 σε αρχείο .rtf
- Διάβασμα απο αρχείο .rtf στο RichTextBo2

Socrates



Socrates lived during the time of the transition from the height of the **Athenian hegemony** to its decline with the defeat by Sparta and its allies in the Peloponnesian War. At a time when Athens sought to stabilize and recover from its humiliating defeat, the Athenian public may have been entertaining doubts about democracy as an efficient form of government. Socrates appears to have been a critic of democracy, and some scholars[who?] interpret his trial as an expression of political infighting.

Xenophon and Plato agree that Socrates had an opportunity to escape, as his followers were able to bribe the prison guards. He chose to stay for several reasons.

- Αν πατήσουμε το button για την **Αποθήκευση** του **richTextBox2** σε **.rtf** αρχείο, θα δημιουργηθεί στον δίσκο το αρχείο "**Σημειώσεις.rtf**". Αν το ανοίξουμε στο Word θα δούμε το rtf περιεχόμενο:

Socrates




Socrates lived during the time of the transition from the height of the **Athenian hegemony** to its decline with the defeat by Sparta and its allies in the Peloponnesian War. At a time when Athens sought to stabilize and recover from its humiliating defeat, the Athenian public may have been entertaining doubts about democracy as an efficient form of government. Socrates appears to have been a critic of democracy, and some scholars[who?] interpret his trial as an expression of political infighting. Xenophon and Plato agree that Socrates had an opportunity to escape, as his followers were able to bribe the prison guards. He chose to stay for several reasons.

(Η μορφοποίηση ExtraBold του τίτλου δεν "πέρασε" στο rtf. Ίσως, πρέπει να προτιμήσουμε το απλό Bold).

- Αν πατήσουμε το button για το **Διάβασμα** του αρχείου "**Σημειώσεις.rtf**" και **φόρτωσή** του στο **richTextBox2**:

Socrates



Socrates lived during the time of the transition from the height of the **Athenian hegemony** to its decline with the defeat by Sparta and its allies in the Peloponnesian War. At a time when Athens sought to stabilize and recover from its humiliating defeat, the Athenian public may have been entertaining doubts about democracy as an efficient form of government. Socrates appears to have been a critic of democracy, and some scholars[who?] interpret his trial as an expression of political infighting.

Xenophon and Plato agree that Socrates had an opportunity to escape, as his followers were able to bribe the prison guards. He chose to stay for several reasons.

Socrates



Socrates lived during the time of the transition from the height of the **Athenian hegemony** to its decline with the defeat by Sparta and its allies in the Peloponnesian War. At a time when Athens sought to stabilize and recover from its humiliating defeat, the Athenian public may have been entertaining doubts about democracy as an efficient form of government. Socrates appears to have been a critic of democracy, and some scholars[who?] interpret his trial as an expression of political infighting. Xenophon and Plato agree that Socrates had an opportunity to escape, as his followers were able to bribe the prison guards. He chose to stay for several reasons.

Καθάρισμα του 2ου RichTextBox

Αντιγραφή στο 2ο RichTextBox

Ορίσε συγκεκριμένο περιεχόμενο στο RichTextBo2

Αποθήκευση του RichTextBox2 σε αρχείο .rtf

Διάβασμα απο αρχείο .rtf στο RichTextBo2

Βλέπουμε, βέβαια, ότι υπάρχουν μικρές διαφορές, καθότι το πρότυπο .rtf έχει συγκεκριμένες προδιαγραφές. Σε κάθε περίπτωση, όμως, αποτελεί χρήσιμη δυνατότητα η αποθήκευση/ανάκτηση σε/από .rtf αρχείο αφού επιτρέπει την αναγνωσιμότητα του αρχείου από άλλα προγράμματα. Γενικά, γύρω από το RichTextBox μπορούμε να "χτίσουμε" ένα TextEditor control.

Αν επιθυμούμε να αποθηκεύσουμε ακριβώς στη μορφή που βλέπουμε τότε αντί του format .rtf θα προτιμήσουμε το format XAML Package:

```
docTextRange.Save(fs, DataFormats.XamlPackage)
```

Η χρήση του, όμως, θα περιορίζεται εντός της εφαρμογής μας.

Σημείωση: Το απόσπασμα και η εικόνα προέρχονται από το αγγλικό λήμμα της Wikipedia.

RichTextBoxCopy(fromRichTextBox, toRichTextBox)

Αντιγράφει το περιεχόμενο ενός RichTextBox σε άλλο. Επιστρέφει True, αν η αντιγραφή ήταν επιτυχής.

Σύνταξη:

VB :

```
Public Shared Function RichTextBoxCopy(  
    fromRichTextBox As RichTextBox,  
    toRichTextBox As RichTextBox  
)  
    As Boolean
```

Παράμετροι:

fromRichTextBox

Τύπος: System.Windows.Controls.RichTextBox

Το αντικείμενο RichTextBox, το περιεχόμενο του οποίου θα αντιγραφεί (source).

toRichTextBox

Τύπος: System.Windows.Controls.RichTextBox

Το αντικείμενο RichTextBox, στο οποίο θα γίνει η επικόλληση (target).

Επιστρεφόμενη τιμή

Τύπος: System.Boolean

Επιστρέφει True, αν η αντιγραφή ήταν επιτυχής.

Παρατηρήσεις:

Το περιεχόμενο ενός RichTextBox είναι ένα FlowDocument. Δεν μπορούμε να θέσουμε άμεσα την ιδιότητα Document ενός RichTextBox σε ένα Document που ανήκει σε άλλο RichTextBox. Με άλλα λόγια, δεν μπορούν δύο RichTextBoxes να αναφέρονται (point) στο ίδιο αντικείμενο FlowDocument. Έτσι, ο παρακάτω VB κώδικας:

```
richTextBox2.Document = richTextBox1.Document
```

θα προκαλέσει run-time error. Η χρήση της μεθόδου RichTextBoxCopy κάνει εύκολη την αντιγραφή καθώς, τεχνικά μιλώντας, κάνει xaml parse το αντικείμενο FlowDocument του source RichTextBox, το επαναδημιουργεί σε ένα νέο αντικείμενο FlowDocument και το αποθέτει στο target RichTextBox.

Παράδειγμα:

Στο παρακάτω παράδειγμα, αντιγράφουμε το περιεχόμενο του richTextBox1 στο richTextBox2 και αν αυτή ήταν επιτυχής τότε εφαρμόζουμε και ένα fade-in εφέ στο δεύτερο.

VB:

```
Imports Zeus.WPF.Classes.Utilities.FlowDocumentUtilities
```

```
Imports Zeus.WPF.Classes.Utilities.AnimationUtilities
```

```
...
```

```
If RichTextBoxCopy(richTextBox1, richTextBox2) Then _  
    FadeIn(richTextBox2, AnimationSpeedEnum.Fast)
```

- Για ένα πλήρες παράδειγμα, μπορούμε να δούμε στο λήμμα [RichTextBoxClearDocument.](#)

RichTextBoxIsEmpty(richTextBox)

Επιστρέφει True αν το RichTextBox δεν έχει περιεχόμενο.

Σύνταξη:

VB :

```
Public Shared Function RichTextBoxIsEmpty(  
    richTextBox As RichTextBox  
) As Boolean
```

Παράμετροι:

richTextBox

Τύπος: System.Windows.Controls.RichTextBox

Το αντικείμενο RichTextBox, το περιεχόμενο του οποίου θα ελεγχθεί.

Επιστρεφόμενη τιμή

Τύπος: System.Boolean

Επιστρέφει True, αν δεν έχει περιεχόμενο (κενό FlowDocument).

Παρατηρήσεις:

Το περιεχόμενο ενός RichTextBox είναι ένα FlowDocument.

Παράδειγμα:

Στο παρακάτω παράδειγμα, αν το richTextBox2 είναι κενό τότε εφαρμόζεται ένα move-in εφέ.

VB:

```
Imports Zeus.WPF.Classes.Utilities.FlowDocumentUtilities
```

```
Imports Zeus.WPF.Classes.Utilities.AnimationUtilities
```

```
...
```

```
If RichTextBoxIsEmpty(richTextBox2) Then _  
    MoveIn(richTextBox2, -100, 0, AnimationSpeedEnum.Fast)
```

- Για ένα πλήρες παράδειγμα, μπορούμε να δούμε στο λήμμα [RichTextBoxClearDocument.](#)

RichTextBoxSetDocument(richTextBox, newDocument)

Θέτει ένα **FlowDocument** ως περιεχόμενο του **RichTextBox**. Επιστρέφει **True**, αν η απόθεση ήταν επιτυχής.

Σύνταξη:

VB :

```
Public Shared Function RichTextBoxSetDocument(  
    richTextBox As RichTextBox,  
    newDocument As FlowDocument  
)  
    As Boolean
```

Παράμετροι:

richTextBox

Τύπος: System.Windows.Controls.RichTextBox

Το αντικείμενο RichTextBox, στο οποίο θα τεθεί το νέο FlowDocument.

toRichTextBox

Τύπος: System.Windows.Documents.FlowDocument

Το αντικείμενο FlowDocument, το οποίο θα τεθεί ως περιεχόμενο.

Επιστρεφόμενη τιμή

Τύπος: System.Boolean

Επιστρέφει True, αν η απόθεση ήταν επιτυχής.

Παρατηρήσεις:

Το περιεχόμενο ενός RichTextBox είναι ένα FlowDocument.

Παράδειγμα:

Στο παρακάτω παράδειγμα, θέτουμε ως περιεχόμενο στο richTextBox2 ένα νέο FlowDocument. Το τελευταίο, μπορούμε να το "χτίσουμε" στον VB κώδικα εκτός από το να θέσουμε άμεσα ένα xaml string:

VB:

```
Imports Zeus.WPF.Classes.Utilities.FlowDocumentUtilities
```

```
Imports Zeus.WPF.Classes.Utilities.AnimationUtilities
```

```
...
```

```
Dim flowDoc As New FlowDocument() With {.Foreground = Brushes.Orange, _  
    .FontFamily = New FontFamily("Courier New"), _  
    .FontSize = 22}
```

With flowDoc

```
.Blocks.Add(New Paragraph(New Run( "Δεν υπάρχει περιεχόμενο.")))
```

End With

```
richTextBox2.Padding = New Thickness(10)
```

```
If RichTextBoxSetDocument(richTextBox2, flowDoc) Then _  
    FadeIn(richTextBox2, AnimationSpeedEnum.Fast)
```

Το αποτέλεσμα φαίνεται στην παρακάτω εικόνα:



- Για ένα πλήρες παράδειγμα, μπορούμε να δούμε στο λήμμα [RichTextBoxClearDocument.](#)

RichTextBoxSetDocument(richTextBox, flowDocumentXamlText)

Θέτει ένα xaml string με root element το <FlowDocument> ως περιεχόμενο του RichTextBox. Επιστρέφει True, αν η απόθεση ήταν επιτυχής.

Σύνταξη:

VB :

```
Public Shared Function RichTextBoxSetDocument(  
    richTextBox As RichTextBox,  
    flowDocumentXamlText As String  
)  
    As Boolean
```

Παράμετροι:

richTextBox

Τύπος: System.Windows.Controls.RichTextBox

Το αντικείμενο RichTextBox, στο οποίο θα τεθεί το νέο FlowDocument.

flowDocumentXamlText

Τύπος: System.String

Το xaml string με root element το <FlowDocument>.

Επιστρεφόμενη τιμή

Τύπος: System.Boolean

Επιστρέφει True, αν η απόθεση ήταν επιτυχής.

Παρατηρήσεις:

Το περιεχόμενο ενός RichTextBox είναι ένα FlowDocument. Εδώ, δημιουργείται με βάση το xaml string. Ένας άλλος τρόπος, είναι να δομήσουμε το FlowDocument στον VB κώδικα με τη χρήση Block (π.χ. Section, Paragraph) και Inline elements (π.χ. Run, Span, Bold, Italic, Underline, Hyperlink, LineBreak κ.α.).

Παράδειγμα:

Στο παρακάτω παράδειγμα, θέτουμε ως περιεχόμενο στο richTextBox2 ένα νέο FlowDocument με βάση ένα xaml string.

VB:

```
Imports Zeus.WPF.Classes.Utilities.FlowDocumentUtilities  
Imports Zeus.WPF.Classes.Utilities.AnimationUtilities
```

...


```
Dim xamlString As String = "<FlowDocument Name=""richlowDoc2""  
xmlns=""http://schemas.microsoft.com/winfx/2006/xaml/presentation""><Paragraph  
FontSize=""16"" Foreground=""Orange"">Δεν υπάρχει  
περιεχόμενο.</Paragraph></FlowDocument>"  
  
If RichTextBoxSetDocument(richTextBox2, xamlString) Then _  
    FadeIn(richTextBox2, AnimationSpeedEnum.Fast)
```

Το αποτέλεσμα φαίνεται στην παρακάτω εικόνα:



- Για ένα πλήρες παράδειγμα, μπορούμε να δούμε στο λήμμα [RichTextBoxClearDocument](#).



ImageUtilities

Περιλαμβάνει μία συλλογή από στατικές μεθόδους σχετικές με **διαχείριση εικόνων (images)**.

Σύνταξη:

VB:

```
Public Class ImageUtilities
```

Ο παρακάτω πίνακας περιέχει τις διαθέσιμες **μεθόδους**:

Όνομα	Περιγραφή
GetColor (colorHexString As String) As Nullable(Of Color)	Επιστρέφει ένα αντικείμενο Nullable(Of Color) με βάση την δεκαεξαδική τιμή χρώματος της colorHexString .
GetImageBytes (filepath As String) As Byte()	Επιστρέφει έναν πίνακα bytes με τα image data ενός αρχείου εικόνας .
GetImageBytes (bmSource As BitmapImage) As Byte()	Επιστρέφει έναν πίνακα bytes με τα image data ενός BitmapImage source .
GetImageSource (filePath As String) As ImageSource	Επιστρέφει ένα αντικείμενο ImageSource με βάση την filePath . Χρησιμοποιείται για την απόδοση εικόνας σε ένα Image element με βάση το όνομα ενός αρχείου.

GetImageSource (componentAssembly As Assembly, imagePathInComponent As String) As ImageSource	<p>Επιστρέφει ένα αντικείμενο ImageSource, που βρίσκεται στο componentAssembly στη θέση imagePathInComponent.</p> <p>Χρησιμοποιείται για την απόδοση εικόνας σε ένα Image element με βάση το απόλυτο Uri (σε ποιό assembly βρίσκεται και το path μέσα στο assembly).</p>
GetThumbnailImage (filePath As String, newHeight As Integer) As BitmapImage	<p>Επιστρέφει ένα BitmapImage source, με μικρότερη ανάλυση, από ένα αρχείο εικόνας. Προσδιορίζουμε το απαιτούμενο ύψος μέσω της παραμέτρου newHeight, ενώ το πλάτος καθορίζεται αυτόματα λόγω διατήρησης του aspect ratio.</p> <p>Η μέθοδος κάνει resizing μέσω pixel decoding, κάτι που εξοικονομεί μνήμη όταν η ανάλυση είναι πολύ υψηλή.</p>
GetThumbnailImage (imgBytes As Byte(), newHeight As Integer) As BitmapImage	<p>Επιστρέφει ένα BitmapImage source, με μικρότερη ανάλυση, από ένα image data bytes array. Προσδιορίζουμε το απαιτούμενο ύψος μέσω της παραμέτρου newHeight, ενώ το πλάτος καθορίζεται αυτόματα λόγω διατήρησης του aspect ratio.</p> <p>Η μέθοδος κάνει resizing μέσω pixel decoding, κάτι που εξοικονομεί μνήμη όταν η ανάλυση είναι πολύ υψηλή.</p>

- **Σημείωση:** Η κατηγορία αυτή έχει χώρο για πολλές ακόμα μεθόδους. Ελπίζω, μελλοντικά να προσθέσω περισσότερες.

GetColor(colorHexString)

Επιστρέφει ένα **αντικείμενο Nullable(Of Color)** με βάση την δεκαεξαδική τιμή χρώματος της παραμέτρου **colorHexString**.

Σύνταξη:

VB :

```
Public Shared Function GetColor(  
    colorHexString As String  
)  
    As Nullable(Of Color)
```

Παράμετροι:

colorHexString

Τύπος: System.String

Η δεκαεξαδική τιμή χρώματος. Για παράδειγμα, "#D4E157".

Επιστρεφόμενη τιμή

Τύπος: System.Nullable(Of Color)

Επιστρέφει το αντίστοιχο χρώμα, ως αντικείμενο Nullable(Of Color). Αν συμβεί σφάλμα, επιστρέφει Nothing.

Παράδειγμα:

Παρακάτω, δίνουμε μία απόχρωση πράσινου στο χρώμα φόντου του button με όνομα btnAddStudent:

VB :

```
btnAddStudent.Background = New SolidColorBrush(GetColor("#D4E157"))
```

GetImageBytes(filepath)

Επιστρέφει έναν πίνακα bytes με τα image data ενός αρχείου εικόνας, το οποίο προσδιορίζεται στην παράμετρο filePath.

Σύνταξη:

VB :

```
Public Shared Function GetImageBytes(  
    filePath As String  
)  
    As Byte()
```

Παράμετροι:

filePath

Τύπος: System.String

Το path name του αρχείου εικόνας.

Επιστρεφόμενη τιμή

Τύπος: System.Byte()

Επιστρέφει μία δομή πίνακα από Byte. Αν συμβεί σφάλμα, επιστρέφει Nothing.

Παράδειγμα:

Παρακάτω, ζητάμε από τον χρήστη να επιλέξει ένα αρχείο εικόνας μέσω ενός παραθύρου διαλόγου, και εφόσον επιλέξει κάποιο, χρησιμοποιούμε τη μέθοδο **GetImageBytes** για να αποθηκεύσει τα **image data** σε έναν πίνακα bytes, με όνομα imgBytes,

VB :

```
Dim winTitle As String = "Επιλογή αρχείου εικόνας "  
  
Dim imgFile As New Microsoft.Win32.OpenFileDialog  
  
With imgFile  
    .Title = winTitle  
    .Filter = "Αρχεία εικόνων(*.JPG, *.JPEG, *.GIF, *.PNG)|*.JPG;*.JPEG;*.GIF;*.PNG"  
    .FilterIndex = 1  
    .CheckFileExists = True  
  
    If .ShowDialog() = True Then  
        'Image bytes array.  
        Dim imgBytes As Byte() = GetImageBytes(.Filename)  
  
    End If  
  
End With
```

GetImageBytes(bmSource)

Επιστρέφει έναν πίνακα bytes με τα image data ενός BitmapImage source, το οποίο προσδιορίζεται στην παράμετρο bmSource.

Σύνταξη:

VB :

```
Public Shared Function GetImageBytes(  
    bmSource As BitmapImage  
)  
    As Byte()
```

Παράμετροι:

bmSource

Τύπος: System.Windows.Media.Imaging

Το αντικείμενο BitmapImage, που περιέχει δεδομένα εικόνας.

Επιστρεφόμενη τιμή

Τύπος: System.Byte()

Επιστρέφει μία δομή πίνακα από Byte. Αν συμβεί σφάλμα, επιστρέφει Nothing.

Παρατηρήσεις:

Υποστηριζόμενα **formats** για το BitmapImage source είναι **JPEG/JPG, PNG, GIF, TIFF** και **BMP**.

Παράδειγμα:

Παρακάτω, ζητάμε από τον χρήστη να επιλέξει ένα αρχείο εικόνας μέσω ενός παραθύρου διαλόγου, και εφόσον επιλέξει κάποιο, χρησιμοποιούμε τη **μέθοδο GetImageBytes** για να αποθηκεύσει τα **image data** σε έναν πίνακα bytes, με όνομα imgBytes,

VB :

```
Dim winTitle As String = "Επιλογή αρχείου εικόνας "
```

```
Dim imgFile As New Microsoft.Win32.OpenFileDialog
```

```
With imgFile
```

```
    .Title = winTitle
```

```
    .Filter = "Αρχεία εικόνων (*.JPG, *.JPEG, *.GIF, *.PNG) | *.JPG;*.JPEG;*.GIF;*.PNG"
```

```
    .FilterIndex = 1
```

```
    .CheckFileExists = True
```

```
If .ShowDialog() = True Then
    'Image bytes array.
    Dim bmSource As New BitmapImage(New Uri(.FileName, _
                                           UriKind.RelativeOrAbsolute))
    Dim imgBytes As Byte() = GetImageBytes(bmSource)

End If

End With
```

GetImageSource(filePath)

Επιστρέφει ένα **αντικείμενο ImageSource** με βάση την παράμετρο **filePath**.

Σύνταξη:

VB :

```
Public Shared Function GetImageSource(  
    filePath As String  
)  
    As ImageSource
```

Παράμετροι:

filePath

Τύπος: System.String

Το path name του αρχείου εικόνας.

Επιστρεφόμενη τιμή

Τύπος: System.Windows.Media.ImageSource

Επιστρέφει το αντικείμενο ImageSource, το οποίο μπορεί να τεθεί ως περιεχόμενο στην ιδιότητα Source ενός Image element. Αν συμβεί σφάλμα, επιστρέφει Nothing.

Παρατηρήσεις:

- Το **αντικείμενο ImageSource** μπορεί να τεθεί ως περιεχόμενο στην **ιδιότητα Source** ενός **Image element**.

Παράδειγμα:

Παρακάτω, έχουμε ένα Image element με όνομα imgSunset και σε run-time mode θέτουμε το αρχείο "Sunset.png", που βρίσκεται στον φάκελο της εφαρμογής, στην ιδιότητα Source:

XAML :

```
<Image Name="imgSunset" Margin="10" Width="300" Height="200"  
    Stretch="Uniform" />
```

VB :

```
Dim imgSource As ImageSource = GetImageSource("Sunset.png")
```

```
If imgSource IsNot Nothing Then
```

```
    imgSunset.Source = imgSource
```


End If

GetImageSource(componentAssembly, imagePathInComponent)

Επιστρέφει ένα **αντικείμενο ImageSource**, που βρίσκεται στο **componentAssembly**, στη θέση **imagePathInComponent**.

Σύνταξη:

VB :

```
Public Shared Function GetImageSource(  
    componentAssembly As Assembly  
    imagePathInComponent As String  
)  
    As ImageSource
```

Παράμετροι:

componentAssembly

Τύπος: System.Reflection.Assembly

Το αντικείμενο Assembly που περιέχει το component, εντός του οποίου βρίσκεται η εικόνα.

imagePathInComponent

Τύπος: System.String

Το path της εικόνας μέσα στο component.

Επιστρεφόμενη τιμή

Τύπος: System.Windows.Media.ImageSource

Επιστρέφει το αντικείμενο ImageSource, το οποίο μπορεί να τεθεί ως περιεχόμενο στην ιδιότητα Source ενός Image element. Αν συμβεί σφάλμα, επιστρέφει Nothing.

Παρατηρήσεις:

Το **αντικείμενο ImageSource** μπορεί να τεθεί ως περιεχόμενο στην **ιδιότητα Source** ενός **Image element**.

Παράδειγμα:

Παρακάτω, έχουμε ένα Image element με όνομα imgSunset και σε run-time mode θέτουμε το αρχείο "Sunset.jpg", στην ιδιότητα Source. Το αρχείο αυτό το έχουμε εισάγει, ως Resource, στον Solution Explorer σε έναν υποφάκελο Images. Η μέθοδος, εσωτερικά, σχηματίζει ένα absolute Uri για να προσδιορίσει τη θέση του αρχείου εικόνας μέσα στο assembly της εφαρμογής.

XAML :

```
<Image Name="imgSunset" Margin="10" Width="300" Height="200"  
Stretch="Uniform" />
```

VB :

```
Dim imgSource As ImageSource = _  
    GetImageSource(System.Reflection.Assembly.GetExecutingAssembly, _  
                    "Images/Sunset.jpg")  
  
If imgSource IsNot Nothing Then  
    imgSunset.Source = imgSource  
  
End If
```

GetThumbnailImage(filePath, newHeight)

Επιστρέφει ένα **BitmapImage source**, με μικρότερη ανάλυση, από ένα αρχείο εικόνας. Προσδιορίζουμε το απαιτούμενο ύψος μέσω της παραμέτρου `newHeight`, ενώ το πλάτος καθορίζεται αυτόματα λόγω διατήρησης του `aspect ratio`.

Σύνταξη:

VB :

```
Public Shared Function GetThumbnailImage(  
    filePath As String,  
    newHeight As Integer  
)  
    As BitmapImage
```

Παράμετροι:

filePath

Τύπος: `System.String`
Το path name του αρχείου εικόνας.

newHeight

Τύπος: `System.Integer`
Το νέο ύψος.

Επιστρεφόμενη τιμή

Τύπος: `System.Windows.Media.Imaging.BitmapImage`
Επιστρέφει ένα αντικείμενο `BitmapImage`. Αν συμβεί σφάλμα, επιστρέφει `Nothing`.

Παρατηρήσεις:

- Ορίζουμε μόνο την παράμετρο του ύψους. **Το πλάτος θα προσδιοριστεί αυτόματα διότι διατηρεί το `aspect ratio`.**
- Η **μείωση του μεγέθους** γίνεται εσωτερικά στη μέθοδο μέσω **pixel decoding**, κάτι που εξοικονομεί μνήμη στη χρήση του επιστρεφόμενου `BitmapImage`.

Παράδειγμα:

Παρακάτω, ζητάμε από τον χρήστη να επιλέξει ένα αρχείο εικόνας μέσω ενός παραθύρου διαλόγου, και εφόσον επιλέξει κάποιον, χρησιμοποιούμε τη **μέθοδο `GetThumbnailImage`** για να **μειώσουμε το μέγεθος της εικόνας σε ύψος 512pixels**. Στη συνέχεια, θέτουμε το επιστρεφόμενο `BitmapImage source` στην ιδιότητα `Source` ενός υπάρχοντος `Image control` με όνομα `img`.

VB :

```
Dim winTitle As String = "Επιλογή αρχείου εικόνας "  
  
Dim imgFile As New Microsoft.Win32.OpenFileDialog  
  
With imgFile  
    .Title = winTitle  
    .Filter = "Αρχεία εικόνων(*.JPG, *.JPEG, *.GIF, *.PNG)|*.JPG;*.JPEG;*.GIF;*.PNG"  
    .FilterIndex = 1  
    .CheckFileExists = True  
  
    If .ShowDialog() = True Then  
        'Image thumbnail.  
        Dim imgSourceSmall As BitmapImage = GetThumbnailImage(.FileName, 512)  
        img.Source = imgSourceSmall  
    End If  
  
End With
```

Θα μπορούσαμε να γράψουμε επίσης:

```
Dim imgSourceSmall As ImageSource = GetThumbnailImage(imgBytes, 512)
```

διότι η κλάση BitmapImage είναι sub-class της ImageSource.

GetThumbnailImage(imgBytes, newHeight)

Επιστρέφει ένα **BitmapImage source**, με μικρότερη ανάλυση, από ένα **image data bytes array**. Προσδιορίζουμε το απαιτούμενο ύψος μέσω της παραμέτρου `newHeight`, ενώ το πλάτος καθορίζεται αυτόματα λόγω διατήρησης του `aspect ratio`.

Σύνταξη:

VB :

```
Public Shared Function GetThumbnailImage(  
    imgBytes As Byte(),  
    newHeight As Integer  
)  
    As BitmapImage
```

Παράμετροι:

imgBytes

Τύπος: `System.Byte()`
Το byte array που περιέχει image data .

newHeight

Τύπος: `System.Integer`
Το νέο ύψος .

Επιστρεφόμενη τιμή

Τύπος: `System.Windows.Media.Imaging.BitmapImage`
Επιστρέφει ένα αντικείμενο `BitmapImage`. Αν συμβεί σφάλμα, επιστρέφει `Nothing`.

Παρατηρήσεις:

- Ορίζουμε μόνο την παράμετρο του ύψους. Το πλάτος θα προσδιοριστεί αυτόματα διότι διατηρεί το **aspect ratio**.

Παράδειγμα:

Παρακάτω, χρησιμοποιούμε τη μέθοδο **GetThumbnailImage** για να μειώσουμε το μέγεθος της εικόνας σε ύψος **128pixels**. Η πηγή είναι ένα `image data array`. Στη συνέχεια, θέτουμε το επιστεφόμενο `BitmapImage source` στην ιδιότητα `Source` ενός υπάρχοντος `Image control` με όνομα `img`.

VB :

```
Dim imgSourceSmall As BitmapImage = GetThumbnailImage(imgBytes, 128)  
img.Source = imgSourceSmall
```

Θα μπορούσαμε να γράψουμε επίσης:

```
Dim imgSourceSmall As ImageSource = GetThumbnailImage(imgBytes, 128)
```

διότι η κλάση BitmapImage είναι sub-class της ImageSource.



MeasureUtilities

Περιλαμβάνει μία συλλογή από στατικές μεθόδους σχετικές με **διαχείριση και μετατροπή μετρήσεων (measures)**.

Σύνταξη:

VB:

```
Public Class MeasureUtilities
```

Ο παρακάτω πίνακας περιέχει τις διαθέσιμες **μεθόδους**:

Όνομα	Περιγραφή
ConvertCentimetersToInches (centimeters As Double) As Double	Μετατρέπει τα εκατοστά σε ίντσες .
ConvertCentimetersToPixels (centimeters As Double, dpiScaleFactor As Double) As Double	Μετατρέπει τα εκατοστά σε pixels , με βάση το dpi scale factor (<i>dpiX</i> ή <i>dpiY</i> , Τυπικά, 1.0 σημαίνει 96dpi).
ConvertInchesToCentimeters (inches As Double) As Double	Μετατρέπει τις ίντσες σε εκατοστά .
ConvertPixelsToCentimeters (pixels As Double, dpiScaleFactor As Double) As Double	Μετατρέπει τα pixels σε εκατοστά , με βάση το dpi scale factor (<i>dpiX</i> ή <i>dpiY</i> , Τυπικά, 1.0 σημαίνει 96dpi).
GetSystemDpiY () As Double	Επιστρέφει το dpiY του συστήματος .
GetSystemDpiX () As Double	Επιστρέφει το dpiX του συστήματος .

ConvertCentimetersToInches(centimeters)

Μετατρέπει τα εκατοστά σε ίντσες.

Σύνταξη:

VB :

```
Public Shared Function ConvertCentimetersToInches(  
    centimeters As Double  
) As Double
```

Παράμετροι:

centimeters

Τύπος: System.Double
Τα εκατοστά προς μετατροπή.

Επιστρεφόμενη τιμή

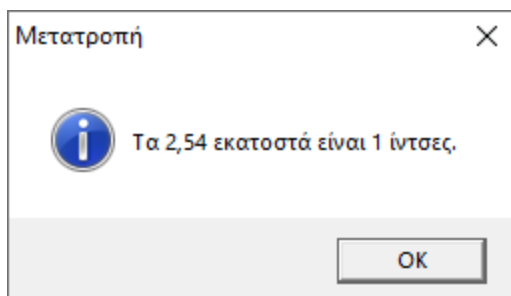
Τύπος: System.Double
Επιστρέφει τις αντίστοιχες ίντσες.

Παράδειγμα:

VB :

```
Dim centimeters As Double = 2.54  
MessageBox.Show(String.Format("Τα {0} εκατοστά είναι {1} ίντσες.", _  
    centimeters, ConvertCentimetersToInches(centimeters)), _  
    "Μετατροπή", MessageBoxButton.OK, _  
    MessageBoxImage.Information)
```

Το αποτέλεσμα φαίνεται παρακάτω:



ConvertCentimetersToPixels(centimeters, dpi)

Μετατρέπει τα **εκατοστά σε pixels**, με βάση το **dpi scale factor** (*dpiX ή dpiY, Τυπικά, 1.0 σημαίνει 96dpi*).

Σύνταξη:

VB :

```
Public Shared Function ConvertCentimetersToPixels(  
    centimeters As Double,  
    dpiScaleFactor As Double  
    )  
    As Double
```

Παράμετροι:

centimeters

Τύπος: System.Double
Τα εκατοστά προς μετατροπή.

dpiScaleFactor

Τύπος: System.Double
Το dpi scale factor (dpiX ή dpiY). Μία τυπική τιμή είναι 1, που σημαίνει 96dpi.

Επιστρεφόμενη τιμή

Τύπος: System.Double
Επιστρέφει τα αντίστοιχα pixels, με βάση το dpi scale factor.

Παρατηρήσεις:

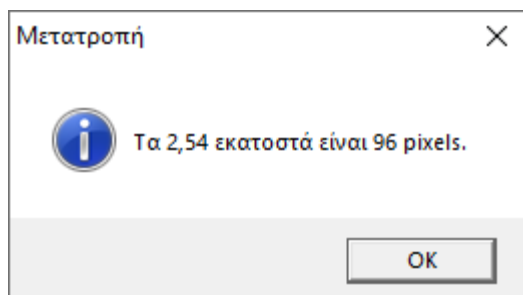
1 ίντσα = 2.54 εκατοστά.

Παράδειγμα 1:

VB :

```
Dim centimeters As Double = 2.54  
  
MessageBox.Show(String.Format("Τα {0} εκατοστά είναι {1} pixels.", _  
    centimeters, ConvertCentimetersToPixels(centimeters, 1)), _  
    "Μετατροπή", MessageBoxButton.OK, _  
    MessageBoxImage.Information)
```

Το αποτέλεσμα φαίνεται παρακάτω (*dpi scale factor=1 που αντιστοιχεί σε 96dpi*):



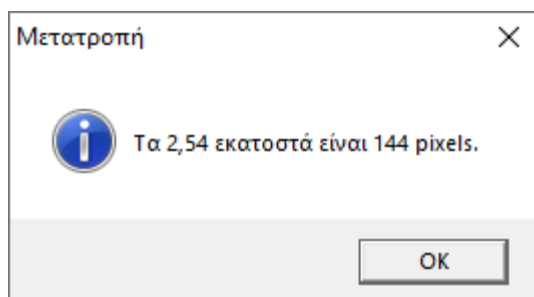
Παράδειγμα 2:

VB :

```
Dim centimeters As Double = 2.54
```

```
MessageBox.Show(String.Format("Τα {0} εκατοστά είναι {1} pixels.", _  
centimeters, ConvertCentimetersToPixels(centimeters, 1.5)), _  
"Μετατροπή", MessageBoxButtons.OK, _  
MessageBoxImage.Information)
```

Το αποτέλεσμα φαίνεται παρακάτω (*dpi scale factor=1.5 που αντιστοιχεί σε 144dpi*):



ConvertInchesToCentimeters(inches)

Μετατρέπει τις **ίντσες σε εκατοστά**.

Σύνταξη:

VB :

```
Public Shared Function ConvertInchesToCentimeters(  
    inches As Double  
) As Double
```

Παράμετροι:

inches

Τύπος: System.Double
Ο ίντσες προς μετατροπή.

Επιστρεφόμενη τιμή

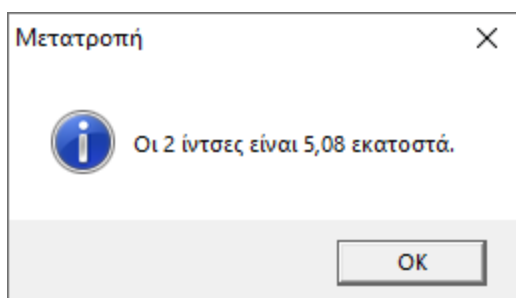
Τύπος: System.Double
Επιστρέφει τα αντίστοιχα εκατοστά.

Παράδειγμα:

VB :

```
Dim inches As Double = 2  
  
MessageBox.Show(String.Format("Οι {0} ίντσες είναι {1} εκατοστά.", _  
    inches, ConvertInchesToCentimeters(inches)), _  
    "Μετατροπή", MessageBoxButtons.OK, _  
    MessageBoxIcon.Information)
```

Το αποτέλεσμα φαίνεται παρακάτω:



ConvertPixelsToCentimeters(pixels, dpiScaleFactor)

Μετατρέπει τα **pixels** σε εκατοστά, με βάση το **dpi scale factor** (*dpiX* ή *dpiY*, Τυπικά, 1.0 σημαίνει 96dpi).

Σύνταξη:

VB :

```
Public Shared Function ConvertPixelsToCentimeters(  
    pixels As Double,  
    dpiScaleFactor As Double  
)  
    As Double
```

Παράμετροι:

pixels

Τύπος: System.Double

Τα pixels προς μετατροπή, λαμβανομένου υπόψη του dpi scale factor..

dpiScaleFactor

Τύπος: System.Double

Το dpi scale factor (*dpiX* ή *dpiY*). Μία τυπική τιμή είναι 1, που σημαίνει 96dpi.

Επιστρεφόμενη τιμή

Τύπος: System.Double

Επιστρέφει τα αντίστοιχα εκατοστά.

Παρατηρήσεις:

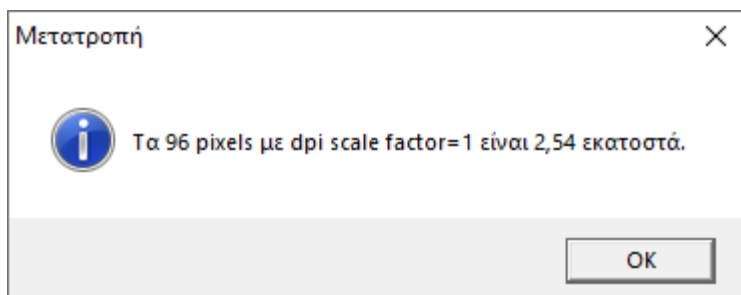
1 ίντσα = 2.54 εκατοστά.

Παράδειγμα 1:

VB :

```
Dim pixels As Double = 96  
  
MessageBox.Show(String.Format("Τα {0} pixels με dpi scale factor={1} είναι {2}  
εκατοστά.", pixels, 1.0, ConvertPixelsToCentimeters(pixels, 1.0)), _  
    "Μετατροπή", MessageBoxButton.OK, _  
    MessageBoxImage.Information)
```

Το αποτέλεσμα φαίνεται παρακάτω (*dpi scale factor=1 που αντιστοιχεί σε 96dpi*):



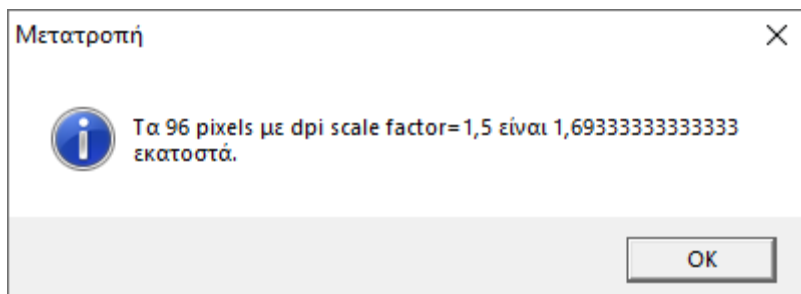
Παράδειγμα 2:

VB :

```
Dim pixels As Double = 96

MessageBox.Show(String.Format("Τα {0} pixels με dpi scale factor={1} είναι {2} εκατοστά.", pixels, 1.5, ConvertPixelsToCentimeters(pixels, 1.5)), _
    "Μετατροπή", MessageBoxButton.OK, _
    MessageBoxImage.Information)
```

Το αποτέλεσμα φαίνεται παρακάτω (*dpi scale factor=1.5 που αντιστοιχεί σε 144dpi*):



GetSystemDpiY()

Επιστρέφει το **dpiY** του συστήματος.

Σύνταξη:

VB :

```
Public Shared Function GetSystemDpiY() As Double
```

Επιστρεφόμενη τιμή

Τύπος: System.Double

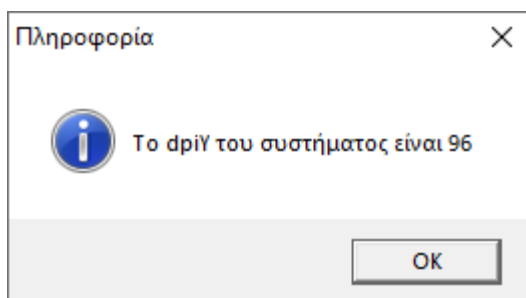
Επιστρέφει το dpiY του συστήματος. Μία τυπική τιμή είναι 96.

Παράδειγμα:

VB :

```
MessageBox.Show(String.Format("Το dpiY του συστήματος είναι {0}", _  
                             GetSystemDpiY), _  
                "Πληροφορία", MessageBoxButtons.OK, _  
                MessageBoxIcon.Information)
```

Το αποτέλεσμα φαίνεται παρακάτω:



GetSystemDpiX()

Επιστρέφει το **dpiX** του συστήματος.

Σύνταξη:

VB :

```
Public Shared Function GetSystemDpiX() As Double
```

Επιστρεφόμενη τιμή

Τύπος: System.Double

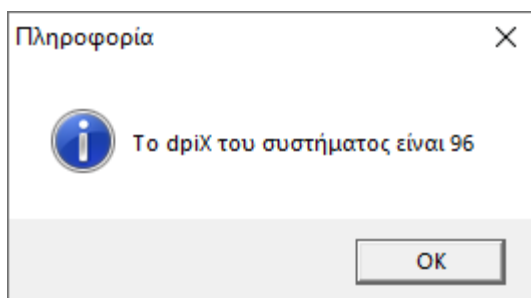
Επιστρέφει το dpiX του συστήματος. Μία τυπική τιμή είναι 96.

Παράδειγμα:

VB :

```
MessageBox.Show(String.Format("Το dpiX του συστήματος είναι {0}", _  
                             GetSystemDpiX), _  
                "Πληροφορία", MessageBoxButtons.OK, _  
                MessageBoxIcon.Information)
```

Το αποτέλεσμα φαίνεται παρακάτω:





SecurityUtilities

Περιλαμβάνει μία συλλογή από στατικές μεθόδους σχετικές με **ασφάλεια δεδομένων**.

Σύνταξη:

VB:

`Public Class SecurityUtilities`

Ο παρακάτω πίνακας περιέχει τις διαθέσιμες **κατηγορίες**:

Κατηγορία	Περιγραφή
Κρυπτογράφηση MD5	Περιλαμβάνει στατικές μεθόδους για κρυπτογράφηση κατά τον αλγόριθμο MD5 .
Κρυπτογράφηση SHA-1	Περιλαμβάνει στατικές μεθόδους για κρυπτογράφηση κατά τον αλγόριθμο SHA-1 .

Κρυπτογράφηση MD5

Όνομα	Περιγραφή
GetMD5Hash (inputString As String) As String	<p>Επιστρέφει ένα MD5 hash string, το οποίο δημιουργείται με βάση το input string.</p> <p>Χρησιμοποιεί εσωτερικά ένα MD5 αντικείμενο για την εκτέλεση του αντίστοιχου αλγόριθμου κρυπτογράφησης. Το επιστρεφόμενο MD5 hash string είναι 32 χαρακτήρων δεκαεξαδικής μορφής.</p>
GetMD5Hash (md5 As MD5, inputString As String) As String	<p>Επιστρέφει ένα MD5 hash string, το οποίο δημιουργείται με βάση το αντικείμενο MD5 και το input string.</p> <p>Το επιστρεφόμενο MD5 hash string είναι 32 χαρακτήρων δεκαεξαδικής μορφής.</p>
VerifyMD5Hash (inputString As String, hashString As String) As Boolean	<p>Επιστρέφει True, αν το MD5 hash του input string είναι το ίδιο με το hashString.</p> <p>Χρησιμοποιεί εσωτερικά ένα MD5 αντικείμενο για την εκτέλεση του αντίστοιχου αλγόριθμου κρυπτογράφησης του input string. Μόλις δημιουργηθεί το αντίστοιχο MD5 hash string, συγκρίνεται με αυτό της παραμέτρου hashString.</p>
VerifyMD5Hash (md5 As MD5, inputString As String, hashString As String) As Boolean	<p>Επιστρέφει True, αν το MD5 hash του input string είναι το ίδιο με το hashString.</p> <p>Χρησιμοποιεί το MD5 αντικείμενο της παραμέτρου για την εκτέλεση του αντίστοιχου αλγόριθμου κρυπτογράφησης του input string. Μόλις δημιουργηθεί το αντίστοιχο MD5 hash string, συγκρίνεται με αυτό της παραμέτρου hashString.</p>

GetMD5Hash(inputString)

Επιστρέφει ένα MD5 hash string, το οποίο δημιουργείται με βάση το **input string**. Χρησιμοποιεί εσωτερικά ένα MD5 αντικείμενο για την εκτέλεση του αντίστοιχου αλγόριθμου κρυπτογράφησης. Το επιστρεφόμενο MD5 hash string είναι 32 χαρακτήρων δεκαεξαδικής μορφής.

Σύνταξη:

VB :

```
Public Shared Function GetMD5Hash(  
    inputString As String  
) As String
```

Παράμετροι:

inputString

Τύπος: System.String

Το string εισόδου, βάσει του οποίου ο αλγόριθμος MD5 θα δημιουργήσει το hash string.

Επιστρεφόμενη τιμή

Τύπος: System.String

Επιστρέφει το MD5 hash string, 32 χαρακτήρων δεκαεξαδικής μορφής.

Παράδειγμα:

Παρακάτω, δίνουμε δημιουργούμε ένα MD5 hash string για το input string "mouratx".

VB :

```
Imports Zeus.WPF.Classes.Utilities.SecurityUtilities  
  
Dim md5Hash As String = GetMD5Hash("mouratx")
```

Το εξαγόμενο hash string είναι το εξής:

```
62c0ad3d8c1302c133f497574e394c5c
```

Το hash string μπορεί να αποθηκευτεί σε κάποιο αρχείο ή σε ένα database πεδίο, το οποίο να αντιστοιχεί σε κάποιο *password*. Όταν σε κατοπινή στιγμή ζητείται το password μπορούμε να πάρουμε το input string που θα δώσει ο χρήστης και να το συγκρίνουμε με αυτό το hash string, μέσω της μεθόδου VerifyMD5Hash.

GetMD5Hash(md5, inputString)

Επιστρέφει ένα **MD5 hash string**, το οποίο δημιουργείται με βάση το **αντικείμενο MD5** και το **input string**. Το επιστρεφόμενο MD5 hash string είναι 32 χαρακτήρων δεκαεξαδικής μορφής.

Σύνταξη:

VB :

```
Public Shared Function GetMD5Hash(  
    md5 As MD5,  
    inputString As String  
)  
    As String
```

Παράμετροι:

md5

Τύπος: System.Security.Cryptography.MD5
Το αντικείμενο MD5, για την εκτέλεση του αλγορίθμου MD5 επί του input string.

inputString

Τύπος: System.String
Το string εισόδου, βάσει του οποίου ο αλγόριθμος MD5 θα δημιουργήσει το hash string.

Επιστρεφόμενη τιμή

Τύπος: System.String
Επιστρέφει το MD5 hash string, 32 χαρακτήρων δεκαεξαδικής μορφής.

Παράδειγμα:

Παρακάτω, δίνουμε δημιουργούμε ένα MD5 hash string για το input string "mouratx".

VB :

```
Imports Zeus.WPF.Classes.Utilities.SecurityUtilities  
Imports System.Security.Cryptography  
  
Dim md5 As MD5 = MD5.Create  
Dim md5Hash As String = GetMD5Hash(md5, "mouratx")
```

Το εξαγόμενο hash string είναι το εξής:

62c0ad3d8c1302c133f497574e394c5c

Το hash string μπορεί να αποθηκευτεί σε κάποιο αρχείο ή σε ένα database πεδίο, το οποίο να αντιστοιχεί σε κάποιο *password*. Όταν σε κατοπινή στιγμή ζητείται το password μπορούμε να πάρουμε το input string που θα δώσει ο χρήστης, και να συγκρίνουμε το hash του με αυτό το hash string, μέσω της μεθόδου `VerifyMD5Hash`.

VerifyMD5Hash(inputString, hashString)

Επιστρέφει True, αν το MD5 hash του **input string** είναι το ίδιο με το **hashString**. Χρησιμοποιεί εσωτερικά ένα MD5 αντικείμενο για την εκτέλεση του αντίστοιχου αλγόριθμου κρυπτογράφησης του input string. Μόλις δημιουργηθεί το αντίστοιχο MD5 hash string, συγκρίνεται με αυτό της παραμέτρου hashString.

Σύνταξη:

VB :

```
Public Shared Function VerifyMD5Hash(  
    inputString As String,  
    hashString As String  
)  
    As Boolean
```

Παράμετροι:

inputString

Τύπος: System.String

Το string εισόδου, βάσει του οποίου ο αλγόριθμος MD5 εσωτερικά θα δημιουργήσει το hash string. Αυτό, θα συγκριθεί με εκείνο της παραμέτρου hashString.

hashString

Τύπος: System.String

Το MD5 hash string προς σύγκριση.

Επιστρεφόμενη τιμή

Τύπος: System.Boolean

Επιστρέφει True, αν η σύγκριση δώσει ισότητα.

Παράδειγμα:

Παρακάτω, στον XAML κώδικα, έχουμε ένα PasswordBox element σε μία Windows φόρμα, στο οποίο ο χρήστης πληκτρολογεί τον κωδικό του. Φυσικά, αυτός εμφανίζεται με κουκκίδες. Το περιεχόμενο του PasswordBox είναι το inputString που θέλουμε. Στον VB κώδικα, μέσω της μεθόδου VerifyMD5Hash, συγκρίνουμε το παραγόμενο MD5 hash key με αυτό που έχουμε αποθηκευμένο σε ένα αρχείο "Password.txt" κι εμφανίζουμε το αποτέλεσμα της σύγκρισης στην οθόνη.

XAML :

```
<PasswordBox Name="txtLoginCode" VerticalAlignment="Center" Width="100"/>
```

VB :

```
Imports Zeus.WPF.Classes.Utilities.SecurityUtilities
```

...

```
Dim hashKeyFromFile As String = String.Empty

'Πάρε το hash key από το αρχείο.
Using sReader As New IO.StreamReader("Password.txt")

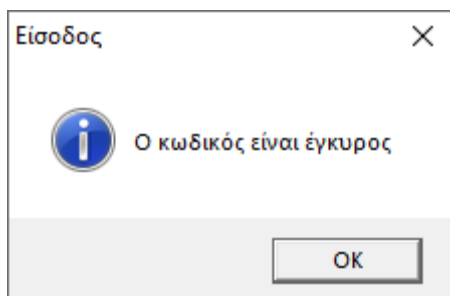
    hashKeyFromFile = sReader.ReadToEnd
    sReader.Close()

End Using

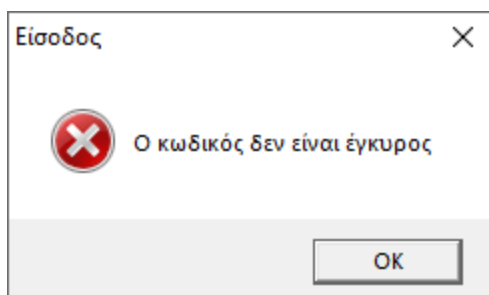
Dim isValid As Boolean = VerifyMD5Hash(inputString, hashKeyFromFile)

Dim strMessage As String = If(isValid, "Ο κωδικός είναι έγκυρος", _
    "Ο κωδικός δεν είναι έγκυρος")
Dim iconMessage As MessageBoxIcon = If(isValid, MessageBoxIcon.Information, _
    MessageBoxIcon.Error)
MessageBox.Show(strMessage, "Είσοδος", MessageBoxButtons.OK, iconMessage)
```

- Αν δοθεί σωστός κωδικός (το hash key του inputString συμφωνεί με αυτό του αρχείου):



- Αν δοθεί λάθος κωδικός (το hash key του inputString δεν συμφωνεί με αυτό του αρχείου):



VerifyMD5Hash(md5, inputString, hashString)

Επιστρέφει True, αν το MD5 hash του **input string** είναι το ίδιο με το **hashString**. Χρησιμοποιεί το MD5 αντικείμενο της παραμέτρου για την εκτέλεση του αντίστοιχου αλγόριθμου κρυπτογράφησης του input string. Μόλις δημιουργηθεί το αντίστοιχο MD5 hash string, συγκρίνεται με αυτό της παραμέτρου hashString.

Σύνταξη:

VB :

```
Public Shared Function VerifyMD5Hash(  
    md5 As MD5,  
    inputString As String,  
    hashString As String  
)  
    As Boolean
```

Παράμετροι:

md5

Τύπος: System.Security.Cryptography.MD5
Το αντικείμενο MD5, για την εκτέλεση του αλγόριθμου MD5 επί του input string.

inputString

Τύπος: System.String
Το string εισόδου, βάσει του οποίου ο αλγόριθμος MD5 εσωτερικά θα δημιουργήσει το hash string. Αυτό, θα συγκριθεί με εκείνο της παραμέτρου hashString.

hashString

Τύπος: System.String
Το MD5 hash string προς σύγκριση.

Επιστρεφόμενη τιμή

Τύπος: System.Boolean
Επιστρέφει True, αν η σύγκριση δώσει ισότητα.

Παράδειγμα:

Παρακάτω, στον XAML κώδικα, έχουμε ένα PasswordBox element σε μία Windows φόρμα, στο οποίο ο χρήστης πληκτρολογεί τον κωδικό του. Φυσικά, αυτός εμφανίζεται με κουκκίδες. Το περιεχόμενο του PasswordBox είναι το inputString που θέλουμε. Στον VB κώδικα, μέσω της μεθόδου VerifyMD5Hash, συγκρίνουμε το παραγόμενο MD5 hash key με αυτό που έχουμε αποθηκευμένο σε ένα αρχείο "Password.txt" κι εμφανίζουμε το αποτέλεσμα της σύγκρισης στην οθόνη.

XAML :

```
<PasswordBox Name="txtLoginCode" VerticalAlignment="Center" Width="100"/>
```

VB :

```
Imports Zeus.WPF.Classes.Utilities.SecurityUtilities
Imports System.Security.Cryptography

...

Dim md5 As MD5 = MD5.Create
Dim hashKeyFromFile As String = String.Empty

'Πάρε το hash key από το αρχείο.
Using sReader As New IO.StreamReader("Password.txt")

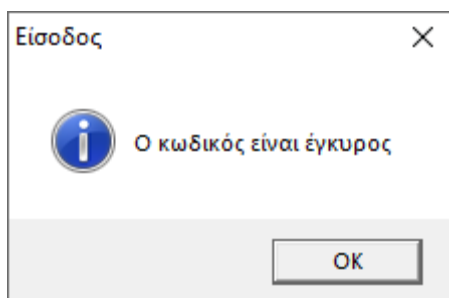
    hashKeyFromFile = sReader.ReadToEnd
    sReader.Close()

End Using

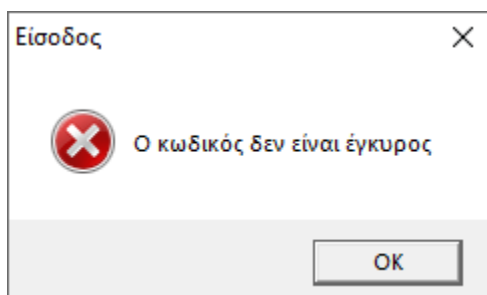
Dim isValid As Boolean = VerifyMD5Hash(md5, inputString, hashKeyFromFile)

Dim strMessage As String = If(isValid, "Ο κωδικός είναι έγκυρος", _
                              "Ο κωδικός δεν είναι έγκυρος")
Dim iconMessage As MessageBoxImage = If(isValid, MessageBoxImage.Information, _
                                         MessageBoxImage.Error)
MessageBox.Show(strMessage, "Είσοδος", MessageBoxButton.OK, iconMessage)
```

- Αν δοθεί σωστός κωδικός (το hash key του inputString συμφωνεί με αυτό του αρχείου):



- Αν δοθεί λάθος κωδικός (το hash key του inputString δεν συμφωνεί με αυτό του αρχείου):



Κρυπτογράφηση SHA-1

Όνομα	Περιγραφή
GetSHA1Hash (inputString As String) As String	<p>Επιστρέφει ένα SHA1 hash string, το οποίο δημιουργείται με βάση το input string.</p> <p>Χρησιμοποιεί εσωτερικά ένα SHA1 αντικείμενο για την εκτέλεση του αντίστοιχου αλγόριθμου κρυπτογράφησης. Το επιστρεφόμενο SHA-1 hash string είναι 40 χαρακτήρων δεκαεξαδικής μορφής.</p>
GetSHA1Hash (sha1Hash As SHA1, inputString As String) As String	<p>Επιστρέφει ένα SHA1 hash string, το οποίο δημιουργείται με βάση το αντικείμενο SHA1 και το input string.</p> <p>Το επιστρεφόμενο SHA-1 hash string είναι 40 χαρακτήρων δεκαεξαδικής μορφής.</p>
VerifySHA1Hash (inputString As String, hashString As String) As Boolean	<p>Επιστρέφει True, αν το SHA-1 hash του input string είναι το ίδιο με το hashString.</p> <p>Χρησιμοποιεί εσωτερικά ένα SHA1 αντικείμενο για την εκτέλεση του αντίστοιχου αλγόριθμου κρυπτογράφησης του input string. Μόλις δημιουργηθεί το αντίστοιχο SHA-1 hash string, συγκρίνεται με αυτό της παραμέτρου hashString.</p>
VerifySHA1Hash (sha1Hash As SHA1, inputString As String, hashString As String) As Boolean	<p>Επιστρέφει True, αν το SHA-1 hash του input string είναι το ίδιο με το hashString.</p> <p>Χρησιμοποιεί το SHA1 αντικείμενο της παραμέτρου για την εκτέλεση του αντίστοιχου αλγόριθμου κρυπτογράφησης του input string. Μόλις δημιουργηθεί το αντίστοιχο SHA-1 hash string, συγκρίνεται με αυτό της παραμέτρου hashString.</p>

GetSHA1Hash(inputString)

Επιστρέφει ένα SHA1 hash string, το οποίο δημιουργείται με βάση το **input string**. Χρησιμοποιεί εσωτερικά ένα SHA1 αντικείμενο για την εκτέλεση του αντίστοιχου αλγόριθμου κρυπτογράφησης. Το επιστρεφόμενο SHA-1 hash string είναι 40 χαρακτήρων δεκαεξαδικής μορφής.

Σύνταξη:

VB :

```
Public Shared Function GetSHA1Hash(  
    inputString As String  
) As String
```

Παράμετροι:

inputString

Τύπος: System.String

Το string εισόδου, βάσει του οποίου ο αλγόριθμος SHA-1 θα δημιουργήσει το hash string.

Επιστρεφόμενη τιμή

Τύπος: System.String

Επιστρέφει το SHA-1 hash string, 40 χαρακτήρων δεκαεξαδικής μορφής.

Παράδειγμα:

Παρακάτω, δίνουμε δημιουργούμε ένα SHA-1 hash string για το input string "mouratx".

VB :

```
Imports Zeus.WPF.Classes.Utilities.SecurityUtilities
```

```
Dim sha1Hash As String = GetSHA1Hash("mouratx")
```

Το εξαγόμενο hash string είναι το εξής:

```
598d58e28411e224fd1290194048d6b904c8d4dd
```

Το hash string μπορεί να αποθηκευτεί σε κάποιο αρχείο ή σε ένα database πεδίο, το οποίο να αντιστοιχεί σε κάποιο *password*. Όταν σε κατοπινή στιγμή ζητείται το password μπορούμε να πάρουμε το input string που θα δώσει ο χρήστης και να το συγκρίνουμε με αυτό το hash string, μέσω της μεθόδου VerifySHA1Hash.

GetSHA1Hash(*sha1*, *inputString*)

Επιστρέφει ένα **SHA1 hash string**, το οποίο δημιουργείται με βάση το **αντικείμενο SHA1** και το **input string**. Το επιστρεφόμενο SHA-1 hash string είναι 40 χαρακτήρων δεκαεξαδικής μορφής.

Σύνταξη:

VB :

```
Public Shared Function GetSHA1Hash(  
    sha1 As SHA1,  
    inputString As String  
)  
    As String
```

Παράμετροι:

sha1

Τύπος: System.Security.Cryptography.SHA1
Το αντικείμενο SHA1, για την εκτέλεση του αλγορίθμου SHA-1 επί του input string.

inputString

Τύπος: System.String
Το string εισόδου, βάσει του οποίου ο αλγόριθμος SHA-1 θα δημιουργήσει το hash string.

Επιστρεφόμενη τιμή

Τύπος: System.String
Επιστρέφει το SHA-1 hash string, 40 χαρακτήρων δεκαεξαδικής μορφής.

Παράδειγμα:

Παρακάτω, δίνουμε δημιουργούμε ένα SHA-1 hash string για το input string "mouratx".

VB :

```
Imports Zeus.WPF.Classes.Utilities.SecurityUtilities  
Imports System.Security.Cryptography  
  
Dim sha1 As SHA1 = SHA1.Create  
Dim sha1Hash As String = GetSHA1Hash(sha1, "mouratx")
```

Το εξαγόμενο hash string είναι το εξής:

598d58e28411e224fd1290194048d6b904c8d4dd

Το hash string μπορεί να αποθηκευτεί σε κάποιο αρχείο ή σε ένα database πεδίο, το οποίο να αντιστοιχεί σε κάποιο *password*. Όταν σε κατοπινή στιγμή ζητείται το password μπορούμε να πάρουμε το input string που θα δώσει ο χρήστης και να το συγκρίνουμε με αυτό το hash string, μέσω της μεθόδου `VerifySHA1Hash`.

VerifySHA1Hash(inputString, hashString)

Επιστρέφει True, αν το SHA-1 hash του **input string** είναι το ίδιο με το **hashString**. Χρησιμοποιεί εσωτερικά ένα SHA1 αντικείμενο για την εκτέλεση του αντίστοιχου αλγόριθμου κρυπτογράφησης του input string. Μόλις δημιουργηθεί το αντίστοιχο SHA-1 hash string, συγκρίνεται με αυτό της παραμέτρου hashString.

Σύνταξη:

VB :

```
Public Shared Function VerifySHA1Hash(  
    inputString As String,  
    hashString As String  
)  
    As Boolean
```

Παράμετροι:

inputString

Τύπος: System.String

Το string εισόδου, βάσει του οποίου ο αλγόριθμος SHA-1 εσωτερικά θα δημιουργήσει το hash string. Αυτό, θα συγκριθεί με εκείνο της παραμέτρου hashString.

hashString

Τύπος: System.String

Το SHA-1 hash string προς σύγκριση.

Επιστρεφόμενη τιμή

Τύπος: System.Boolean

Επιστρέφει True, αν η σύγκριση δώσει ισότητα.

Παράδειγμα:

Παρακάτω, στον XAML κώδικα, έχουμε ένα PasswordBox element σε μία Windows φόρμα, στο οποίο ο χρήστης πληκτρολογεί τον κωδικό του. Φυσικά, αυτός εμφανίζεται με κουκκίδες. Το περιεχόμενο του PasswordBox είναι το inputString που θέλουμε. Στον VB κώδικα, μέσω της μεθόδου VerifySHA1Hash, συγκρίνουμε το παραγόμενο SHA-1 hash key με αυτό που έχουμε αποθηκευμένο σε ένα αρχείο "Password.txt" κι εμφανίζουμε το αποτέλεσμα της σύγκρισης στην οθόνη.

XAML :

```
<PasswordBox Name="txtLoginCode" VerticalAlignment="Center" Width="100"/>
```

VB :

```
Imports Zeus.WPF.Classes.Utilities.SecurityUtilities
```

...

```
Dim hashKeyFromFile As String = String.Empty

'Πάρε το hash key από το αρχείο.
Using sReader As New IO.StreamReader("Password.txt")

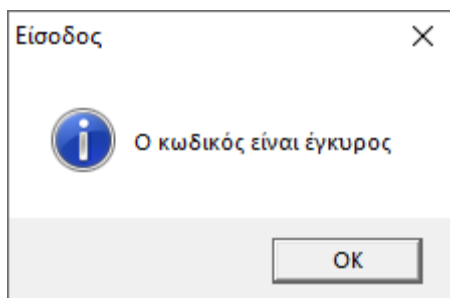
    hashKeyFromFile = sReader.ReadToEnd
    sReader.Close()

End Using

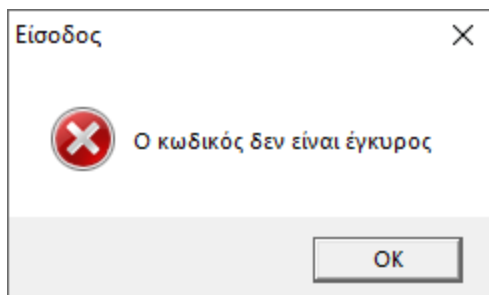
Dim isValid As Boolean = VerifySHA1Hash(inputString, hashKeyFromFile)

Dim strMessage As String = If(isValid, "Ο κωδικός είναι έγκυρος", _
    "Ο κωδικός δεν είναι έγκυρος")
Dim iconMessage As MessageBoxIcon = If(isValid, MessageBoxIcon.Information, _
    MessageBoxIcon.Error)
MessageBox.Show(strMessage, "Είσοδος", MessageBoxButtons.OK, iconMessage)
```

- Αν δοθεί σωστός κωδικός (το hash key του inputString συμφωνεί με αυτό του αρχείου):



- Αν δοθεί λάθος κωδικός (το hash key του inputString δεν συμφωνεί με αυτό του αρχείου):



VerifySHA1Hash(*sha1*, *inputString*, *hashString*)

Επιστρέφει True, αν το SHA-1 hash του **input string** είναι το ίδιο με το **hashString**. Χρησιμοποιεί το SHA1 αντικείμενο της παραμέτρου για την εκτέλεση του αντίστοιχου αλγόριθμου κρυπτογράφησης του input string. Μόλις δημιουργηθεί το αντίστοιχο SHA-1 hash string, συγκρίνεται με αυτό της παραμέτρου hashString.

Σύνταξη:

VB :

```
Public Shared Function VerifySHA1Hash(  
    sha1 As SHA1,  
    inputString As String,  
    hashString As String  
)  
    As Boolean
```

Παράμετροι:

sha1

Τύπος: System.Security.Cryptography.SHA1
Το αντικείμενο SHA1, για την εκτέλεση του αλγορίθμου SHA-1 επί του input string.

inputString

Τύπος: System.String
Το string εισόδου, βάσει του οποίου ο αλγόριθμος SHA-1 εσωτερικά θα δημιουργήσει το hash string. Αυτό, θα συγκριθεί με εκείνο της παραμέτρου hashString.

hashString

Τύπος: System.String
Το SHA-1 hash string προς σύγκριση.

Επιστρεφόμενη τιμή

Τύπος: System.Boolean
Επιστρέφει True, αν η σύγκριση δώσει ισότητα.

Παράδειγμα:

Παρακάτω, στον XAML κώδικα, έχουμε ένα PasswordBox element σε μία Windows φόρμα, στο οποίο ο χρήστης πληκτρολογεί τον κωδικό του. Φυσικά, αυτός εμφανίζεται με κουκκίδες. Το περιεχόμενο του PasswordBox είναι το inputString που θέλουμε. Στον VB κώδικα, μέσω της μεθόδου VerifySHA1Hash, συγκρίνουμε το παραγόμενο SHA-1 hash key με αυτό που έχουμε αποθηκευμένο σε ένα αρχείο "Password.txt" κι εμφανίζουμε το αποτέλεσμα της σύγκρισης στην οθόνη.

XAML :

```
<PasswordBox Name="txtLoginCode" VerticalAlignment="Center" Width="100"/>
```

VB :

```
Imports Zeus.WPF.Classes.Utilities.SecurityUtilities
Imports System.Security.Cryptography

...

Dim sha1 As SHA1 = SHA1.Create
Dim hashKeyFromFile As String = String.Empty

'Πάρε το hash key από το αρχείο.
Using sReader As New IO.StreamReader("Password.txt")

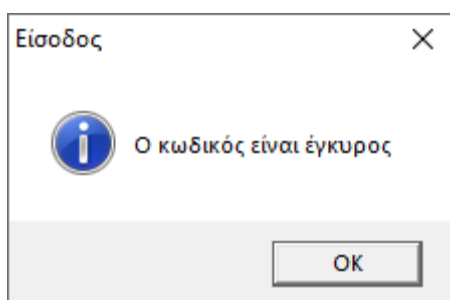
    hashKeyFromFile = sReader.ReadToEnd
    sReader.Close()

End Using

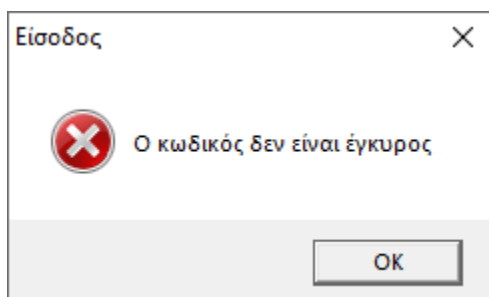
Dim isValid As Boolean = VerifySHA1Hash(sha1, inputString, hashKeyFromFile)

Dim strMessage As String = If(isValid, "Ο κωδικός είναι έγκυρος", _
                              "Ο κωδικός δεν είναι έγκυρος")
Dim iconMessage As MessageBoxImage = If(isValid, MessageBoxImage.Information, _
                                         MessageBoxImage.Error)
MessageBox.Show(strMessage, "Είσοδος", MessageBoxButton.OK, iconMessage)
```

- Αν δοθεί σωστός κωδικός (το hash key του inputString συμφωνεί με αυτό του αρχείου):



- Αν δοθεί λάθος κωδικός (το hash key του inputString δεν συμφωνεί με αυτό του αρχείου):





WindowUtilities

Περιλαμβάνει μία συλλογή από στατικές μεθόδους σχετικές με **διαχείριση παραθύρων (windows)**.

Σύνταξη:

VB:

```
Public Class WindowUtilities
```

Ο παρακάτω πίνακας περιέχει τις διαθέσιμες **κατηγορίες**:

Κατηγορία	Περιγραφή
Mouse Cursor	Περιλαμβάνει στατικές μεθόδους σχετικές με το mouse cursor .
Εξειδικευμένα παράθυρα	Περιλαμβάνει στατικές μεθόδους για εξειδικευμένα παράθυρα .

Απαριθμήσεις

Όνομα	Περιγραφή
ShowMessageButton	Περιλαμβάνει προκαθορισμένες τιμές για το ποιό/ποιά buttons θα εμφανίζονται στην μέθοδο ShowImage.
ShowMessageImage	Περιλαμβάνει προκαθορισμένες τιμές για το ποιό icon θα εμφανίζεται στην μέθοδο ShowImage.
ShowMessageResult	Περιλαμβάνει προκαθορισμένες τιμές για το ποιό αποτέλεσμα επιστρέφει η μέθοδος ShowImage.

Mouse Cursor

Όνομα	Περιγραφή
SetNormalCursor()	Θέτει το normal icon στον δείκτη του ποντικιού. Την κλήση την κάνουμε μετά την εκτέλεση κάποιων χρονοβόρων εντολών.
SetWaitCursor()	Θέτει το wait icon στον δείκτη του ποντικιού. Την κλήση την κάνουμε πριν την εκτέλεση κάποιων χρονοβόρων εντολών.

SetNormalCursor()

Θέτει το **normal icon** στον δείκτη του ποντικιού. Την κλήση την κάνουμε μετά την εκτέλεση κάποιων χρονοβόρων εντολών.

Σύνταξη:

VB :

```
Public Shared Sub SetNormalCursor()
```

Παράδειγμα:

VB :

```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities  
  
SetWaitCursor()  
  
'... μία χρονοβόρα διαδικασία...  
  
SetNormalCursor()
```

SetWaitCursor()

Θέτει το **wait icon** στον δείκτη του ποντικιού. Την κλήση την κάνουμε πριν την εκτέλεση κάποιων χρονοβόρων εντολών.

Σύνταξη:

VB :

```
Public Shared Sub SetWaitCursor()
```

Παράδειγμα:

VB :

```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities  
  
SetWaitCursor()  
  
'... μία χρονοβόρα διαδικασία...  
  
SetNormalCursor()
```

Εξειδικευμένα παράθυρα

Ο παρακάτω πίνακας περιέχει τα διαθέσιμα **εξειδικευμένα παράθυρα**:

Κατηγορία	Περιγραφή
InputMessage <i>(νέο στην έκδοση 1.1)</i>	Το αντίστοιχο στο WPF της κλασικής μεθόδου InputBox . Εμφανίζει ένα παράθυρο με prompt για εισαγωγή δεδομένου.
ShowMessage <i>(νέο στην έκδοση 1.1)</i>	Το αντίστοιχο στο WPF της κλασικής μεθόδου MessageBox.Show . Εμφανίζει ένα παράθυρο μηνύματος με πιθανές απαντήσεις OK/Ακύρωση ή ερώτησης με πιθανές απαντήσεις Ναι/Όχι.
WaitWindow	Περιλαμβάνει στατικές μεθόδους σχετικές με τη διαχείριση ενός modeless παραθύρου καθυστέρησης (wait window).

InputMessage

Όνομα	Περιγραφή
InputMessage (prompt As String, Optional leftPos As Double = -1, Optional topPos As Double = -1)	<p>Εμφανίζει ένα παράθυρο εισόδου δεδομένου που έχει ένα προτρεπτικό μήνυμα.</p> <p>Επιστρέφει την απάντηση που έδωσε ο χρήστης ή κενό αν ακύρωσε. Προαιρετικά, το παράθυρο ανοίγει στη θέση της οθόνης που προσδιορίζεται από τις leftPos και topPos.</p>
InputMessage (prompt As String, title As String, Optional leftPos As Double = -1, Optional topPos As Double = -1)	<p>Εμφανίζει ένα παράθυρο εισόδου δεδομένου που έχει ένα προτρεπτικό μήνυμα κι έναν τίτλο.</p> <p>Επιστρέφει την απάντηση που έδωσε ο χρήστης ή κενό αν ακύρωσε. Προαιρετικά, το παράθυρο ανοίγει στη θέση της οθόνης που προσδιορίζεται από τις leftPos και topPos.</p>
InputMessage (prompt As String, title As String, defaultResponse As String, Optional leftPos As Double = -1, Optional topPos As Double = -1)	<p>Εμφανίζει ένα παράθυρο εισόδου δεδομένου που έχει ένα προτρεπτικό μήνυμα κι έναν τίτλο. Στο input textbox εμφανίζεται μία προκαθορισμένη απάντηση.</p> <p>Επιστρέφει την απάντηση που έδωσε ο χρήστης ή κενό αν ακύρωσε. Προαιρετικά, το παράθυρο ανοίγει στη θέση της οθόνης που προσδιορίζεται από τις leftPos και topPos.</p>
InputMessage (prompt As String, title As String, defaultResponse As String, icon As Image, Optional leftPos As Double = -1, Optional topPos As Double = -1)	<p>Εμφανίζει ένα παράθυρο εισόδου δεδομένου που έχει ένα προτρεπτικό μήνυμα κι έναν τίτλο. Στο input textbox εμφανίζεται μία προκαθορισμένη απάντηση. Επίσης, εμφανίζεται και μία εικόνα μεγέθους 48x48px αριστερά του προτρεπτικού μηνύματος.</p> <p>Επιστρέφει την απάντηση που έδωσε ο χρήστης ή κενό αν ακύρωσε. Προαιρετικά, το παράθυρο ανοίγει στη θέση της</p>

	οθόνης που προσδιορίζεται από τις leftPos και topPos.
InputMessage (prompt As String, title As String, defaultResponse As String, icon As Image, cornerRadius As CornerRadius, Optional leftPos As Double = -1, Optional topPos As Double = -1)	<p>Εμφανίζει ένα παράθυρο εισόδου δεδομένου που έχει ένα προτρεπτικό μήνυμα κι έναν τίτλο. Στο input textbox εμφανίζεται μία προκαθορισμένη απάντηση. Επίσης, εμφανίζεται και μία εικόνα μεγέθους 48x48px αριστερά του προτρεπτικού μηνύματος. Προσδιορίζουμε και μία στρογγυλότητα στις γωνίες του παραθύρου.</p> <p>Επιστρέφει την απάντηση που έδωσε ο χρήστης ή κενό αν ακύρωσε. Προαιρετικά, το παράθυρο ανοίγει στη θέση της οθόνης που προσδιορίζεται από τις leftPos και topPos.</p>
InputMessage (prompt As String, title As String, defaultResponse As String, icon As Image, cornerRadius As CornerRadius, width As Double, height As Double, Optional leftPos As Double = -1, Optional topPos As Double = -1)	<p>Εμφανίζει ένα παράθυρο εισόδου δεδομένου που έχει ένα προτρεπτικό μήνυμα κι έναν τίτλο. Στο input textbox εμφανίζεται μία προκαθορισμένη απάντηση. Επίσης, εμφανίζεται και μία εικόνα μεγέθους 48x48px αριστερά του προτρεπτικού μηνύματος. Προσδιορίζουμε τη στρογγυλότητα στις γωνίες του παραθύρου καθώς και το πλάτος/ύψος.</p> <p>Επιστρέφει την απάντηση που έδωσε ο χρήστης ή κενό αν ακύρωσε. Προαιρετικά, το παράθυρο ανοίγει στη θέση της οθόνης που προσδιορίζεται από τις leftPos και topPos.</p>

<p>InputMessage(prompt As String, title As String, defaultResponse As String, icon As Image, cornerRadius As CornerRadius, width As Double, height As Double, foreground As Brush, background As Brush, Optional leftPos As Double = -1, Optional topPos As Double = -1)</p>	<p>Εμφανίζει ένα παράθυρο εισόδου δεδομένου που έχει ένα προτροπικό μήνυμα κι έναν τίτλο. Στο input textbox εμφανίζεται μία προκαθορισμένη απάντηση. Επίσης, εμφανίζεται και μία εικόνα μεγέθους 48x48px αριστερά του προτροπικού μηνύματος. Προσδιορίζουμε τη στρογγυλότητα στις γωνίες του παραθύρου καθώς και το πλάτος/ύψος. Καθορίζουμε το χρώμα μελάνης του προτροπικού μηνύματος και το χρώμα φόντου του παραθύρου.</p> <p>Επιστρέφει την απάντηση που έδωσε ο χρήστης ή κενό αν ακύρωσε. Προαιρετικά, το παράθυρο ανοίγει στη θέση της οθόνης που προσδιορίζεται από τις leftPos και topPos.</p>
<p>InputMessage(prompt As String, title As String, defaultResponse As String, icon As Image, cornerRadius As CornerRadius, width As Double, height As Double, foreground As Brush, background As Brush, titleStyle As Style, promptStyle As Style, inputTextStyle As Style, Optional leftPos As Double = -1, Optional topPos As Double = -1)</p>	<p>Εμφανίζει ένα παράθυρο εισόδου δεδομένου που έχει ένα προτροπικό μήνυμα κι έναν τίτλο. Στο input textbox εμφανίζεται μία προκαθορισμένη απάντηση. Επίσης, εμφανίζεται και μία εικόνα μεγέθους 48x48px αριστερά του προτροπικού μηνύματος. Προσδιορίζουμε τη στρογγυλότητα στις γωνίες του παραθύρου καθώς και το πλάτος/ύψος. Καθορίζουμε το χρώμα μελάνης του προτροπικού μηνύματος και το χρώμα φόντου του παραθύρου. Καθορίζουμε style για τον τίτλο, το προτροπικό μήνυμα και το input textbox.</p> <p>Επιστρέφει την απάντηση που έδωσε ο χρήστης ή κενό αν ακύρωσε. Προαιρετικά, το παράθυρο ανοίγει στη θέση της οθόνης που προσδιορίζεται από τις leftPos και topPos.</p>

<p>InputMessage(prompt As String, title As String, defaultResponse As String, icon As Image, cornerRadius As CornerRadius, width As Double, height As Double, foreground As Brush, background As Brush, titleStyle As Style, promptStyle As Style, inputTextStyle As Style, okButtonStyle As Style, cancelButtonStyle As Style, Optional leftPos As Double = -1, Optional topPos As Double = -1)</p>	<p>Εμφανίζει ένα παράθυρο εισόδου δεδομένου που έχει ένα προτρεπτικό μήνυμα κι έναν τίτλο. Στο input textbox εμφανίζεται μία προκαθορισμένη απάντηση. Επίσης, εμφανίζεται και μία εικόνα μεγέθους 48x48px αριστερά του προτρεπτικού μηνύματος. Προσδιορίζουμε τη στρογγυλότητα στις γωνίες του παραθύρου καθώς και το πλάτος/ύψος. Καθορίζουμε το χρώμα μελάνης του προτρεπτικού μηνύματος και το χρώμα φόντου του παραθύρου. Καθορίζουμε style για τον τίτλο, το προτρεπτικό μήνυμα, το input textbox και για τα buttons OK και Cancel.</p> <p>Επιστρέφει την απάντηση που έδωσε ο χρήστης ή κενό αν ακύρωσε. Προαιρετικά, το παράθυρο ανοίγει στη θέση της οθόνης που προσδιορίζεται από τις leftPos και topPos.</p>
---	--

InputMessage(prompt, Optional leftPos, Optional topPos)

Εμφανίζει ένα παράθυρο εισόδου δεδομένου που έχει ένα προτρεπτικό μήνυμα. Επιστρέφει την απάντηση που έδωσε ο χρήστης ή κενό αν ακύρωσε.

Προαιρετικά, το παράθυρο ανοίγει στη θέση της οθόνης που προσδιορίζεται από τις leftPos και topPos.

Σύνταξη:

VB :

```
Public Shared Function InputMessage(  
    prompt As String,  
    Optional leftPos As Double = -1,  
    Optional topPos As Double = -1  
) As String
```

Παράμετροι:

prompt

Τύπος: System.String
Το προτρεπτικό μήνυμα.

Optional leftPos

Τύπος: System.Double
Καθορίζει τη θέση του αριστερού άκρου του παραθύρου σε λογικά pixels.

Optional topPos

Τύπος: System.Double
Καθορίζει τη θέση του άνω άκρου του παραθύρου σε λογικά pixels.

Παρατηρήσεις:

Το παράθυρο που ανοίγει είναι **modal**. Εμφανίζεται στο κέντρο της οθόνης εκτός αν προσδιορίσουμε κάποια από τις παραμέτρους leftPos, topPos.

Παράδειγμα 1:

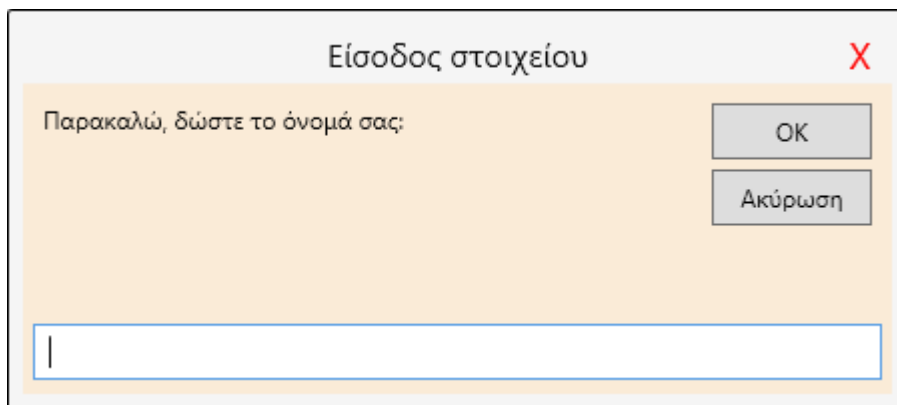
Παρακάτω, στον VB κώδικα, εμφανίζουμε το προτρεπτικό μήνυμα "Παρακαλώ, δώστε το όνομά σας". Τα υπόλοιπα στοιχεία του παραθύρου είναι προκαθορισμένα:

VB :

```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities
```

```
...
```

```
Dim strAnswer As String = InputMessage("Παρακαλώ, δώστε το όνομά σας: ")
```



Παράδειγμα 2:

Παρακάτω, στον VB κώδικα, εμφανίζουμε παράθυρο 200px αριστερά από το άκρο της οθόνης. Επειδή, δεν προσδιορίζουμε την παράμετρο `topPos` αυτό θα εμφανιστεί στο κέντρο κατακόρυφα:

```
Dim strAnswer As String = InputMessage("Παρακαλώ, δώστε το όνομά σας: ", _  
                                     leftPos:=200)
```

InputMessage(prompt, title, Optional leftPos, Optional topPos)

Εμφανίζει ένα παράθυρο εισόδου δεδομένου που έχει ένα προτρεπτικό μήνυμα κι έναν τίτλο. Επιστρέφει την απάντηση που έδωσε ο χρήστης ή κενό αν ακύρωσε.

Προαιρετικά, το παράθυρο ανοίγει στη θέση της οθόνης που προσδιορίζεται από τις leftPos και topPos.

Σύνταξη:

VB :

```
Public Shared Function InputMessage(  
    prompt As String,  
    title As String,  
    Optional leftPos As Double = -1,  
    Optional topPos As Double = -1  
    ) As String
```

Παράμετροι:

prompt

Τύπος: System.String
Το προτρεπτικό μήνυμα.

title

Τύπος: System.String
Ο τίτλος του παραθύρου. Η default τιμή είναι "Εισαγωγή δεδομένου".

Optional leftPos

Τύπος: System.Double
Καθορίζει τη θέση του αριστερού άκρου του παραθύρου σε λογικά pixels.

Optional topPos

Τύπος: System.Double
Καθορίζει τη θέση του άνω άκρου του παραθύρου σε λογικά pixels.

Παρατηρήσεις:

Το παράθυρο που ανοίγει είναι **modal**. Εμφανίζεται στο κέντρο της οθόνης εκτός αν προσδιορίσουμε κάποια από τις παραμέτρους leftPos, topPos.

Παράδειγμα:

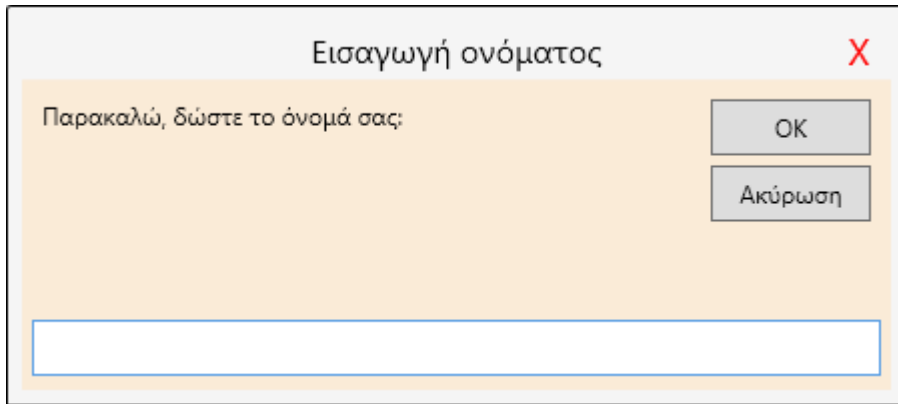
Παρακάτω, στον VB κώδικα, εμφανίζουμε το προτρεπτικό μήνυμα "Παρακαλώ, δώστε το όνομά σας", με τίτλο "Εισαγωγή ονόματος". Τα υπόλοιπα στοιχεία του παραθύρου είναι προκαθορισμένα:

VB :

```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities
```

...

```
Dim strAnswer As String = InputMessage("Παρακαλώ, δώστε το όνομά σας: ", _  
"Εισαγωγή ονόματος")
```



InputMessage(prompt, title, defaultResponse, Optional leftPos, Optional topPos)

Εμφανίζει ένα παράθυρο εισόδου δεδομένου που έχει ένα προτροπτικό μήνυμα κι έναν τίτλο. Στο input textbox εμφανίζεται μία προκαθορισμένη απάντηση.

Επιστρέφει την απάντηση που έδωσε ο χρήστης ή κενό αν ακύρωσε.

Προαιρετικά, το παράθυρο ανοίγει στη θέση της οθόνης που προσδιορίζεται από τις leftPos και topPos.

Σύνταξη:

VB :

```
Public Shared Function InputMessage(  
    prompt As String,  
    title As String,  
    defaultResponse As String,  
    Optional leftPos As Double = -1,  
    Optional topPos As Double = -1  
    ) As String
```

Παράμετροι:

prompt

Τύπος: System.String
Το προτροπτικό μήνυμα.

title

Τύπος: System.String
Ο τίτλος του παραθύρου. Η default τιμή είναι "Εισαγωγή δεδομένου".

defaultResponse

Τύπος: System.String
Εμφανίζει μία προκαθορισμένη απάντηση στο input textbox. Η default τιμή είναι String.Empty.

Optional leftPos

Τύπος: System.Double
Καθορίζει τη θέση του αριστερού άκρου του παραθύρου σε λογικά pixels.

Optional topPos

Τύπος: System.Double
Καθορίζει τη θέση του άνω άκρου του παραθύρου σε λογικά pixels.

Παρατηρήσεις:

Το παράθυρο που ανοίγει είναι **modal**. Εμφανίζεται στο κέντρο της οθόνης εκτός αν προσδιορίσουμε κάποια από τις παραμέτρους leftPos, topPos.

Παράδειγμα:

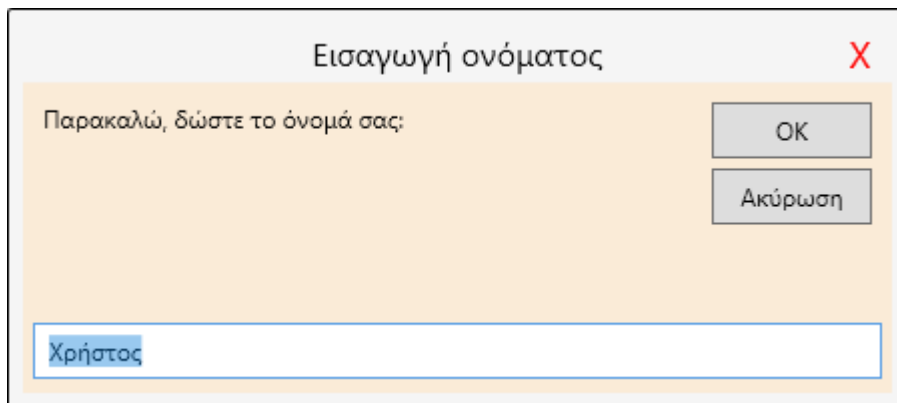
Παρακάτω, στον VB κώδικα, εμφανίζουμε το προτρεπτικό μήνυμα "Παρακαλώ, δώστε το όνομά σας", με τίτλο "Εισαγωγή ονόματος". Στο input textbox εμφανίζεται η προκαθορισμένη απάντηση "Χρήστος". Τα υπόλοιπα στοιχεία του παραθύρου είναι προκαθορισμένα:

VB :

```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities
```

```
...
```

```
Dim strAnswer As String = InputMessage("Παρακαλώ, δώστε το όνομά σας: ",  
                                         "Εισαγωγή ονόματος", "Χρήστος")
```



InputMessage(prompt, title, defaultResponse, icon, Optional leftPos, Optional topPos)

Εμφανίζει ένα παράθυρο εισόδου δεδομένου που έχει ένα προτροπτικό μήνυμα κι έναν τίτλο. Στο input textbox εμφανίζεται μία προκαθορισμένη απάντηση. Επίσης, εμφανίζεται και μία εικόνα μεγέθους 48x48px αριστερά του προτροπτικού μηνύματος. Επιστρέφει την απάντηση που έδωσε ο χρήστης ή κενό αν ακύρωσε.

Προαιρετικά, το παράθυρο ανοίγει στη θέση της οθόνης που προσδιορίζεται από τις leftPos και topPos.

Σύνταξη:

VB :

```
Public Shared Function InputMessage(  
    prompt As String,  
    title As String,  
    defaultResponse As String,  
    icon As Image,  
    Optional leftPos As Double = -1,  
    Optional topPos As Double = -1  
    ) As String
```

Παράμετροι:

prompt

Τύπος: System.String
Το προτροπτικό μήνυμα.

title

Τύπος: System.String
Ο τίτλος του παραθύρου. Η default τιμή είναι "Εισαγωγή δεδομένου".

defaultResponse

Τύπος: System.String
Εμφανίζει μία προκαθορισμένη απάντηση στο input textbox. Η default τιμή είναι String.Empty.

icon

Τύπος: System.Windows.Controls.Image
Εμφανίζει μία εικόνα ως icon μεγέθους 48x48px αριστερά του προτροπτικού μηνύματος. Η default τιμή είναι Nothing.

Optional leftPos

Τύπος: System.Double
Καθορίζει τη θέση του αριστερού άκρου του παραθύρου σε λογικά pixels.

Optional topPos

Τύπος: System.Double
Καθορίζει τη θέση του άνω άκρου του παραθύρου σε λογικά pixels.

Παρατηρήσεις:

Το παράθυρο που ανοίγει είναι **modal**. Εμφανίζεται στο κέντρο της οθόνης εκτός αν προσδιορίσουμε κάποια από τις παραμέτρους `leftPos`, `topPos`.

Παράδειγμα:

Παρακάτω, στον VB κώδικα, εμφανίζουμε το προτρεπτικό μήνυμα "Παρακαλώ, δώστε το όνομά σας", με τίτλο "Εισαγωγή ονόματος". Στο input textbox εμφανίζεται η προκαθορισμένη απάντηση "Χρήστος". Στον φάκελο της εφαρμογής έχουμε τοποθετήσει ένα **αρχείο εικόνας** με όνομα "Person.png". Αυτήν την εικόνα εμφανίζουμε ως **εικονίδιο** στο παράθυρο. Τα υπόλοιπα στοιχεία του παραθύρου είναι προκαθορισμένα:

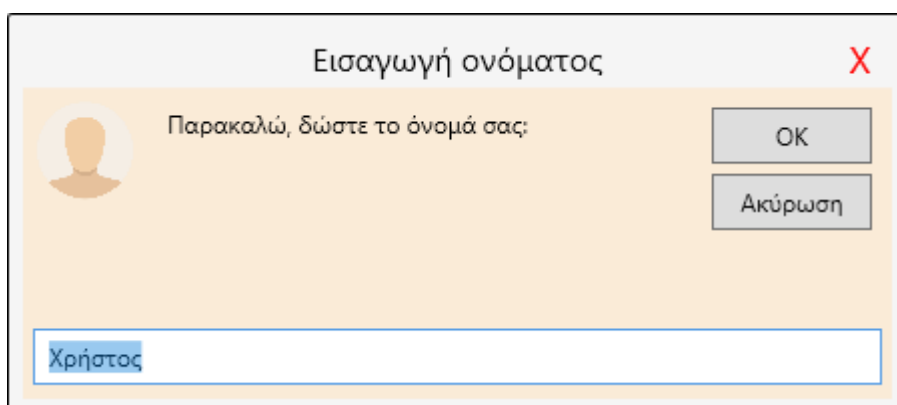
VB :

```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities

...

Dim bm As New BitmapImage(New Uri("Person.png", UriKind.Relative))
Dim img As New Image With {.Source = bm}

Dim strAnswer As String = InputMessage("Παρακαλώ, δώστε το όνομά σας: ", _
                                         "Εισαγωγή ονόματος", "Χρήστος", img)
```



InputMessage(prompt, title, defaultResponse, icon, cornerRadius, Optional leftPos, Optional topPos)

Εμφανίζει ένα παράθυρο εισόδου δεδομένου που έχει ένα προτρεπτικό μήνυμα κι έναν τίτλο. Στο input textbox εμφανίζεται μία προκαθορισμένη απάντηση. Επίσης, εμφανίζεται και μία εικόνα μεγέθους 48x48px αριστερά του προτρεπτικού μηνύματος. Προσδιορίζουμε και μία στρογγυλότητα στις γωνίες του παραθύρου.

Επιστρέφει την απάντηση που έδωσε ο χρήστης ή κενό αν ακύρωσε.

Προαιρετικά, το παράθυρο ανοίγει στη θέση της οθόνης που προσδιορίζεται από τις leftPos και topPos.

Σύνταξη:

VB :

```
Public Shared Function InputMessage(  
    prompt As String,  
    title As String,  
    defaultResponse As String,  
    icon As Image,  
    cornerRadius As CornerRadius,  
    Optional leftPos As Double = -1,  
    Optional topPos As Double = -1  
    ) As String
```

Παράμετροι:

prompt

Τύπος: System.String
Το προτρεπτικό μήνυμα.

title

Τύπος: System.String
Ο τίτλος του παραθύρου. Η default τιμή είναι "Εισαγωγή δεδομένου".

defaultResponse

Τύπος: System.String
Εμφανίζει μία προκαθορισμένη απάντηση στο input textbox. Η default τιμή είναι String.Empty.

icon

Τύπος: System.Windows.Controls.Image
Εμφανίζει μία εικόνα ως icon μεγέθους 48x48px αριστερά του προτρεπτικού μηνύματος. Η default τιμή είναι Nothing.

cornerRadius

Τύπος: System.Windows.CnerRadius
Καθορίζουμε μία στρογγυλότητα στις γωνίες του παραθύρου. Η default τιμή είναι CornerRadius(2,2,0,0).

Optional LeftPos

Τύπος: System.Double
Καθορίζει τη θέση του αριστερού άκρου του παραθύρου σε λογικά pixels.

Optional topPos

Τύπος: System.Double

Καθορίζει τη θέση του άνω άκρου του παραθύρου σε λογικά pixels.

Παρατηρήσεις:

Το παράθυρο που ανοίγει είναι **modal**. Εμφανίζεται στο κέντρο της οθόνης εκτός αν προσδιορίσουμε κάποια από τις παραμέτρους `leftPos`, `topPos`.

Παράδειγμα:

Παρακάτω, στον VB κώδικα, εμφανίζουμε το προτροπικό μήνυμα "Παρακαλώ, δώστε το όνομά σας", με τίτλο "Εισαγωγή ονόματος". Στον φάκελο της εφαρμογής έχουμε τοποθετήσει ένα **αρχείο εικόνας** με όνομα "Person.png". Αυτήν την εικόνα εμφανίζουμε ως **εικονίδιο** στο παράθυρο. Καθορίζουμε τις δύο άνω γωνίες του παραθύρου να είναι έντονα στρογγυλές. Τα υπόλοιπα στοιχεία του παραθύρου είναι προκαθορισμένα:

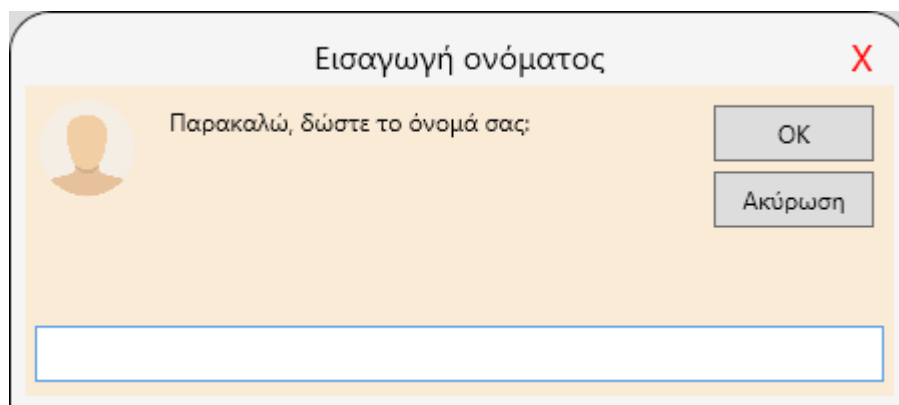
VB :

```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities

...

Dim bm As New BitmapImage(New Uri("Person.png", UriKind.Relative))
Dim img As New Image With {.Source = bm}

Dim strAnswer As String = InputMessage("Παρακαλώ, δώστε το όνομά σας: ", _
    "Εισαγωγή ονόματος", "", img, New CornerRadius(20, 20, 2, 2))
```



InputMessage(prompt, title, defaultResponse, icon, cornerRadius, width, height, Optional leftPos, Optional topPos)

Εμφανίζει ένα παράθυρο εισόδου δεδομένου που έχει ένα προτροπτικό μήνυμα κι έναν τίτλο. Στο input textbox εμφανίζεται μία προκαθορισμένη απάντηση. Επίσης, εμφανίζεται και μία εικόνα μεγέθους 48x48px αριστερά του προτροπτικού μηνύματος. Προσδιορίζουμε τη στρογγυλότητα στις γωνίες του παραθύρου καθώς και το πλάτος/ύψος.

Επιστρέφει την απάντηση που έδωσε ο χρήστης ή κενό αν ακύρωσε.

Προαιρετικά, το παράθυρο ανοίγει στη θέση της οθόνης που προσδιορίζεται από τις leftPos και topPos.

Σύνταξη:

VB :

```
Public Shared Function InputMessage(  
    prompt As String,  
    title As String,  
    defaultResponse As String,  
    icon As Image,  
    cornerRadius As CornerRadius,  
    width As Double,  
    height As Double,  
    Optional leftPos As Double = -1,  
    Optional topPos As Double = -1  
    ) As String
```

Παράμετροι:

prompt

Τύπος: System.String
Το προτροπτικό μήνυμα.

title

Τύπος: System.String
Ο τίτλος του παραθύρου. Η default τιμή είναι "Εισαγωγή δεδομένου".

defaultResponse

Τύπος: System.String
Εμφανίζει μία προκαθορισμένη απάντηση στο input textbox. Η default τιμή είναι String.Empty.

icon

Τύπος: System.Windows.Controls.Image
Εμφανίζει μία εικόνα ως icon μεγέθους 48x48px αριστερά του προτροπτικού μηνύματος. Η default τιμή είναι Nothing.

cornerRadius

Τύπος: System.Windows.CnerRadius
Καθορίζουμε μία στρογγυλότητα στις γωνίες του παραθύρου. Η default τιμή είναι CornerRadius(2,2,0,0).

width

Τύπος: System.Double

Καθορίζουμε το πλάτος του παραθύρου σε λογικά pixels. Η default τιμή είναι 450 pixels.

height

Τύπος: System.Double

Καθορίζουμε το ύψος του παραθύρου σε λογικά pixels. Η default τιμή είναι 200 pixels.

Optional LeftPos

Τύπος: System.Double

Καθορίζει τη θέση του αριστερού άκρου του παραθύρου σε λογικά pixels.

Optional topPos

Τύπος: System.Double

Καθορίζει τη θέση του άνω άκρου του παραθύρου σε λογικά pixels.

Παρατηρήσεις:

Το παράθυρο που ανοίγει είναι **modal**. Εμφανίζεται στο κέντρο της οθόνης εκτός αν προσδιορίσουμε κάποια από τις παραμέτρους leftPos, topPos.

Παράδειγμα:

Παρακάτω, στον VB κώδικα, εμφανίζουμε το προτρεπτικό μήνυμα "Παρακαλώ, δώστε το όνομά σας", με τίτλο "Εισαγωγή ονόματος". Στον φάκελο της εφαρμογής έχουμε τοποθετήσει ένα **αρχείο εικόνας** με όνομα "Person.png". Αυτήν την εικόνα εμφανίζουμε ως **εικονίδιο** στο παράθυρο. Καθορίζουμε τις δύο άνω γωνίες του παραθύρου να είναι έντονα στρογγυλές. Το πλάτος είναι 500px και το ύψος 180px. Τα υπόλοιπα στοιχεία του παραθύρου είναι προκαθορισμένα:

VB :


```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities
```

```
...
```

```
Dim bm As New BitmapImage(New Uri("Person.png", UriKind.Relative))  
Dim img As New Image With {.Source = bm}
```

```
Dim strAnswer As String = InputMessage("Παρακαλώ, δώστε το όνομά σας: ", _  
    "Εισαγωγή ονόματος", "", img, New CornerRadius(20, 20, 2, 2), _  
    width:=500, height:=180)
```


Εισαγωγή ονόματος X

 Παρακαλώ, δώστε το όνομά σας:

InputMessage(prompt, title, defaultResponse, icon, cornerRadius, width, height, foreground, background, Optional leftPos, Optional topPos)

Εμφανίζει ένα παράθυρο εισόδου δεδομένου που έχει ένα προτροπτικό μήνυμα κι έναν τίτλο. Στο input textbox εμφανίζεται μία προκαθορισμένη απάντηση. Επίσης, εμφανίζεται και μία εικόνα μεγέθους 48x48px αριστερά του προτροπτικού μηνύματος. Προσδιορίζουμε τη στρογγυλότητα στις γωνίες του παραθύρου καθώς και το πλάτος/ύψος. Καθορίζουμε το χρώμα μελάνης του προτροπτικού μηνύματος και το χρώμα φόντου του παραθύρου.

Επιστρέφει την απάντηση που έδωσε ο χρήστης ή κενό αν ακύρωσε.

Προαιρετικά, το παράθυρο ανοίγει στη θέση της οθόνης που προσδιορίζεται από τις leftPos και topPos.

Σύνταξη:

VB :

```
Public Shared Function InputMessage(  
    prompt As String,  
    title As String,  
    defaultResponse As String,  
    icon As Image,  
    cornerRadius As CornerRadius,  
    width As Double,  
    height As Double,  
    foreground As Brush,  
    background As Brush,  
    Optional leftPos As Double = -1,  
    Optional topPos As Double = -1  
    ) As String
```

Παράμετροι:

prompt

Τύπος: System.String
Το προτροπτικό μήνυμα.

title

Τύπος: System.String
Ο τίτλος του παραθύρου. Η default τιμή είναι "Εισαγωγή δεδομένου".

defaultResponse

Τύπος: System.String
Εμφανίζει μία προκαθορισμένη απάντηση στο input textbox. Η default τιμή είναι String.Empty.

icon

Τύπος: System.Windows.Controls.Image
Εμφανίζει μία εικόνα ως icon μεγέθους 48x48px αριστερά του προτροπτικού μηνύματος. Η default τιμή είναι Nothing.

cornerRadius

Τύπος: System.Windows.CornerRadius

Καθορίζουμε μία στρογγυλότητα στις γωνίες του παραθύρου. Η default τιμή είναι `CornerRadius(2,2,0,0)`.

width

Τύπος: `System.Double`

Καθορίζουμε το πλάτος του παραθύρου σε λογικά pixels. Η default τιμή είναι 450 pixels.

height

Τύπος: `System.Double`

Καθορίζουμε το ύψος του παραθύρου σε λογικά pixels. Η default τιμή είναι 200 pixels .

foreground

Τύπος: `System.Windows.Media`

Καθορίζουμε το χρώμα μελάνης του προτρεπτικού κειμένου με ένα αντικείμενο `Brush`. Η default τιμή είναι `Brushes.Black`.

background

Τύπος: `System.Windows.Media`

Καθορίζουμε το χρώμα φόντου του παραθύρου με ένα αντικείμενο `Brush`. Η default τιμή είναι `Brushes.AntiqueWhite`.

Optional LeftPos

Τύπος: `System.Double`

Καθορίζει τη θέση του αριστερού άκρου του παραθύρου σε λογικά pixels.

Optional topPos

Τύπος: `System.Double`

Καθορίζει τη θέση του άνω άκρου του παραθύρου σε λογικά pixels.

Παρατηρήσεις:

Το παράθυρο που ανοίγει είναι **modal**. Εμφανίζεται στο κέντρο της οθόνης εκτός αν προσδιορίσουμε κάποια από τις παραμέτρους `leftPos`, `topPos`.

Παράδειγμα:

Παρακάτω, στον VB κώδικα, εμφανίζουμε το προτρεπτικό μήνυμα "Παρακαλώ, δώστε το όνομά σας", με τίτλο "Εισαγωγή ονόματος". Στον φάκελο της εφαρμογής έχουμε τοποθετήσει ένα **αρχείο εικόνας** με όνομα "Person.png". Αυτήν την εικόνα εμφανίζουμε ως **εικονίδιο** στο παράθυρο. Καθορίζουμε τις δύο άνω γωνίες του παραθύρου να είναι έντονα στρογγυλές. Το πλάτος είναι 500px και το ύψος 200px. Το `foreground` είναι άσπρο και το `background` είναι μπλε. Τα υπόλοιπα στοιχεία του παραθύρου είναι προκαθορισμένα:

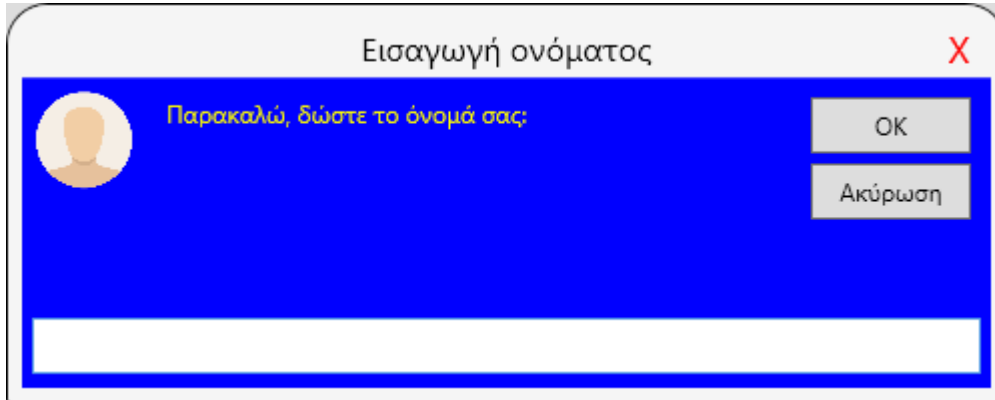
VB :

```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities
```

```
...
```

```
Dim bm As New BitmapImage(New Uri("Person.png", UriKind.Relative))  
Dim img As New Image With {.Source = bm}
```

```
Dim strAnswer As String = InputMessage("Παρακαλώ, δώστε το όνομά σας: ",  
    "Εισαγωγή ονόματος", "", img, New CornerRadius(20, 20, 2, 2),  
    500, 200, Brushes.Yellow, Brushes.Blue)
```



InputMessage(prompt, title, defaultResponse, icon, cornerRadius, width, height, foreground, background, titleStyle, promptStyle, inputTextStyle, Optional leftPos, Optional topPos)

Εμφανίζει ένα παράθυρο εισόδου δεδομένου που έχει ένα προτροπικό μήνυμα κι έναν τίτλο. Στο input textbox εμφανίζεται μία προκαθορισμένη απάντηση. Επίσης, εμφανίζεται και μία εικόνα μεγέθους 48x48px αριστερά του προτροπικού μηνύματος. Προσδιορίζουμε τη στρογγυλότητα στις γωνίες του παραθύρου καθώς και το πλάτος/ύψος. Καθορίζουμε το χρώμα μελάνης του προτροπικού μηνύματος και το χρώμα φόντου του παραθύρου. Καθορίζουμε style για τον τίτλο, το προτροπικό μήνυμα και το input textbox.

Επιστρέφει την απάντηση που έδωσε ο χρήστης ή κενό αν ακύρωσε.

Προαιρετικά, το παράθυρο ανοίγει στη θέση της οθόνης που προσδιορίζεται από τις leftPos και topPos.

Σύνταξη:

VB :

```
Public Shared Function InputMessage(  
    prompt As String,  
    title As String,  
    defaultResponse As String,  
    icon As Image,  
    cornerRadius As CornerRadius,  
    width As Double,  
    height As Double,  
    foreground As Brush,  
    background As Brush,  
    titleStyle As Style,  
    promptStyle As Style,  
    inputTextStyle As Style,  
    Optional leftPos As Double = -1,  
    Optional topPos As Double = -1  
    ) As String
```

Παράμετροι:

prompt

Τύπος: System.String
Το προτροπικό μήνυμα.

title

Τύπος: System.String
Ο τίτλος του παραθύρου. Η default τιμή είναι "Εισαγωγή δεδομένου".

defaultResponse

Τύπος: System.String
Εμφανίζει μία προκαθορισμένη απάντηση στο input textbox. Η default τιμή είναι String.Empty.

icon

Τύπος: System.Windows.Controls.Image

Εμφανίζει μία εικόνα ως icon μεγέθους 48x48px αριστερά του προτρεπτικού μηνύματος. Η default τιμή είναι Nothing.

cornerRadius

Τύπος: System.Windows.CnerRadius

Καθορίζουμε μία στρογγυλότητα στις γωνίες του παραθύρου. Η default τιμή είναι CornerRadius(2,2,0,0).

width

Τύπος: System.Double

Καθορίζουμε το πλάτος του παραθύρου σε λογικά pixels. Η default τιμή είναι 450 pixels.

height

Τύπος: System.Double

Καθορίζουμε το ύψος του παραθύρου σε λογικά pixels. Η default τιμή είναι 200 pixels .

foreground

Τύπος: System.Windows.Media

Καθορίζουμε το χρώμα μελάνης του προτρεπτικού κειμένου με ένα αντικείμενο Brush. Η default τιμή είναι Brushes.Black.

background

Τύπος: System.Windows.Media

Καθορίζουμε το χρώμα φόντου του παραθύρου με ένα αντικείμενο Brush. Η default τιμή είναι Brushes.AntiqueWhite.

titleStyle

Τύπος: System.Windows.Style

Καθορίζουμε το στυλ του τίτλου με ένα αντικείμενο Style. Το στυλ είναι στοχευμένο σε TextBlock .

Η default τιμή περιλαμβάνει: FontSize σε 12, HorizontalAlignment και VerticalAlignment σε Center, Margin σε Thickness(0,4,0,4), Foreground σε Brushes.Black.

promptStyle

Τύπος: System.Windows.Style

Καθορίζουμε το στυλ του προτρεπτικού μηνύματος με ένα αντικείμενο Style. Το στυλ είναι στοχευμένο σε TextBlock.

Η default τιμή περιλαμβάνει: FontSize σε 12, Margin σε Thickness(5), Padding σε Thickness(5).

inputTextStyle

Τύπος: System.Windows.Style

Καθορίζουμε το στυλ του input textbox με ένα αντικείμενο Style. Το στυλ είναι στοχευμένο σε TextBox.

Η default τιμή περιλαμβάνει: FontSize σε 12, Margin σε Thickness(5,7,5,7), Padding σε Thickness(5).

Optional LeftPos

Τύπος: System.Double

Καθορίζει τη θέση του αριστερού άκρου του παραθύρου σε λογικά pixels.

Optional topPos

Τύπος: System.Double

Καθορίζει τη θέση του άνω άκρου του παραθύρου σε λογικά pixels.

Παρατηρήσεις:

Το παράθυρο που ανοίγει είναι **modal**. Εμφανίζεται στο κέντρο της οθόνης εκτός αν προσδιορίσουμε κάποια από τις παραμέτρους leftPos, topPos.

Παράδειγμα:

Παρακάτω, στον VB κώδικα, εμφανίζουμε το προτροπτικό μήνυμα "Παρακαλώ, δώστε το όνομά σας", με τίτλο "Εισαγωγή ονόματος". Στον φάκελο της εφαρμογής έχουμε τοποθετήσει ένα **αρχείο εικόνας** με όνομα "Person.png". Αυτήν την εικόνα εμφανίζουμε ως **εικονίδιο** στο παράθυρο. Καθορίζουμε ομοιόμορφα την στρογγυλότητα των γωνιών του παραθύρου. Το πλάτος είναι 450px και το ύψος 200px. Το foreground είναι απαλό κίτρινο και το background είναι βαθύ μπλε. Επίσης, θέτουμε **styles** για τον τίτλο, το προτροπτικό μήνυμα και το input textbox. Τα υπόλοιπα στοιχεία του παραθύρου είναι προκαθορισμένα.

Τα styles μπορούμε να τα δημιουργήσουμε είτε σε XAML κώδικα είτε σε VB κώδικα. Για να δείξουμε και τους δύο τρόπους θα δημιουργήσουμε το style του τίτλου σε XAML κώδικα, στο τμήμα Windows.Resources, και τα υπόλοιπα δύο (για το προτροπτικό μήνυμα και το input textbox) σε VB κώδικα.

XAML:

```
<Window.Resources >
    ...
    <Style x:Key="inputMessageTitleStyle"
          TargetType="{x:Type TextBlock}">
        <Setter Property="HorizontalAlignment" Value="Center"/>
        <Setter Property="VerticalAlignment" Value="Center"/>
        <Setter Property="FontSize" Value="22"/>
        <Setter Property="Foreground" Value="Green"/>
        <Setter Property="Margin" Value="0,4"/>
    </Style>
</Window.Resources >
```

VB :

```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities
...
Dim tbTitleStyle As Style = FindResource("inputMessageTitleStyle")

Dim promptStyle As New Style With {.TargetType = GetType(TextBlock)}
With promptStyle.Setters
```

```

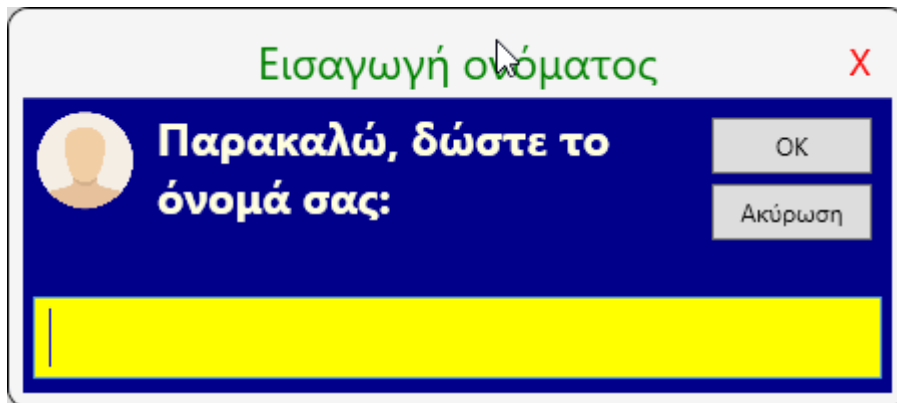
        .Add(New Setter With {.[Property] = TextBlock.FontSizeProperty, _
                                .Value = 22.0})
        .Add(New Setter With {.[Property] = TextBlock.PaddingProperty, _
                                .Value = New Thickness(5)})
        .Add(New Setter With {.[Property] = TextBlock.FontWeightProperty, _
                                .Value = FontWeights.Bold})
    End With

    Dim inputTextStyle As New Style With { .TargetType = GetType(TextBox) }
    With inputTextStyle.Setters
        .Add(New Setter With {.[Property] = TextBox.FontSizeProperty, _
                                .Value = 22.0})
        .Add(New Setter With {.[Property] = TextBox.PaddingProperty, _
                                .Value = New Thickness(5)})
        .Add(New Setter With {.[Property] = TextBox.MarginProperty, _
                                .Value = New Thickness(5, 7, 5, 7)})
        .Add(New Setter With {.[Property] = TextBox.FontWeightProperty, _
                                .Value = FontWeights.Bold})
    End With
    With inputTextStyle.Triggers
        Dim trigger As New Trigger With _
            { .[Property] = TextBox.IsKeyboardFocusedProperty, .Value = True }
        trigger.Setters.Add(New Setter With _
            { .[Property] = TextBox.BackgroundColorProperty, .Value = Brushes.Yellow })
        .Add(trigger)
    End With

    Dim bm As New BitmapImage(New Uri("Person.png", UriKind.Relative))
    Dim img As New Image With { .Source = bm }

    Dim strAnswer As String = InputMessage("Παρακαλώ, δώστε το όνομά σας: ", _
        "Εισαγωγή ονόματος", String.Empty, img, New CornerRadius(10), _
        450, 200, Brushes.LightYellow, Brushes.DarkBlue, _
        tbTitleStyle, promptStyle, inputTextStyle)

```



Στο style του input textbox έχουμε θέσει ένα style trigger: Όταν το control έχει την εστίαση τότε το φόντο χρωματίζεται κίτρινο.

Αν σε μία style παράμετρο δεν επιθυμούμε να θέσουμε ένα custom style αλλά προτιμούμε το default τότε θέτουμε την τιμή *Nothing*.

InputMessage(prompt, title, defaultResponse, icon, cornerRadius, width, height, foreground, background, titleStyle, promptStyle, inputTextStyle, okButtonStyle, cancelButtonStyle, Optional leftPos, Optional topPos)

Εμφανίζει ένα παράθυρο εισόδου δεδομένου που έχει ένα προτρεπτικό μήνυμα κι έναν τίτλο. Στο input textbox εμφανίζεται μία προκαθορισμένη απάντηση. Επίσης, εμφανίζεται και μία εικόνα μεγέθους 48x48px αριστερά του προτρεπτικού μηνύματος. Προσδιορίζουμε τη στρογγυλότητα στις γωνίες του παραθύρου καθώς και το πλάτος/ύψος. Καθορίζουμε το χρώμα μελάνης του προτρεπτικού μηνύματος και το χρώμα φόντου του παραθύρου. Καθορίζουμε style για τον τίτλο, το προτρεπτικό μήνυμα, το input textbox και για τα buttons OK και Cancel.

Επιστρέφει την απάντηση που έδωσε ο χρήστης ή κενό αν ακύρωσε.

Προαιρετικά, το παράθυρο ανοίγει στη θέση της οθόνης που προσδιορίζεται από τις leftPos και topPos.

Σύνταξη:

VB :

```
Public Shared Function InputMessage(  
    prompt As String,  
    title As String,  
    defaultResponse As String,  
    icon As Image,  
    cornerRadius As CornerRadius,  
    width As Double,  
    height As Double,  
    foreground As Brush,  
    background As Brush,  
    titleStyle As Style,  
    promptStyle As Style,  
    inputTextStyle As Style,  
    okButtonStyle As Style,  
    cancelButtonStyle As Style,  
    Optional leftPos As Double = -1,  
    Optional topPos As Double = -1  
    ) As String
```

Παράμετροι:

prompt

Τύπος: System.String
Το προτρεπτικό μήνυμα.

title

Τύπος: System.String
Ο τίτλος του παραθύρου. Η default τιμή είναι "Εισαγωγή δεδομένου".

defaultResponse

Τύπος: System.String
Εμφανίζει μία προκαθορισμένη απάντηση στο input textbox. Η default τιμή είναι String.Empty.

icon

Τύπος: `System.Windows.Controls.Image`
Εμφανίζει μία εικόνα ως `icon` μεγέθους 48x48px αριστερά του προτρεπτικού μηνύματος. Η default τιμή είναι `Nothing`.

cornerRadius

Τύπος: `System.Windows.CornerRadius`
Καθορίζουμε μία στρογγυλότητα στις γωνίες του παραθύρου. Η default τιμή είναι `CornerRadius(2,2,0,0)`.

width

Τύπος: `System.Double`
Καθορίζουμε το πλάτος του παραθύρου σε λογικά pixels. Η default τιμή είναι 450 pixels.

height

Τύπος: `System.Double`
Καθορίζουμε το ύψος του παραθύρου σε λογικά pixels. Η default τιμή είναι 200 pixels .

foreground

Τύπος: `System.Windows.Media`
Καθορίζουμε το χρώμα μελάνης του προτρεπτικού κειμένου με ένα αντικείμενο `Brush`. Η default τιμή είναι `Brushes.Black`.

background

Τύπος: `System.Windows.Media`
Καθορίζουμε το χρώμα φόντου του παραθύρου με ένα αντικείμενο `Brush`. Η default τιμή είναι `Brushes.AntiqueWhite`.

titleStyle

Τύπος: `System.Windows.Style`
Καθορίζουμε το στυλ του τίτλου με ένα αντικείμενο `Style`. Το στυλ είναι στοχευμένο σε `TextBlock` .
Η default τιμή περιλαμβάνει: `FontSize` σε 12, `HorizontalAlignment` και `VerticalAlignment` σε `Center`, `Margin` σε `Thickness(0,4,0,4)`, `Foreground` σε `Brushes.Black`.

promptStyle

Τύπος: `System.Windows.Style`
Καθορίζουμε το στυλ του προτρεπτικού μηνύματος με ένα αντικείμενο `Style`. Το στυλ είναι στοχευμένο σε `TextBlock`.
Η default τιμή περιλαμβάνει: `FontSize` σε 12, `Margin` σε `Thickness(5)`, `Padding` σε `Thickness(5)`.

inputTextStyle

Τύπος: `System.Windows.Style`
Καθορίζουμε το στυλ του `input textbox` με ένα αντικείμενο `Style`. Το στυλ είναι στοχευμένο σε `TextBox`.
Η default τιμή περιλαμβάνει: `FontSize` σε 12, `Margin` σε `Thickness(5,7,5,7)`, `Padding` σε `Thickness(5)`.

okButtonStyle

Τύπος: `System.Windows.Style`

Καθορίζουμε το στυλ του button OK με ένα αντικείμενο Style. Το στυλ είναι στοχευμένο σε Button.

Η default τιμή περιλαμβάνει: Content σε "OK", Padding σε Thickness(5), Width σε 80.

cancelButtonStyle

Τύπος: System.Windows.Style

Καθορίζουμε το στυλ του button Cancel με ένα αντικείμενο Style. Το στυλ είναι στοχευμένο σε Button.

Η default τιμή περιλαμβάνει: Content σε "Ακύρωση", Margin σε Thickness(0,5,0,0), Padding σε Thickness(5), Width σε 80.

Optional LeftPos

Τύπος: System.Double

Καθορίζει τη θέση του αριστερού άκρου του παραθύρου σε λογικά pixels.

Optional topPos

Τύπος: System.Double

Καθορίζει τη θέση του άνω άκρου του παραθύρου σε λογικά pixels.

Παρατηρήσεις:

Το παράθυρο που ανοίγει είναι **modal**. Εμφανίζεται στο κέντρο της οθόνης εκτός αν προσδιορίσουμε κάποια από τις παραμέτρους leftPos, topPos.

Παράδειγμα:

Παρακάτω, στον VB κώδικα, εμφανίζουμε το προτρεπτικό μήνυμα "Παρακαλώ, δώστε το όνομά σας", με τίτλο "Εισαγωγή ονόματος". Στον φάκελο της εφαρμογής έχουμε τοποθετήσει ένα **αρχείο εικόνας** με όνομα "Person.png". Αυτήν την εικόνα εμφανίζουμε ως **εικονίδιο** στο παράθυρο. Καθορίζουμε ομοιόμορφα την στρογγυλότητα των γωνιών του παραθύρου. Το πλάτος είναι 450px και το ύψος 200px. Το foreground είναι απαλό κίτρινο και το background είναι βαθύ μπλε. Επίσης, θέτουμε **styles** για τον τίτλο, το προτρεπτικό μήνυμα, το input textbox και για τα buttons OK και Cancel.

Τα styles μπορούμε να τα δημιουργήσουμε είτε σε XAML κώδικα είτε σε VB κώδικα. Για να δείξουμε και τους δύο τρόπους θα δημιουργήσουμε το style του τίτλου και των buttons σε XAML κώδικα, στο τμήμα Windows.Resources, και τα υπόλοιπα δύο (για το προτρεπτικό μήνυμα και το input textbox) σε VB κώδικα.

XAML:

```
<Window.Resources >
```

```
...
```

```
<Style x:Key="inputMessageTitleStyle"
      TargetType="{x:Type TextBlock}">
```

```
    <Setter Property="HorizontalAlignment" Value="Center"/>
```

```
    <Setter Property="VerticalAlignment" Value="Center"/>
```

```

        <Setter Property="FontSize" Value="22"/>
        <Setter Property="Foreground" Value="Green"/>
        <Setter Property="Margin" Value="0,4"/>

</Style>

<!-- Γενικό στυλ για τα buttons του InputMessage.-->
<Style x:Key="inputMessageButtonStyle" TargetType="{x:Type Button}">
    <Setter Property="HorizontalAlignment" Value="Center"/>
    <Setter Property="VerticalAlignment" Value="Center"/>
    <Setter Property="FontSize" Value="12"/>
    <Setter Property="Padding" Value="5"/>
    <Setter Property="Width" Value="80"/>
    <Setter Property="Height" Value="30"/>
    <Style.Triggers>
        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="FontWeight" Value="Bold"/>
        </Trigger>
    </Style.Triggers>
</Style>

<!-- Ειδικό στυλ για το OK button του InputMessage. -->
<Style x:Key="inputMessageOKButtonStyle"
    TargetType="{x:Type Button}"
    BasedOn="{StaticResource inputMessageButtonStyle}">
    <Setter Property="Foreground" Value="Green"/>
    <Setter Property="Content" Value="OK"/>
</Style>

<!-- Ειδικό στυλ για το Cancel button του InputMessage. -->
<Style x:Key="inputMessageCancelButtonStyle"
    TargetType="{x:Type Button}"
    BasedOn="{StaticResource inputMessageButtonStyle}">
    <Setter Property="Foreground" Value="Red"/>
    <Setter Property="Margin" Value="0,5,0,0"/>
    <Setter Property="Content" Value="Ακύρωση"/>
</Style>

</Window.Resources >

```

VB:

```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities
```

```

...
Dim tbTitleStyle As Style = FindResource("inputMessageTitleStyle")

Dim promptStyle As New Style With {.TargetType = GetType(TextBlock)}
With promptStyle.Setters
    .Add(New Setter With {.[Property] = TextBlock.FontSizeProperty, _
                        .Value = 22.0})
    .Add(New Setter With {.[Property] = TextBlock.PaddingProperty, _
                        .Value = New Thickness(5)})
    .Add(New Setter With {.[Property] = TextBlock.FontWeightProperty, _
                        .Value = FontWeights.Bold})
End With

Dim inputTextStyle As New Style With {.TargetType = GetType(TextBox)}
With inputTextStyle.Setters
    .Add(New Setter With {.[Property] = TextBox.FontSizeProperty, _
                        .Value = 22.0})

```

```

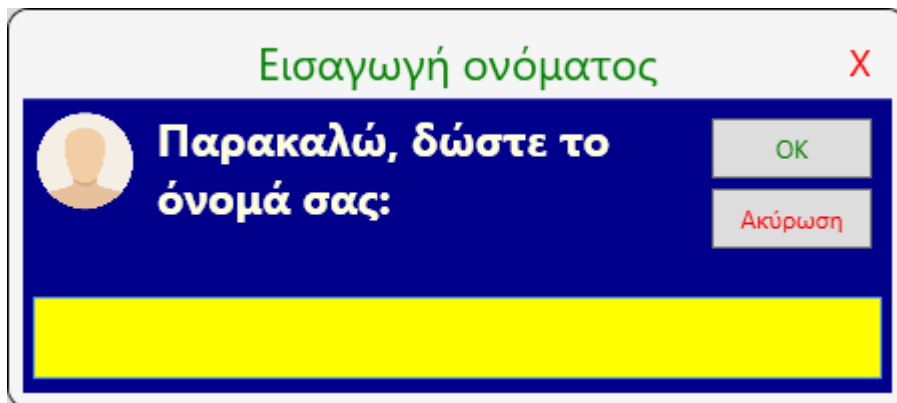
        .Add(New Setter With {.[Property] = TextBox.PaddingProperty, _
            .Value = New Thickness(5)})
        .Add(New Setter With {.[Property] = TextBox.MarginProperty, _
            .Value = New Thickness(5, 7, 5, 7)})
        .Add(New Setter With {.[Property] = TextBox.FontWeightProperty, _
            .Value = FontWeights.Bold})
    End With
    With inputTextStyle.Triggers
        Dim trigger As New Trigger With _
            {.[Property] = TextBox.IsKeyboardFocusedProperty, .Value = True}
        trigger.Setters.Add(New Setter With _
            {.[Property] = TextBox.BackgroundProperty, .Value = Brushes.Yellow})
        .Add(trigger)
    End With

    Dim okButtonStyle As Style = FindResource("inputMessageOKButtonStyle")
    Dim cancelButtonStyle As Style = _
        FindResource("inputMessageCancelButtonStyle")

    Dim bm As New BitmapImage(New Uri("Person.png", UriKind.Relative))
    Dim img As New Image With {.Source = bm}

    Dim strAnswer As String = InputMessage("Παρακαλώ, δώστε το όνομά σας: ", _
        "Εισαγωγή ονόματος", String.Empty, img, New CornerRadius(10), _
        450, 200, Brushes.LightYellow, Brushes.DarkBlue, _
        tbTitleStyle, promptStyle, inputTextStyle, _
        okButtonStyle, cancelButtonStyle)

```



Μεταβάλλοντας το ControlTemplate των buttons (ιδιότητα Template στο αντικείμενο Style) μπορούμε να καθορίσουμε πιο "εξωτικά" buttons.

Αν σε μία style παράμετρο δεν επιθυμούμε να θέσουμε ένα custom style αλλά προτιμούμε το default τότε θέτουμε την τιμή Nothing.

ShowMessage

Όνομα	Περιγραφή
ShowMessage (messageText As String)	<p>Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα.</p> <p>Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.</p>
ShowMessage (messageText As String, caption As String)	<p>Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα κι έναν τίτλο.</p> <p>Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.</p>
ShowMessage (messageText As String, caption As String, button As ShowMessageButton)	<p>Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα κι έναν τίτλο. Καθορίζουμε ποιό/ποιά buttons θα εμφανιστούν.</p> <p>Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.</p>
ShowMessage (messageText As String, caption As String, button As ShowMessageButton, icon As ShowMessageImage)	<p>Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα κι έναν τίτλο. Καθορίζουμε ποιό/ποιά buttons θα εμφανιστούν. Εμφανίζει ένα icon αριστερά του μηνύματος.</p> <p>Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.</p>
ShowMessage (messageText As String, caption As String, button As ShowMessageButton, icon As ShowMessageImage, defaultResult As ShowMessageResult)	<p>Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα κι έναν τίτλο. Καθορίζουμε ποιό/ποιά buttons θα εμφανιστούν. Εμφανίζει ένα icon αριστερά του μηνύματος. Καθορίζουμε ποιό button θα είναι προεπιλεγμένο (default result). Χρήσιμο σε YesNo).</p> <p>Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.</p>

<p>ShowMessage(messageText As String, caption As String, button As ShowMessageButton, icon As ShowMessageImage, defaultResult As ShowMessageResult, cornerRadius As CornerRadius)</p>	<p>Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα κι έναν τίτλο. Καθορίζουμε ποιό/ποιά buttons θα εμφανιστούν. Εμφανίζει ένα icon αριστερά του μηνύματος. Καθορίζουμε ποιό button θα είναι προεπιλεγμένο (default result. Χρήσιμο σε YesNo). Καθορίζουμε και τη στρογγυλότητα των γωνιών του παραθύρου.</p> <p>Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.</p>
<p>ShowMessage(messageText As String, caption As String, button As ShowMessageButton, icon As ShowMessageImage, defaultResult As ShowMessageResult, cornerRadius As CornerRadius, width As Double, height As Double)</p>	<p>Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα κι έναν τίτλο. Καθορίζουμε ποιό/ποιά buttons θα εμφανιστούν. Εμφανίζει ένα icon αριστερά του μηνύματος. Καθορίζουμε ποιό button θα είναι προεπιλεγμένο (default result. Χρήσιμο σε YesNo). Καθορίζουμε τη στρογγυλότητα των γωνιών του παραθύρου καθώς και το πλάτος και ύψος σε pixels.</p> <p>Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.</p>
<p>ShowMessage(messageText As String, caption As String, button As ShowMessageButton, icon As ShowMessageImage, defaultResult As ShowMessageResult, cornerRadius As CornerRadius, width As Double, height As Double, foreground As Brush, background As Brush)</p>	<p>Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα κι έναν τίτλο. Καθορίζουμε ποιό/ποιά buttons θα εμφανιστούν. Εμφανίζει ένα icon αριστερά του μηνύματος. Καθορίζουμε ποιό button θα είναι προεπιλεγμένο (default result. Χρήσιμο σε YesNo). Καθορίζουμε τη στρογγυλότητα των γωνιών του παραθύρου καθώς και το πλάτος και ύψος σε pixels. Επίσης, καθορίζουμε το χρώμα μελάνης του μηνύματος και το background του παραθύρου.</p> <p>Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.</p>

<p>ShowMessage(messageText As String, caption As String, button As ShowMessageButton, icon As ShowMessageImage, defaultResult As ShowMessageResult, cornerRadius As CornerRadius, width As Double, height As Double, foreground As Brush, background As Brush, captionStyle As Style, messageTextStyle As Style)</p>	<p>Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα κι έναν τίτλο. Καθορίζουμε ποιό/ποιά buttons θα εμφανιστούν. Εμφανίζει ένα icon αριστερά του μηνύματος. Καθορίζουμε ποιό button θα είναι προεπιλεγμένο (default result. Χρήσιμο σε YesNo). Καθορίζουμε τη στρογγυλότητα των γωνιών του παραθύρου καθώς και το πλάτος και ύψος σε pixels. Επίσης, καθορίζουμε το χρώμα μελάνης του μηνύματος και το background του παραθύρου. Ορίζουμε styles για τον τίτλο και το μήνυμα.</p> <p>Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.</p>
<p>ShowMessage(messageText As String, caption As String, button As ShowMessageButton, icon As ShowMessageImage, defaultResult As ShowMessageResult, cornerRadius As CornerRadius, width As Double, height As Double, foreground As Brush, background As Brush, captionStyle As Style, messageTextStyle As Style, okOrYesButtonStyle As Style, cancelOrNoButtonStyle As Style)</p>	<p>Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα κι έναν τίτλο. Καθορίζουμε ποιό/ποιά buttons θα εμφανιστούν. Εμφανίζει ένα icon αριστερά του μηνύματος. Καθορίζουμε ποιό button θα είναι προεπιλεγμένο (default result. Χρήσιμο σε YesNo). Καθορίζουμε τη στρογγυλότητα των γωνιών του παραθύρου καθώς και το πλάτος και ύψος σε pixels. Επίσης, καθορίζουμε το χρώμα μελάνης του μηνύματος και το background του παραθύρου. Ορίζουμε styles για τον τίτλο, το μήνυμα και τα buttons.</p> <p>Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.</p>

ShowMessage(messageText)

Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα. Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.

Σύνταξη:

VB :

```
Public Shared Function ShowMessage(  
    messageText As String  
) As ShowMessageResult
```

Παράμετροι:

messageText

Τύπος: System.String

Το μήνυμα που θα εμφανιστεί στο παράθυρο.

Παρατηρήσεις:

Το παράθυρο που ανοίγει είναι **modal**. Εμφανίζεται στο κέντρο της οθόνης.

Παράδειγμα:

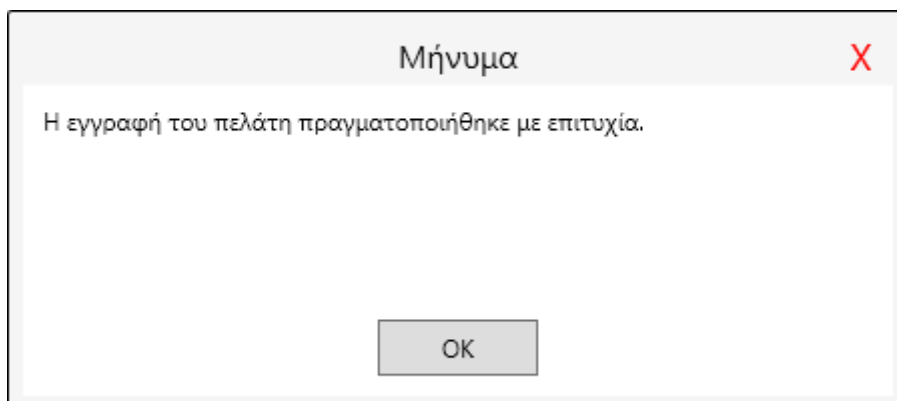
Παρακάτω, στον VB κώδικα, εμφανίζουμε το μήνυμα "Η εγγραφή του πελάτη πραγματοποιήθηκε με επιτυχία". Τα υπόλοιπα στοιχεία του παραθύρου είναι προκαθορισμένα:

VB :

```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities
```

...

```
ShowMessage("Η εγγραφή του πελάτη πραγματοποιήθηκε με επιτυχία")
```



ShowMessage(messageText, caption)

Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα κι έναν τίτλο. Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.

Σύνταξη:

VB :

```
Public Shared Function ShowMessage(  
    messageText As String,  
    caption As String  
    ) As ShowMessageResult
```

Παράμετροι:

messageText

Τύπος: System.String

Το μήνυμα που θα εμφανιστεί στο παράθυρο.

caption

Τύπος: System.String

Ο τίτλος του παραθύρου.

Παρατηρήσεις:

Το παράθυρο που ανοίγει είναι **modal**. Εμφανίζεται στο κέντρο της οθόνης.

Παράδειγμα:

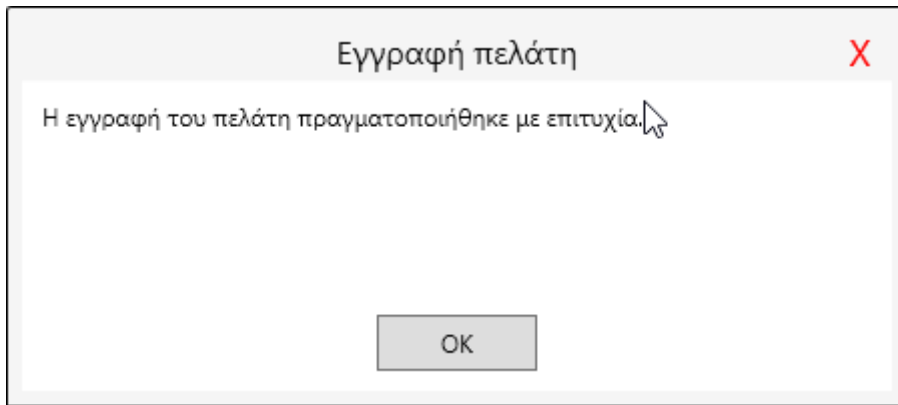
Παρακάτω, στον VB κώδικα, εμφανίζουμε το μήνυμα "Η εγγραφή του πελάτη πραγματοποιήθηκε με επιτυχία" και τίτλο "Εγγραφή πελάτη". Τα υπόλοιπα στοιχεία του παραθύρου είναι προκαθορισμένα:

VB :

```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities
```

```
...
```

```
ShowMessage("Η εγγραφή του πελάτη πραγματοποιήθηκε με επιτυχία",  
            "Εγγραφή πελάτη")
```



ShowMessage(messageText, caption, button)

Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα κι έναν τίτλο. Καθορίζουμε ποιό/ποιά buttons θα εμφανιστούν. Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.

Σύνταξη:

VB :

```
Public Shared Function ShowMessage(  
    messageText As String,  
    caption As String,  
    button As ShowMessageButton  
    ) As ShowMessageResult
```

Παράμετροι:

messageText

Τύπος: System.String
Το μήνυμα που θα εμφανιστεί στο παράθυρο.

caption

Τύπος: System.String
Ο τίτλος του παραθύρου. Η default τιμή είναι "Μήνυμα".

button

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageButton
Ποιό/Ποιά buttons θα εμφανιστούν μέσω της απαρίθμησης ShowMessageButton. Η default τιμή είναι ShowMessageButton.OK

Παρατηρήσεις:

Το παράθυρο που ανοίγει είναι **modal**. Εμφανίζεται στο κέντρο της οθόνης.

Παράδειγμα:

Παρακάτω, στον VB κώδικα, εμφανίζουμε το μήνυμα "Θα προχωρήσετε στη διαγραφή του πελάτη;" και τίτλο "Διαγραφή πελάτη". Καθορίζουμε να εμφανιστούν τα buttons Yes/No. Τα υπόλοιπα στοιχεία του παραθύρου είναι προκαθορισμένα:

VB :

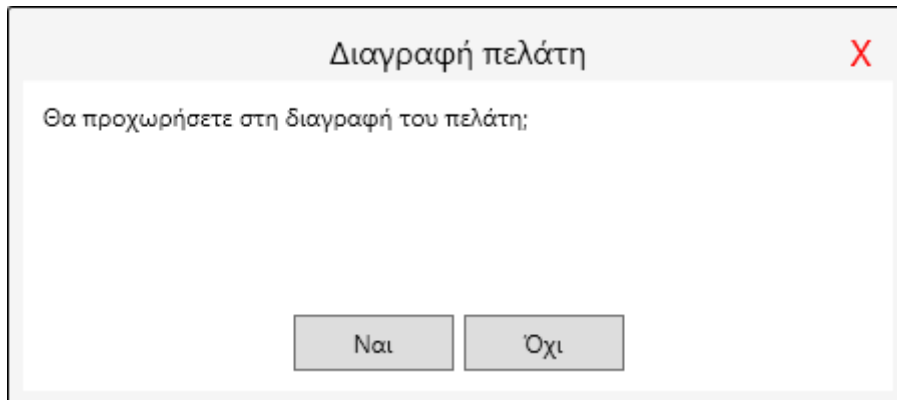
```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities
```

```
...
```

```
If ShowMessage("Θα προχωρήσετε στη διαγραφή του πελάτη;", _  
    "Διαγραφή πελάτη", ShowMessageButton.YesNo) = _  
    ShowMessageResult.Yes Then
```

```
'Delete code.'
```

```
End If
```



ShowMessage(messageText, caption, button, icon)

Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα κι έναν τίτλο. Καθορίζουμε ποιό/ποιά buttons θα εμφανιστούν. Εμφανίζει ένα icon αριστερά του μηνύματος.

Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.

Σύνταξη:

VB :

```
Public Shared Function ShowMessage(  
    messageText As String,  
    caption As String,  
    button As ShowMessageButton,  
    icon As ShowMessageImage  
    ) As ShowMessageResult
```

Παράμετροι:

messageText

Τύπος: System.String

Το μήνυμα που θα εμφανιστεί στο παράθυρο.

caption

Τύπος: System.String

Ο τίτλος του παραθύρου. Η default τιμή είναι "Μήνυμα".

button

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageButton

Ποιό/Ποιά buttons θα εμφανιστούν μέσω της απαρίθμησης

ShowMessageButton. Η default τιμή είναι ShowMessageButton.OK

icon

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageImage

Ποιά εικόνα θα εμφανιστεί αριστερά του μηνύματος μέσω της απαρίθμησης

ShowMessageImage. Η default τιμή είναι ShowMessageImage.None.

Παρατηρήσεις:

Το παράθυρο που ανοίγει είναι **modal**. Εμφανίζεται στο κέντρο της οθόνης.

Παράδειγμα:

Παρακάτω, στον VB κώδικα, εμφανίζουμε το μήνυμα "Θα προχωρήσετε στη διαγραφή του πελάτη;" και τίτλο "Διαγραφή πελάτη". Καθορίζουμε να εμφανιστούν τα buttons Yes/No και το icon που αντιστοιχεί στην τιμή ShowMessageImage.Question. Τα υπόλοιπα στοιχεία του παραθύρου είναι προκαθορισμένα:

VB :

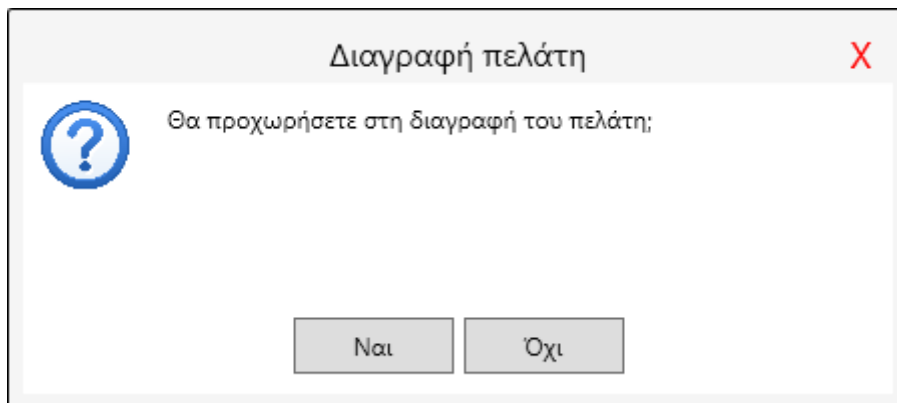
```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities
```

```
...
```

```
If ShowMessage("Θα προχωρήσετε στη διαγραφή του πελάτη;", _  
                "Διαγραφή πελάτη", ShowMessageButton.YesNo, _  
                ShowMessageImage.Question) = ShowMessageResult.Yes Then
```

```
    'Delete code.
```

```
End If
```



ShowMessage(messageText, caption, button, icon, defaultResult)

Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα κι έναν τίτλο. Καθορίζουμε ποιό/ποιά buttons θα εμφανιστούν. Εμφανίζει ένα icon αριστερά του μηνύματος. Καθορίζουμε ποιό button θα είναι προεπιλεγμένο (default result. Χρήσιμο σε YesNo).

Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.

Σύνταξη:

VB :

```
Public Shared Function ShowMessage(  
    messageText As String,  
    caption As String,  
    button As ShowMessageButton,  
    icon As ShowMessageImage,  
    defaultResult As ShowMessageResult  
    ) As ShowMessageResult
```

Παράμετροι:

messageText

Τύπος: System.String

Το μήνυμα που θα εμφανιστεί στο παράθυρο.

caption

Τύπος: System.String

Ο τίτλος του παραθύρου. Η default τιμή είναι "Μήνυμα".

button

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageButton

Ποιό/Ποιά buttons θα εμφανιστούν μέσω της απαρίθμησης

ShowMessageButton. Η default τιμή είναι ShowMessageButton.OK

icon

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageImage

Ποιά εικόνα θα εμφανιστεί αριστερά του μηνύματος μέσω της απαρίθμησης

ShowMessageImage. Η default τιμή είναι ShowMessageImage.None.

defaultResult

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageResult

Ποιά θα είναι η default απόκριση. Η default τιμή είναι ShowMessageResult.OK.

Παρατηρήσεις:

Το παράθυρο που ανοίγει είναι **modal**. Εμφανίζεται στο κέντρο της οθόνης.

Παράδειγμα:

Παρακάτω, στον VB κώδικα, εμφανίζουμε το μήνυμα "Θα προχωρήσετε στη διαγραφή του πελάτη;" και τίτλο "Διαγραφή πελάτη". Καθορίζουμε να εμφανιστούν τα buttons Yes/No και το icon που αντιστοιχεί στην τιμή ShowMessageImage.Question. Το default αποτέλεσμα είναι η τιμή ShowMessageResult.No. Τα υπόλοιπα στοιχεία του παραθύρου είναι προκαθορισμένα:

VB :

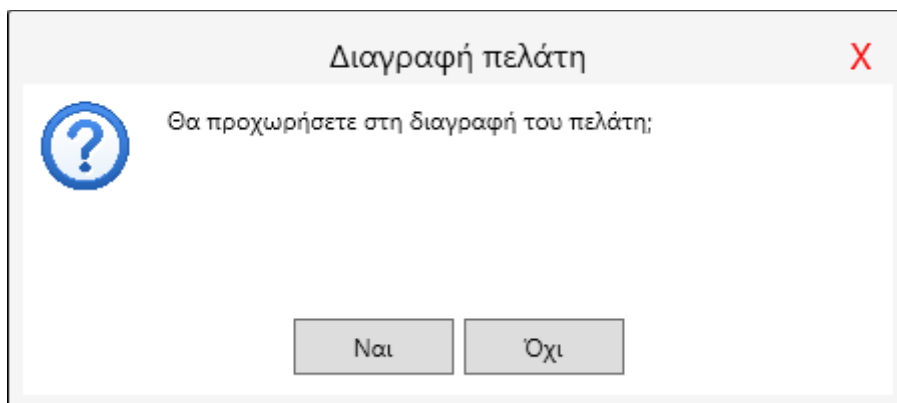
```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities

...

If ShowMessage("Θα προχωρήσετε στη διαγραφή του πελάτη;", _
                "Διαγραφή πελάτη", ShowMessageButton.YesNo, _
                ShowMessageImage.Question, ShowMessageResult.No ) = _
    ShowMessageResult.Yes Then

    'Delete code.

End If
```



Η εστίαση αρχικά είναι στο button No. Έτσι, αν ο χρήστης πατήσει βεβιασμένα το [Enter] το αποτέλεσμα θα είναι ShowMessageResult.No.

ShowMessage(messageText, caption, button, icon, defaultResult, cornerRadius)

Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα κι έναν τίτλο. Καθορίζουμε ποιό/ποιά buttons θα εμφανιστούν. Εμφανίζει ένα icon αριστερά του μηνύματος. Καθορίζουμε ποιό button θα είναι προεπιλεγμένο (default result. Χρήσιμο σε YesNo). Καθορίζουμε και τη στρογγυλότητα των γωνιών του παραθύρου.

Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.

Σύνταξη:

VB :

```
Public Shared Function ShowMessage(  
    messageText As String,  
    caption As String,  
    button As ShowMessageButton,  
    icon As ShowMessageImage,  
    defaultResult As ShowMessageResult,  
    cornerRadius As CornerRadius  
    ) As ShowMessageResult
```

Παράμετροι:

messageText

Τύπος: System.String

Το μήνυμα που θα εμφανιστεί στο παράθυρο.

caption

Τύπος: System.String

Ο τίτλος του παραθύρου. Η default τιμή είναι "Μήνυμα".

button

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageButton

Ποιό/Ποιά buttons θα εμφανιστούν μέσω της απαρίθμησης

ShowMessageButton. Η default τιμή είναι ShowMessageButton.OK

icon

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageImage

Ποιά εικόνα θα εμφανιστεί αριστερά του μηνύματος μέσω της απαρίθμησης

ShowMessageImage. Η default τιμή είναι ShowMessageImage.None.

defaultResult

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageResult

Ποιά θα είναι η default απόκριση. Η default τιμή είναι ShowMessageResult.OK.

cornerRadius

Τύπος: System.Windows.CornerRadius

Καθορίζουμε μία στρογγυλότητα στις γωνίες του παραθύρου. Η default τιμή είναι CornerRadius(2,2,0,0).

Παρατηρήσεις:

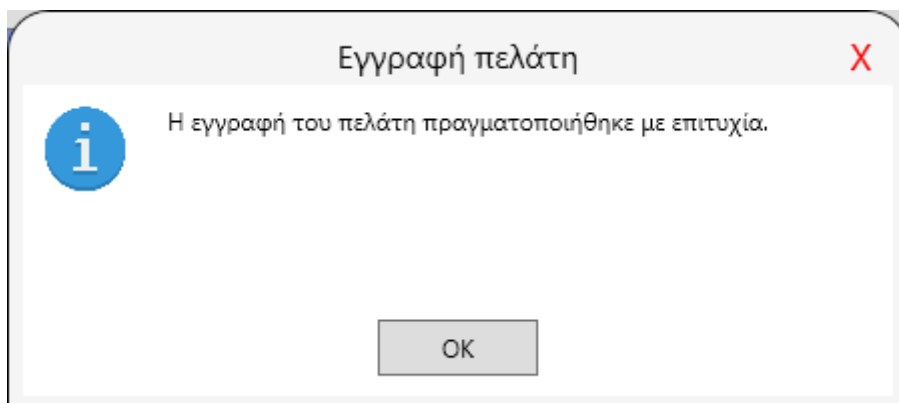
Το παράθυρο που ανοίγει είναι **modal**. Εμφανίζεται στο κέντρο της οθόνης.

Παράδειγμα:

Παρακάτω, στον VB κώδικα, εμφανίζουμε το μήνυμα "Η εγγραφή του πελάτη πραγματοποιήθηκε με επιτυχία" και τίτλο "Εγγραφή πελάτη". Καθορίζουμε το icon "Info" και οι δύο άνω γωνίες του παραθύρου να είναι έντονα στρογγυλές. Τα υπόλοιπα στοιχεία του παραθύρου είναι προκαθορισμένα:

VB :

```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities  
  
...  
  
ShowMessage("Η εγγραφή του πελάτη πραγματοποιήθηκε με επιτυχία.", _  
            "Εγγραφή πελάτη", ShowMessageButton.OK, _  
            ShowMessageImage.Info, ShowMessageResult.OK, _  
            New CornerRadius(20, 20, 2, 2))
```



ShowMessage(messageText, caption, button, icon, defaultResult, cornerRadius, width, height)

Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα κι έναν τίτλο. Καθορίζουμε ποιό/ποιά buttons θα εμφανιστούν. Εμφανίζει ένα icon αριστερά του μηνύματος. Καθορίζουμε ποιό button θα είναι προεπιλεγμένο (default result. Χρήσιμο σε YesNo). Καθορίζουμε τη στρογγυλότητα των γωνιών του παραθύρου καθώς και το πλάτος και ύψος σε pixels.

Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.

Σύνταξη:

VB :

```
Public Shared Function ShowMessage(  
    messageText As String,  
    caption As String,  
    button As ShowMessageButton,  
    icon As ShowMessageImage,  
    defaultResult As ShowMessageResult,  
    cornerRadius As CornerRadius,  
    width As Double,  
    height As Double  
    ) As ShowMessageResult
```

Παράμετροι:

messageText

Τύπος: System.String

Το μήνυμα που θα εμφανιστεί στο παράθυρο.

caption

Τύπος: System.String

Ο τίτλος του παραθύρου. Η default τιμή είναι "Μήνυμα".

button

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageButton

Ποιό/Ποιά buttons θα εμφανιστούν μέσω της απαρίθμησης

ShowMessageButton. Η default τιμή είναι ShowMessageButton.OK

icon

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageImage

Ποιά εικόνα θα εμφανιστεί αριστερά του μηνύματος μέσω της απαρίθμησης

ShowMessageImage. Η default τιμή είναι ShowMessageImage.None.

defaultResult

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageResult

Ποιά θα είναι η default απόκριση. Η default τιμή είναι ShowMessageResult.OK.

cornerRadius

Τύπος: System.Windows.CornerRadius

Καθορίζουμε μία στρογγυλότητα στις γωνίες του παραθύρου. Η default τιμή είναι CornerRadius(2,2,0,0).

width

Τύπος: `System.Double`

Καθορίζουμε το πλάτος του παραθύρου σε λογικά pixels. Η default τιμή είναι 450 pixels.

height

Τύπος: `System.Double`

Καθορίζουμε το ύψος του παραθύρου σε λογικά pixels. Η default τιμή είναι 200 pixels.

Παρατηρήσεις:

Το παράθυρο που ανοίγει είναι **modal**. Εμφανίζεται στο κέντρο της οθόνης.

Παράδειγμα:

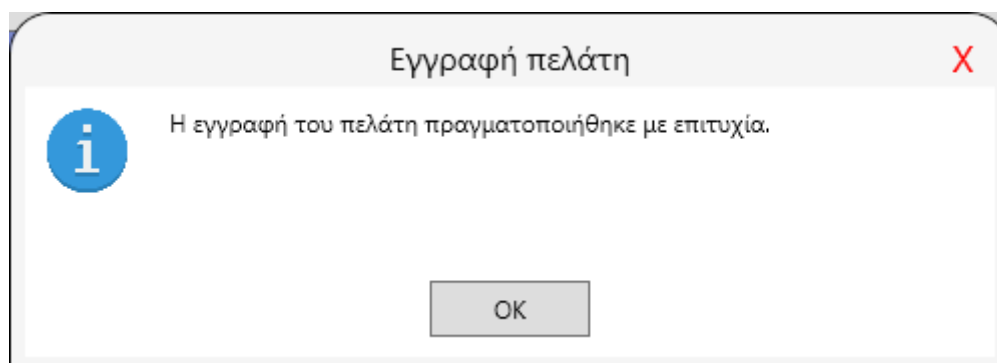
Παρακάτω, στον VB κώδικα, εμφανίζουμε το μήνυμα "Η εγγραφή του πελάτη πραγματοποιήθηκε με επιτυχία" και τίτλο "Εγγραφή πελάτη". Καθορίζουμε το icon "Info" και οι δύο άνω γωνίες του παραθύρου να είναι έντονα στρογγυλές. Το πλάτος είναι 500px και το ύψος 180px. Τα υπόλοιπα στοιχεία του παραθύρου είναι προκαθορισμένα:

VB :

```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities
```

```
...
```

```
ShowMessage("Η εγγραφή του πελάτη πραγματοποιήθηκε με επιτυχία.", _  
            "Εγγραφή πελάτη", ShowMessageButton.OK, _  
            ShowMessageImage.Info, ShowMessageResult.OK, _  
            New CornerRadius(20, 20, 2, 2), 500, 180)
```



ShowMessage(messageText, caption, button, icon, defaultResult, cornerRadius, width, height, foreground, background)

Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα κι έναν τίτλο. Καθορίζουμε ποιό/ποιά buttons θα εμφανιστούν. Εμφανίζει ένα icon αριστερά του μηνύματος. Καθορίζουμε ποιό button θα είναι προεπιλεγμένο (default result. Χρήσιμο σε YesNo). Καθορίζουμε τη στρογγυλότητα των γωνιών του παραθύρου καθώς και το πλάτος και ύψος σε pixels. Επίσης, καθορίζουμε το χρώμα μελάνης του μηνύματος και το background του παραθύρου.

Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.

Σύνταξη:

VB :

```
Public Shared Function ShowMessage(  
    messageText As String,  
    caption As String,  
    button As ShowMessageButton,  
    icon As ShowMessageImage,  
    defaultResult As ShowMessageResult,  
    cornerRadius As CornerRadius,  
    width As Double,  
    height As Double,  
    foreground As Brush,  
    background As Brush  
    ) As ShowMessageResult
```

Παράμετροι:

messageText

Τύπος: System.String

Το μήνυμα που θα εμφανιστεί στο παράθυρο.

caption

Τύπος: System.String

Ο τίτλος του παραθύρου. Η default τιμή είναι "Μήνυμα".

button

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageButton

Ποιό/Ποιά buttons θα εμφανιστούν μέσω της απαρίθμησης

ShowMessageButton. Η default τιμή είναι ShowMessageButton.OK

icon

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageImage

Ποιά εικόνα θα εμφανιστεί αριστερά του μηνύματος μέσω της απαρίθμησης

ShowMessageImage. Η default τιμή είναι ShowMessageImage.None.

defaultResult

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageResult

Ποιά θα είναι η default απόκριση. Η default τιμή είναι ShowMessageResult.OK.

cornerRadius

Τύπος: System.Windows.CornerRadius

Καθορίζουμε μία στρογγυλότητα στις γωνίες του παραθύρου. Η default τιμή είναι `CornerRadius(2,2,0,0)`.

width

Τύπος: `System.Double`

Καθορίζουμε το πλάτος του παραθύρου σε λογικά pixels. Η default τιμή είναι 450 pixels.

height

Τύπος: `System.Double`

Καθορίζουμε το ύψος του παραθύρου σε λογικά pixels. Η default τιμή είναι 200 pixels .

foreground

Τύπος: `System.Windows.Media`

Καθορίζουμε το χρώμα μελάνης του μηνύματος με ένα αντικείμενο Brush. Η default τιμή είναι `Brushes.Black`.

background

Τύπος: `System.Windows.Media`

Καθορίζουμε το χρώμα φόντου του παραθύρου με ένα αντικείμενο Brush. Η default τιμή είναι `Brushes.White`.

Παρατηρήσεις:

Το παράθυρο που ανοίγει είναι **modal**. Εμφανίζεται στο κέντρο της οθόνης.

Παράδειγμα:

Παρακάτω, στον VB κώδικα, εμφανίζουμε το μήνυμα "Η εγγραφή του πελάτη πραγματοποιήθηκε με επιτυχία" και τίτλο "Εγγραφή πελάτη". Καθορίζουμε το icon "Info", τη στρογγυλότητα των γωνιών, πλάτος, ύψος και χρώμα κειμένου και φόντου του παραθύρου. Τα υπόλοιπα στοιχεία του παραθύρου είναι προκαθορισμένα.

Προσδιορίζουμε το φόντο του παραθύρου με ένα `LinearGradientBrush` resource που δηλώνουμε σε XAML και χρησιμοποιούμε στον VB κώδικα.

XAML:

```
<Window.Resources >
    ...
    <!-- Custom background brush for ShowMessage -->
    <LinearGradientBrush x:Key="ShowMessageBackgroundBrush"
        StartPoint="0.5,0" EndPoint="0.5,1" >
        <GradientStop Color="White" Offset="0.1"/>
        <GradientStop Color="Blue" Offset="1"/>
    </LinearGradientBrush>
```

```
</Window.Resources >
```

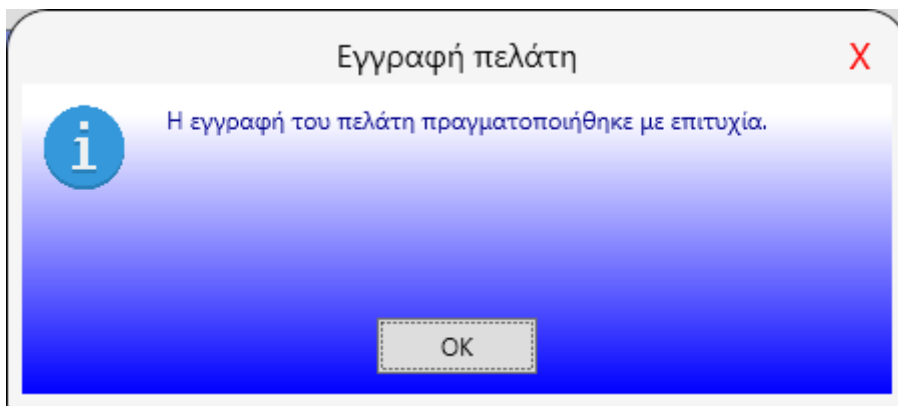
VB :

```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities
```

```
...
```

```
Dim backBrush As Brush = FindResource("ShowMessageBackgroundBrush")
```

```
ShowMessage("Η εγγραφή του πελάτη πραγματοποιήθηκε με επιτυχία.", _  
            "Εγγραφή πελάτη", ShowMessageButton.OK, _  
            ShowMessageImage.Info, ShowMessageResult.OK, _  
            New CornerRadius(20, 20, 2, 2), 450, 200, _  
            Brushes.DarkBlue, backBrush)
```



ShowMessage(messageText, caption, button, icon, defaultResult, cornerRadius, width, height, foreground, background, captionStyle, messageTextStyle)

Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα κι έναν τίτλο. Καθορίζουμε ποιό/ποιά buttons θα εμφανιστούν. Εμφανίζει ένα icon αριστερά του μηνύματος. Καθορίζουμε ποιό button θα είναι προεπιλεγμένο (default result. Χρήσιμο σε YesNo). Καθορίζουμε τη στρογγυλότητα των γωνιών του παραθύρου καθώς και το πλάτος και ύψος σε pixels. Επίσης, καθορίζουμε το χρώμα μελάνης του μηνύματος και το background του παραθύρου. Ορίζουμε styles για τον τίτλο και το μήνυμα. Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.

Σύνταξη:

VB :

```
Public Shared Function ShowMessage(  
    messageText As String,  
    caption As String,  
    button As ShowMessageButton,  
    icon As ShowMessageImage,  
    defaultResult As ShowMessageResult,  
    cornerRadius As CornerRadius,  
    width As Double,  
    height As Double,  
    foreground As Brush,  
    background As Brush,  
    captionStyle As Style,  
    messageTextStyle As Style  
) As ShowMessageResult
```

Παράμετροι:

messageText

Τύπος: System.String

Το μήνυμα που θα εμφανιστεί στο παράθυρο.

caption

Τύπος: System.String

Ο τίτλος του παραθύρου. Η default τιμή είναι "Μήνυμα".

button

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageButton

Ποιό/Ποιά buttons θα εμφανιστούν μέσω της απαρίθμησης

ShowMessageButton. Η default τιμή είναι ShowMessageButton.OK

icon

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageImage

Ποιά εικόνα θα εμφανιστεί αριστερά του μηνύματος μέσω της απαρίθμησης

ShowMessageImage. Η default τιμή είναι ShowMessageImage.None.

defaultResult

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageResult

Ποιά θα είναι η default απόκριση. Η default τιμή είναι ShowMessageResult.OK.

cornerRadius

Τύπος: System.Windows.CnerRadius

Καθορίζουμε μία στρογγυλότητα στις γωνίες του παραθύρου. Η default τιμή είναι CornerRadius(2,2,0,0).

width

Τύπος: System.Double

Καθορίζουμε το πλάτος του παραθύρου σε λογικά pixels. Η default τιμή είναι 450 pixels.

height

Τύπος: System.Double

Καθορίζουμε το ύψος του παραθύρου σε λογικά pixels. Η default τιμή είναι 200 pixels .

foreground

Τύπος: System.Windows.Media

Καθορίζουμε το χρώμα μελάνης του μηνύματος με ένα αντικείμενο Brush. Η default τιμή είναι Brushes.Black.

background

Τύπος: System.Windows.Media

Καθορίζουμε το χρώμα φόντου του παραθύρου με ένα αντικείμενο Brush. Η default τιμή είναι Brushes.White.

captionStyle

Τύπος: System.Windows.Style

Καθορίζουμε το στυλ του τίτλου με ένα αντικείμενο Style. Το στυλ είναι στοχευμένο σε TextBlock .

Η default τιμή περιλαμβάνει: FontSize σε 12, HorizontalAlignment και VerticalAlignment σε Center, Margin σε Thickness(0,4,0,4), Foreground σε Brushes.Black.

messageTextStyle

Τύπος: System.Windows.Style

Καθορίζουμε το στυλ του μηνύματος με ένα αντικείμενο Style. Το στυλ είναι στοχευμένο σε TextBlock.

Η default τιμή περιλαμβάνει: FontSize σε 12, Margin σε Thickness(5), Padding σε Thickness(5).

Παρατηρήσεις:

Το παράθυρο που ανοίγει είναι **modal**. Εμφανίζεται στο κέντρο της οθόνης.

Παράδειγμα:

Παρακάτω, στον VB κώδικα, εμφανίζουμε το μήνυμα "Η εγγραφή του πελάτη πραγματοποιήθηκε με επιτυχία" και τίτλο "Εγγραφή πελάτη". Καθορίζουμε το icon "Info", τη στρογγυλότητα των γωνιών, πλάτος, ύψος και χρώμα κειμένου και φόντου του

παραθύρου. Επιπλέον, θέτουμε custom style για τον τίτλο (caption) και το μήνυμα. Τα υπόλοιπα στοιχεία του παραθύρου είναι προκαθορισμένα.

Προσδιορίζουμε το φόντο του παραθύρου με ένα LinearGradientBrush resource, όπως κάνουμε επίσης και με τα styles τα οποία δηλώνουμε σε XAML και χρησιμοποιούμε στον VB κώδικα.

XAML:

```
<Window.Resources >

    ...

    <!-- Custom background brush for ShowMessage -->
    <LinearGradientBrush x:Key="ShowMessageBackgroundBrush"
        StartPoint="0.5,0" EndPoint="0.5,1" >

        <GradientStop Color="White" Offset="0.1"/>
        <GradientStop Color="Blue" Offset="1"/>

    </LinearGradientBrush>

    <!-- Custom title style for ShowMessage -->
    <Style x:Key="ShowMessageCaptionStyle" TargetType="{x:Type
        TextBlock}" >

        <Setter Property="FontSize" Value="22"/>
        <Setter Property="FontWeight" Value="Bold"/>
        <Setter Property="Foreground" Value="Green"/>
        <Setter Property="HorizontalAlignment" Value="Center"/>
        <Setter Property="Margin" Value="5"/>

    </Style>

    <!-- Custom messageText style for ShowMessage -->
    <Style x:Key="ShowMessageMessageTextStyle" TargetType="{x:Type
        TextBlock}" >

        <Setter Property="FontSize" Value="18"/>
        <Setter Property="Margin" Value="5,10,5,5"/>

    </Style>

</Window.Resources >
```

VB:

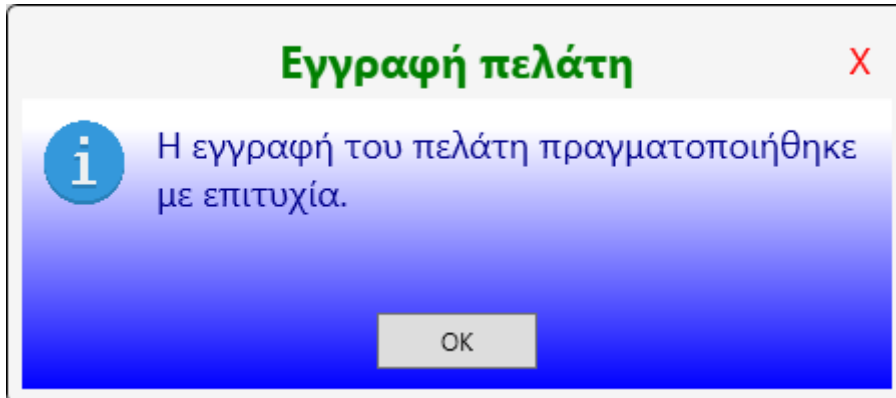
```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities

...

Dim backBrush As Brush = FindResource("ShowMessageBackgroundBrush")
Dim captionStyle As Style = FindResource("ShowMessageCaptionStyle")
Dim messageTextStyle As Style = FindResource("ShowMessageMessageTextStyle")

ShowMessage("Η εγγραφή του πελάτη πραγματοποιήθηκε με επιτυχία.", _
            "Εγγραφή πελάτη", ShowMessageButton.OK, _
            ShowMessageImage.Info, ShowMessageResult.OK, _
            New CornerRadius(5), 450, 200, _
```

```
Brushes.DarkBlue, backButton,  
captionStyle, messageTextStyle)
```



Αν σε μία style παράμετρο δεν επιθυμούμε να θέσουμε ένα custom style αλλά προτιμούμε το default τότε θέτουμε την τιμή `Nothing`.

ShowMessage(messageText, caption, button, icon, defaultResult, cornerRadius, width, height, foreground, background, captionStyle, messageTextStyle, okOrYesButtonStyle, cancelOrNoButtonStyle)

Εμφανίζει ένα παράθυρο που έχει ένα μήνυμα κι έναν τίτλο. Καθορίζουμε ποιό/ποιά buttons θα εμφανιστούν. Εμφανίζει ένα icon αριστερά του μηνύματος. Καθορίζουμε ποιό button θα είναι προεπιλεγμένο (default result. Χρήσιμο σε YesNo). Καθορίζουμε τη στρογγυλότητα των γωνιών του παραθύρου καθώς και το πλάτος και ύψος σε pixels. Επίσης, καθορίζουμε το χρώμα μελάνης του μηνύματος και το background του παραθύρου. Ορίζουμε styles για τον τίτλο, το μήνυμα και τα buttons.

Επιστρέφει μία τιμή από την απαρίθμηση ShowMessageResult ανάλογα με το button που πάτησε ο χρήστης.

Σύνταξη:

VB :

```
Public Shared Function ShowMessage(  
    messageText As String,  
    caption As String,  
    button As ShowMessageButton,  
    icon As ShowMessageImage,  
    defaultResult As ShowMessageResult,  
    cornerRadius As CornerRadius,  
    width As Double,  
    height As Double,  
    foreground As Brush,  
    background As Brush,  
    captionStyle As Style,  
    messageTextStyle As Style,  
    okOrYesButtonStyle As Style,  
    cancelOrNoButtonStyle As Style  
    ) As ShowMessageResult
```

Παράμετροι:

messageText

Τύπος: System.String

Το μήνυμα που θα εμφανιστεί στο παράθυρο.

caption

Τύπος: System.String

Ο τίτλος του παραθύρου. Η default τιμή είναι "Μήνυμα".

button

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageButton

Ποιό/Ποιά buttons θα εμφανιστούν μέσω της απαρίθμησης

ShowMessageButton. Η default τιμή είναι ShowMessageButton.OK

icon

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageImage

Ποιά εικόνα θα εμφανιστεί αριστερά του μηνύματος μέσω της απαρίθμησης ShowMessageImage. Η default τιμή είναι ShowMessageImage.None.

defaultResult

Τύπος: Zeus.WPF.Classes.Utilities.WindowUtilities.ShowMessageResult
Ποιά θα είναι η default απόκριση. Η default τιμή είναι ShowMessageResult.OK.

cornerRadius

Τύπος: System.Windows.CornerRadius
Καθορίζουμε μία στρογγυλότητα στις γωνίες του παραθύρου. Η default τιμή είναι CornerRadius(2,2,0,0).

width

Τύπος: System.Double
Καθορίζουμε το πλάτος του παραθύρου σε λογικά pixels. Η default τιμή είναι 450 pixels.

height

Τύπος: System.Double
Καθορίζουμε το ύψος του παραθύρου σε λογικά pixels. Η default τιμή είναι 200 pixels.

foreground

Τύπος: System.Windows.Media
Καθορίζουμε το χρώμα μελάνης του μηνύματος με ένα αντικείμενο Brush. Η default τιμή είναι Brushes.Black.

background

Τύπος: System.Windows.Media
Καθορίζουμε το χρώμα φόντου του παραθύρου με ένα αντικείμενο Brush. Η default τιμή είναι Brushes.White.

captionStyle

Τύπος: System.Windows.Style
Καθορίζουμε το στυλ του τίτλου με ένα αντικείμενο Style. Το στυλ είναι στοχευμένο σε TextBlock.
Η default τιμή περιλαμβάνει: FontSize σε 12, HorizontalAlignment και VerticalAlignment σε Center, Margin σε Thickness(0,4,0,4), Foreground σε Brushes.Black.

messageTextStyle

Τύπος: System.Windows.Style
Καθορίζουμε το στυλ του μηνύματος με ένα αντικείμενο Style. Το στυλ είναι στοχευμένο σε TextBlock.
Η default τιμή περιλαμβάνει: FontSize σε 12, Margin σε Thickness(5), Padding σε Thickness(5).

okOrYesButtonStyle

Τύπος: System.Windows.Style
Καθορίζουμε το στυλ του OK ή Yes button με ένα αντικείμενο Style. Το στυλ είναι στοχευμένο σε Button.
Η default τιμή περιλαμβάνει: Content σε "OK" ή "Ναι" ανάλογα με την τιμή της ShowMessageButton, Padding σε Thickness(5) και Width σε 80.

cancelOrNoButtonStyle

Τύπος: `System.Windows.Style`

Καθορίζουμε το στυλ του OK ή Yes button με ένα αντικείμενο `Style`. Το στυλ είναι στοχευμένο σε `Button`.

Η default τιμή περιλαμβάνει: `Content` σε "Ακύρωση" ή "Όχι" ανάλογα με την τιμή της `ShowMessageButton`, `Margin` σε `Thickness(5,0,0,0)`, `Padding` σε `Thickness(5)` και `Width` σε 80.

Παρατηρήσεις:

Το παράθυρο που ανοίγει είναι **modal**. Εμφανίζεται στο κέντρο της οθόνης.

Παράδειγμα:

Παρακάτω, στον VB κώδικα, εμφανίζουμε το μήνυμα "Θα προχωρήσετε στη διαγραφή του πελάτη;" και τίτλο "Διαγραφή πελάτη". Καθορίζουμε να εμφανιστούν τα buttons Yes/No και το icon "Info". Ορίζουμε, επίσης, στρογγυλότητα γωνιών, πλάτος, ύψος και styles για τα buttons Yes/No. Επειδή επιθυμούμε να μην αλλάξουμε τα προκαθορισμένα styles για τίτλο και μήνυμα θέτουμε στις αντίστοιχες παραμέτρους την τιμή `Nothing`.

Δηλώνουμε τα button styles σε XAML και τα χρησιμοποιούμε στον VB κώδικα.

XAML:

```
<Window.Resources >
    ...
    <!-- Custom button base style for ShowMessage -->
    <Style x:Key="ShowMessageMessageButtonBaseStyle"
        TargetType="{x:Type Button}" >
        <Setter Property="Content" Value="Ναι" />
        <Setter Property="FontSize" Value="18" />
        <Setter Property="Padding" Value="5" />
        <Setter Property="Width" Value="100" />
        <Setter Property="Height" Value="40" />
    </Style>

    <!-- Custom Yes button style for ShowMessage -->
    <Style x:Key="ShowMessageMessageYesButtonStyle"
        TargetType="{x:Type Button}"
        BasedOn="{StaticResource ShowMessageMessageButtonBaseStyle}">
        <Setter Property="Content" Value="Ναι" />
        <Setter Property="Foreground" Value="Green" />
        <Setter Property="Background" Value="LightGreen" />
    </Style>

    <!-- Custom No button style for ShowMessage -->
    <Style x:Key="ShowMessageMessageNoButtonStyle"
        TargetType="{x:Type Button}"
```

```

BasedOn="{StaticResource ShowMessageMessageButtonBaseStyle}">

<Setter Property="Content" Value="Όχι"/>
<Setter Property="Foreground" Value="Red"/>
<Setter Property="Background" Value="LightPink"/>
<Setter Property="Margin" Value="5,0,0,0"/>

</Style>

</Window.Resources >

```

VB :

```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities
```

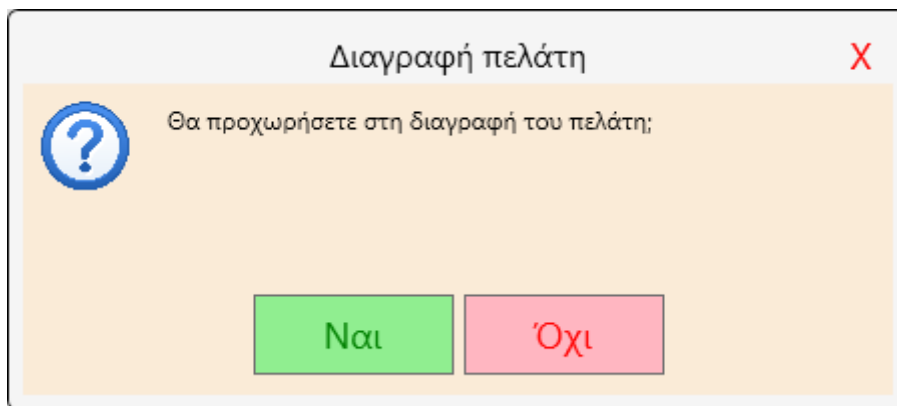
```
...
```

```
Dim yesButtonStyle As Style =
FindResource("ShowMessageMessageYesButtonStyle")
Dim noButtonStyle As Style =
FindResource("ShowMessageMessageNoButtonStyle")
```

```
If ShowMessage("Θα προχωρήσετε στη διαγραφή του πελάτη;", _
                "Διαγραφή πελάτη", ShowMessageButton.YesNo, _
                ShowMessageImage.Question, ShowMessageResult.No, _
                New CornerRadius(5), 450, 200, _
                Brushes.Black, Brushes.AntiqueWhite, _
                Nothing, Nothing, _
                yesButtonStyle, noButtonStyle) = _
    ShowMessageResult.Yes Then

    'Delete code.

End If
```



Μεταβάλλοντας το `ControlTemplate` των buttons (ιδιότητα `Template` στο αντικείμενο `Style`) μπορούμε να καθορίσουμε πιο "εξωτικά" buttons.

Αν σε μία `style` παράμετρο δεν επιθυμούμε να θέσουμε ένα `custom style` αλλά προτιμούμε το `default` τότε θέτουμε την τιμή `Nothing`.

WaitWindow – Μέθοδοι

Όνομα	Περιγραφή
CloseWaitWindow (windowWait As Window)	Κλείνει το συγκεκριμένο wait window.
ShowWaitWindow (windowWait As Window, message As String, Optional gradientBackColor1 As Nullable(Of Color) = Nothing, Optional gradientBackColor2 As Nullable(Of Color) = Nothing)	Εμφανίζει ένα wait window, με συγκεκριμένο message και προαιρετικά background δύο χρωμάτων gradientBackColor1 και gradientBackColor2.

CloseWaitWindow(windowWait)

Κλείνει το συγκεκριμένο wait window.

Σύνταξη:

VB :

```
Public Shared Sub CloseWaitWindow(  
                                windowWait As Window  
                                )
```

Παράμετροι:

windowWait
Τύπος: System.Windows.Window
Το αντικείμενο Window, που πρέπει να κλείσει.

Παρατηρήσεις:

Το αντικείμενο Window είναι **modeless** κι έχει ανοίξει ήδη μέσω της μεθόδου ShowWaitWindow.

Παράδειγμα:

VB :

```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities  
  
...  
  
winWait = New Window  
ShowWaitWindow(winWait, "Παρακαλώ, περιμένετε...")  
'... έναρξη της χρονοθώρας διαδικασίας...  
  
...  
  
'... μετά το πέρας της χρονοθώρας διαδικασίας...  
CloseWaitWindow(winWait)
```

ShowWaitWindow(windowWait, message, Optional gradientBackColor1, Optional gradientBackColor2)

Εμφανίζει ένα **wait window**, με συγκεκριμένο **message** και προαιρετικά background δύο χρωμάτων **gradientBackColor1** και **gradientBackColor2**.

Σύνταξη:

VB :

```
Public Shared Sub ShowWaitWindow(  
    windowWait As Window,  
    message As String,  
    Optional gradientBackColor1 As _  
        Nullable(Of Color) = Nothing,  
    Optional gradientBackColor2 As _  
        Nullable(Of Color) = Nothing  
)
```

Παράμετροι:

windowWait

Τύπος: System.Windows.Window

Το αντικείμενο Window, που θα ανοίξει (modeless).

message

Τύπος: System.String

Το περιεχόμενο του παραθύρου.

Optional gradientBackColor1

Τύπος: System.Nullable(Of Color)

Το πρώτο χρώμα στο linear background. Αν δεν οριστεί, η default τιμή είναι Colors.Orange.

Optional gradientBackColor2

Τύπος: System.Nullable(Of Color)

Το δεύτερο χρώμα στο linear background. Αν δεν οριστεί, η default τιμή είναι Colors.White.

Παρατηρήσεις:

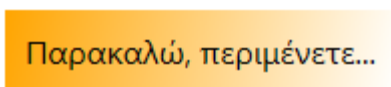
Το αντικείμενο Window είναι **modeless**.

Παράδειγμα 1:

Παρακάτω, στον VB κώδικα, πριν ξεκινήσουμε μία χρονοβόρα διαδικασία, καλούμε τη μέθοδο ShowWaitWindow. Αφήνουμε το default linear background:

VB :

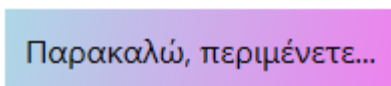
```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities  
  
...  
  
winWait = New Window  
ShowWaitWindow(winWait, "Παρακαλώ, περιμένετε...")  
'... έναρξη της χρονοθώρας διαδικασίας...  
  
...  
  
'... μετά το πέρας της χρονοθώρας διαδικασίας...  
CloseWaitWindow(winWait)
```

**Παράδειγμα 2:**

Το ίδιο με το παραπάνω μόνο που ορίζουμε ως linear background δύο χρώματα : LightBlue και Violet.

VB :

```
Imports Zeus.WPF.Classes.Utilities.WindowUtilities  
  
...  
  
winWait = New Window  
ShowWaitWindow(winWait, "Παρακαλώ, περιμένετε...", Colors.LightBlue,  
Colors.Violet)  
'... έναρξη της χρονοθώρας διαδικασίας...  
  
...  
  
'... μετά το πέρας της χρονοθώρας διαδικασίας...  
CloseWaitWindow(winWait)
```



Τέλος Manual