



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Ανώτατο Εκπαιδευτικό Ίδρυμα Πειραιά
Τεχνολογικού Τομέα



Εισαγωγή στην Πληροφορική & τον Προγραμματισμό

Ενότητα 11^η: Αντικείμενα Γραφικής Διασύνδεσης
(Μέρος 1^ο)

Ι. Ψαρομήλιγκος – Χ. Κυτάγιας
Τμήμα Διοίκησης Επιχειρήσεων



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Ανώτατο Εκπαιδευτικό Ίδρυμα Πειραιά Τεχνολογικού Τομέα**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Σκοποί ενότητας

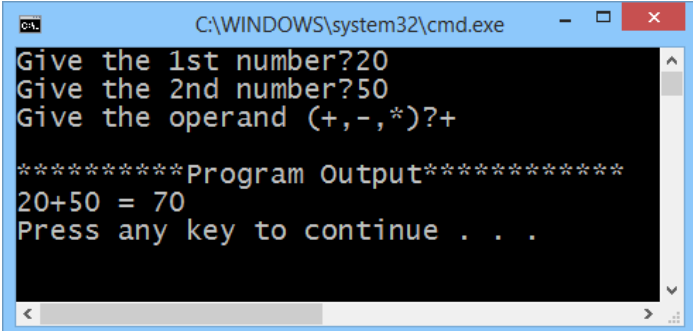
Στην συγκεκριμένη ενότητα γίνεται μια εισαγωγή στις εφαρμογές με γραφική διασύνδεση με το χρήστη και πιο συγκεκριμένα σε εφαρμογές Windows Forms Applications και παρουσιάζονται τα αντικείμενα γραφικής διασύνδεσης Form, Label, TextBox, Button και PictureBox της Γλώσσας Προγραμματισμού Visual Basic.NET.

Περιεχόμενα

- Γραφική Διασύνδεση Εφαρμογής με το Χρήστη
- Εφαρμογές Windows Forms Applications
- Η Φόρμα ως Κλάση και ως Αντικείμενο
- Συμβάντα & Χειριστές Συμβάντων
- Τα αντικείμενα Label, TextBox, Button και PictureBox

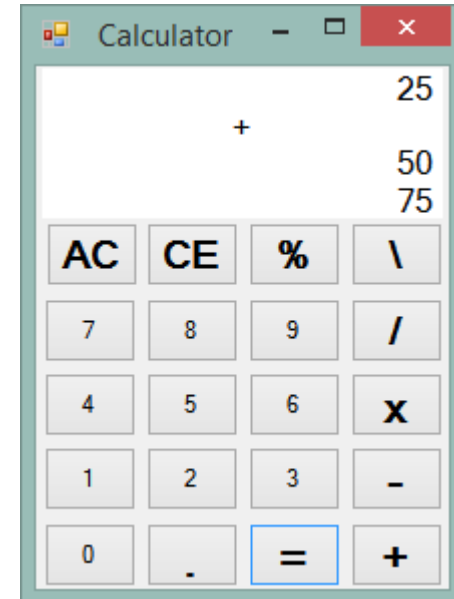
Γραφική Διασύνδεση Εφαρμογής με το Χρήστη

- Οι σύγχρονες εφαρμογές χρησιμοποιούν γραφική διασύνδεση με το χρήστη (**graphical user interface**), δηλαδή, ο χρήστης αλληλεπιδρά με την εφαρμογή μέσα από εικονίδια και όχι κείμενα όπως μέχρι τώρα βλέπαμε στις εφαρμογές Console Applications.
- Η γραφική διασύνδεση με το χρήστη ανέβασε σε υψηλό βαθμό την ευχρηστία των εφαρμογών λογισμικού όχι μόνο μέσα από τα ελκυστικά γραφικά αλλά και από το γεγονός ότι ο χρήστης εξοικειώνεται πολύ γρήγορα με τη λειτουργικότητα της εφαρμογής



```
C:\WINDOWS\system32\cmd.exe
Give the 1st number?20
Give the 2nd number?50
Give the operand (+,-,*)?+

*****Program Output*****
20+50 = 70
Press any key to continue . . .
```



Άλλες μορφές διασύνδεσης με το χρήστη Kinect for windows



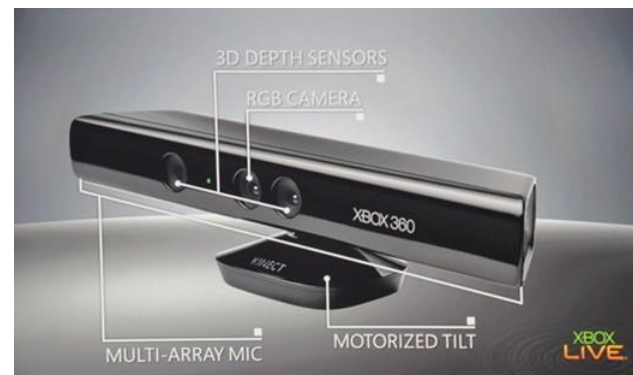
<https://youtu.be/Mr71jrkzWq8>



http://i.telegraph.co.uk/multimedia/archive/02066/minority_2066519b.jpg



<http://www.pcmag.com/article2/0,2817,2402241,00.asp>



<http://www.vankaizer.com/2012/01/>

Άλλες μορφές διασύνδεσης με το χρήστη Leap Motion



<https://youtu.be/Mr71jrkzWq8>



<https://youtu.be/aWIX2yxcMHo>



<http://lifelabs.ucp.org/tiny-device-huge-potential-how-leap-motion-will-change-computing/>

Άλλες μορφές διασύνδεσης με το χρήστη Google Glasses



<http://www.glassappsource.com/glass-developers>



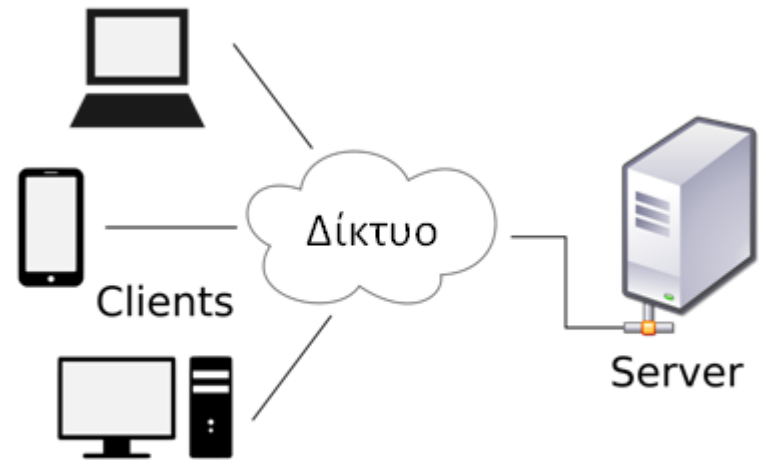
https://youtu.be/4_X6EyqXa2s



<http://www.ryot.org/google-to-sell-internet-glasses-to-contest-winners/81713>

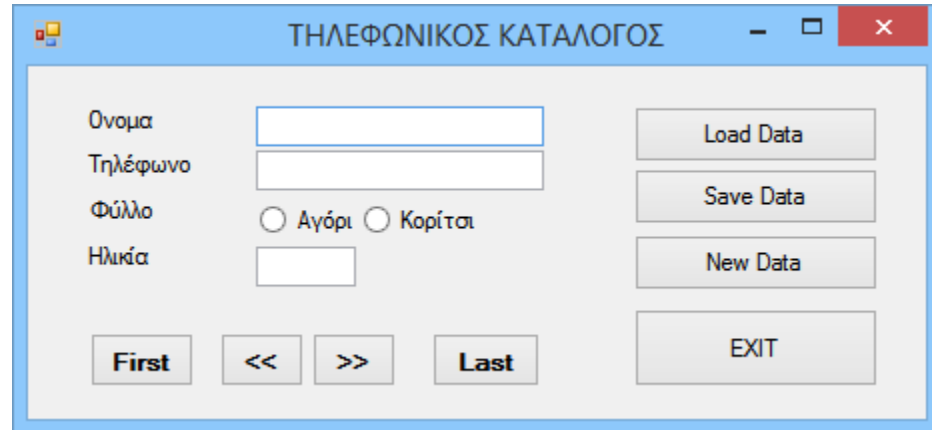
Εφαρμογές Windows Forms Applications (1)

- Με τις εφαρμογές **Windows Forms Applications** θα αναπτύξουμε προγράμματα που χρησιμοποιούν γραφική διασύνδεση με το χρήστη.
- Οι εφαρμογές αυτές αποτελούν συνήθως το «client» τμήμα μιας «client-server» εφαρμογής δηλαδή το τμήμα της εφαρμογής που επικοινωνεί με τον τελικό χρήστη.



Εφαρμογές Windows Forms Applications (2)

Οι εφαρμογές **Windows Forms Applications** χρησιμοποιούν βιβλιοθήκες του περιβάλλοντος .NET οι οποίες παρέχουν ένα σύνολο από γραφικά αντικείμενα για την απλοποίηση των διαδικασιών εμφάνισης πληροφοριών προς το χρήστη ή υποβολής στοιχείων από τον χρήστη στην εφαρμογή.



ΤΗΛΕΦΩΝΙΚΟΣ ΚΑΤΑΛΟΓΟΣ

Όνομα

Τηλέφωνο

Φύλλο Αγόρι Κορίτσι

Ηλικία

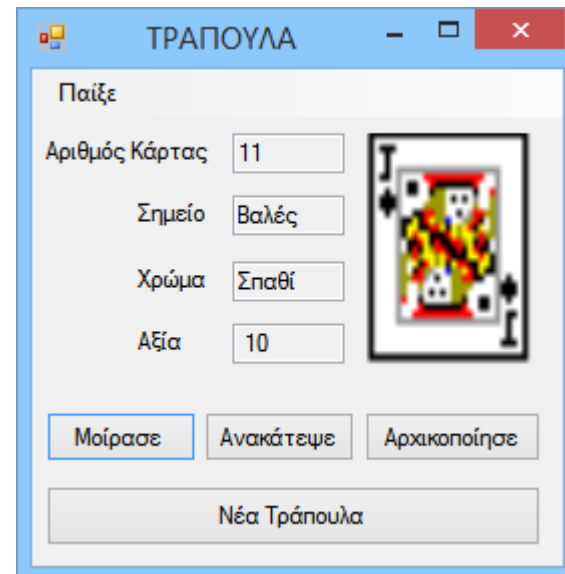
Load Data

Save Data

New Data

EXIT

First << >> Last



ΤΡΑΠΟΥΛΑ

Παίξε

Αριθμός Κάρτας

Σημείο

Χρώμα

Αξία

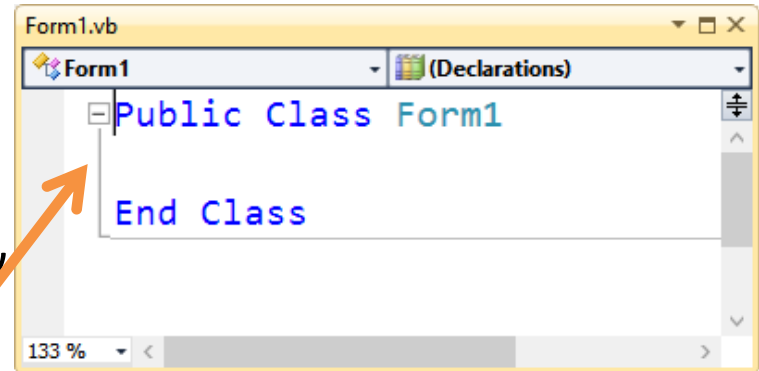
Μοίρασε Ανακάτεψε Αρχικοποίηση

Νέα Τράπουλα

Η Φόρμα ως Κλάση & ως Αντικείμενο

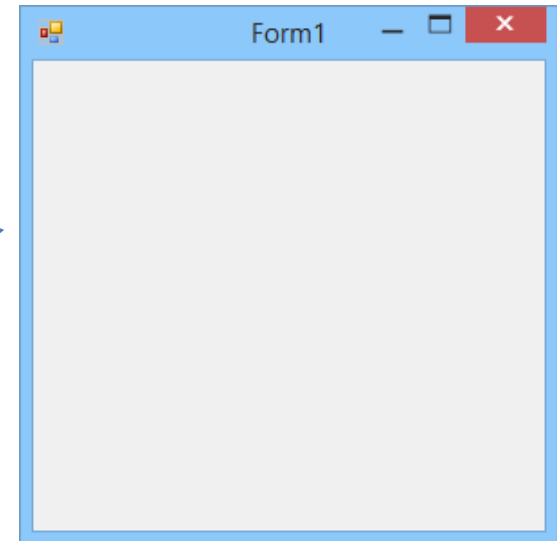
Στις εφαρμογές **Windows Forms Applications** η **φόρμα (form)** αποτελεί το βασικότερο αντικείμενο - εργαλείο.

Για τον χρήστη η φόρμα δεν είναι τίποτε περισσότερο από μια επιφάνεια επάνω στην οποία μπορεί κανείς να εμφανίζει πληροφορίες.



```
Public Class Form1
End Class
```

Για τον προγραμματιστή όταν επιλέξει μια τέτοια εφαρμογή (File/New Project/Windows Forms Application) η **φόρμα εμφανίζεται ως μια κλάση** που δείχνει «άδεια» (χωρίς κώδικα) όμως «τρέχοντας» την εφαρμογή διαπιστώνουμε ότι **το αντικείμενο (φόρμα)** που δημιουργείται παρουσιάζει ένα σύνολο από χαρακτηριστικά αλλά και συμπεριφορά **γιατί;**

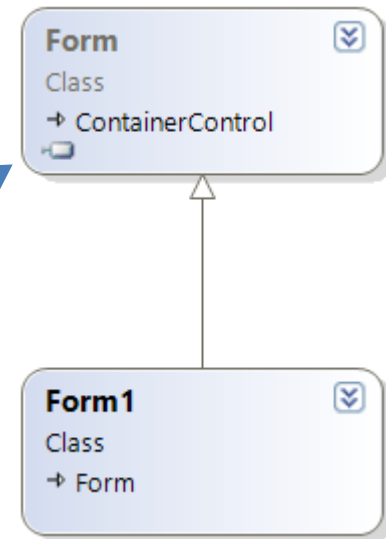
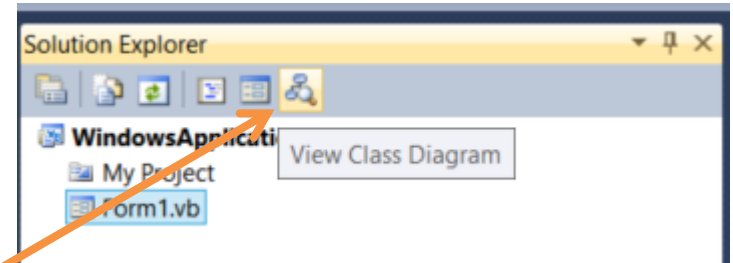


Από πού κληρονομεί η Φόρμα

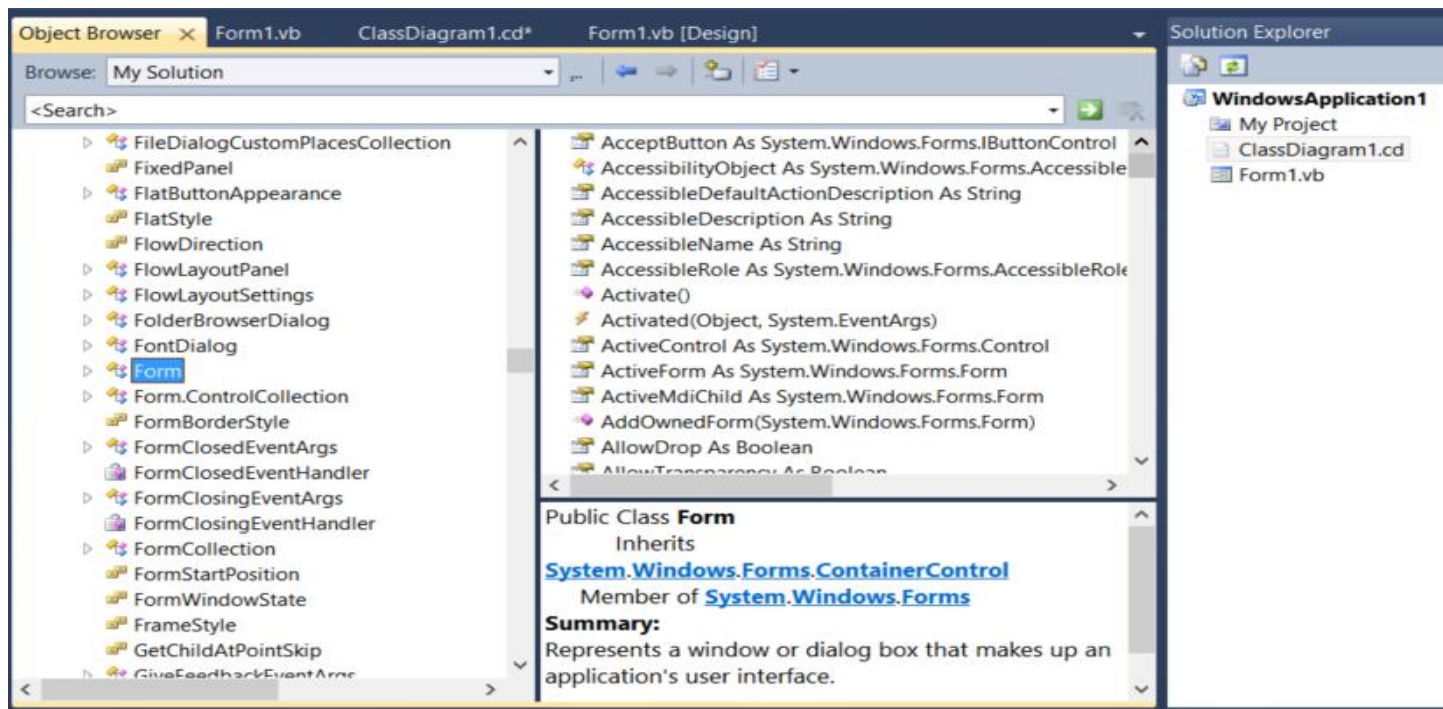
Είναι φανερό ότι τα χαρακτηριστικά και η συμπεριφορά του αντικειμένου φόρμα από «κάπου» κληρονομήθηκαν.

Αυτό μπορούμε να το διαπιστώσουμε ενεργοποιώντας την επιλογή «**View Class Diagram**» από το παράθυρο «Solution Explorer».

Θα εμφανιστεί ένα σχήμα που παριστάνει την κλάση Form1 από το οποίο μπορούμε να επιλέξουμε με δεξί κλικ το «**Show Base Class**», τότε θα δούμε το διπλανό σχήμα που δηλώνει ότι η κλάση Form1 κληρονομεί από την κλάση Form (ή πιο συγκεκριμένα από την κλάση της βιβλιοθήκης .NET με όνομα **System.Windows.Forms.Form**)



Η βασική κλάση Form στον Object Browser



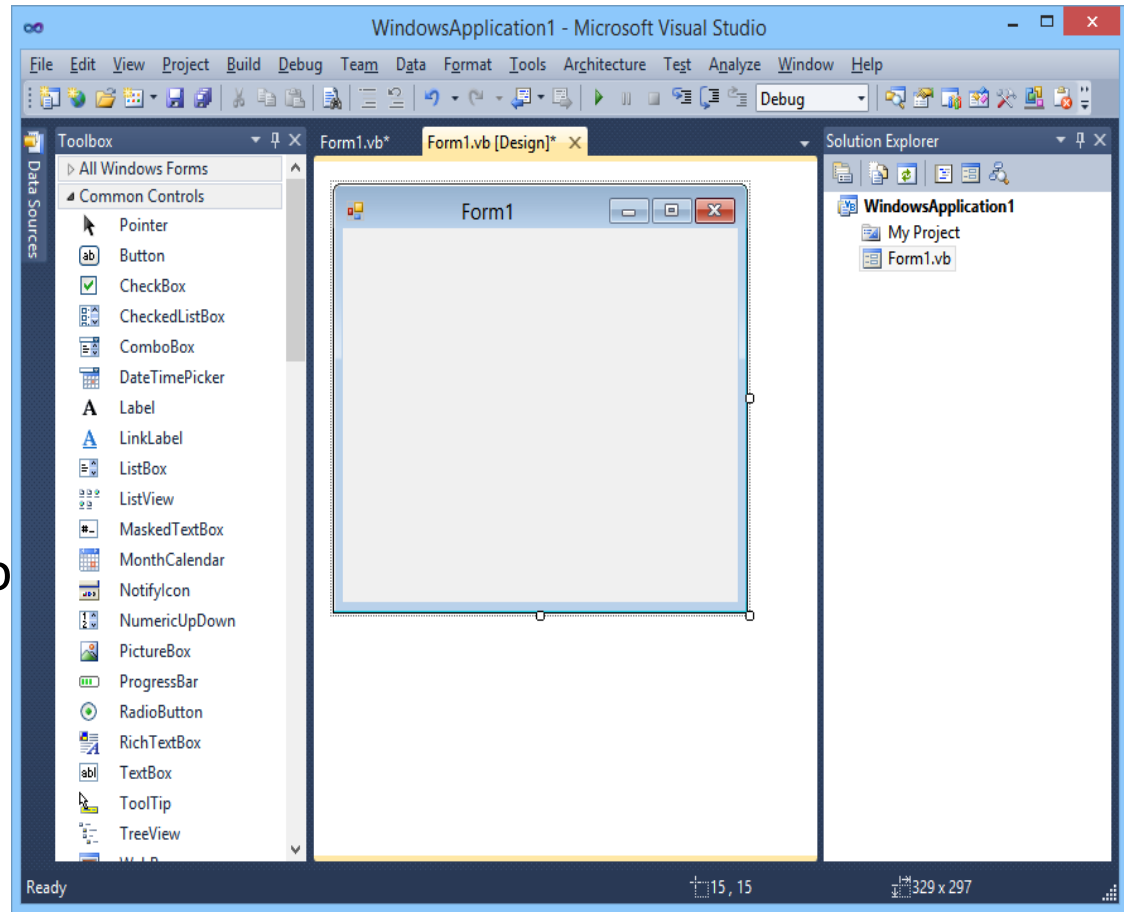
Μπορούμε να δούμε τα διάφορα αντικείμενα της βιβλιοθήκης της γλώσσας ενεργοποιώντας τον **Object Browser** από το μενού View/Object Browser

Στην παραπάνω εικόνα βλέπουμε στο αριστερό τμήμα του παραθύρου του **Object Browser** να είναι επιλεγμένη η κλάση Form (System.Windows.Forms.Form). Στο δεξί πάνω τμήμα βλέπουμε τα μέλη της κλάσης αυτής ενώ στο δεξί κάτω τμήμα την περιγραφή της

Πως εργαζόμαστε με φόρμες (1)

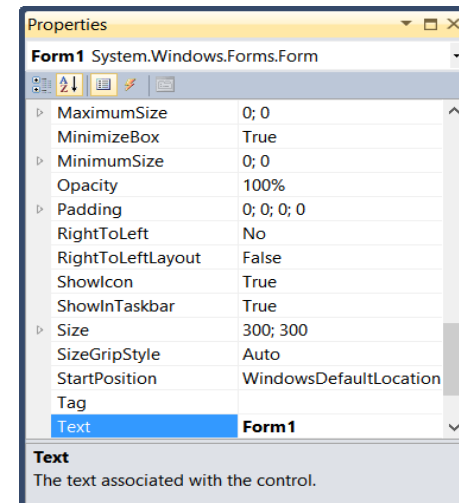
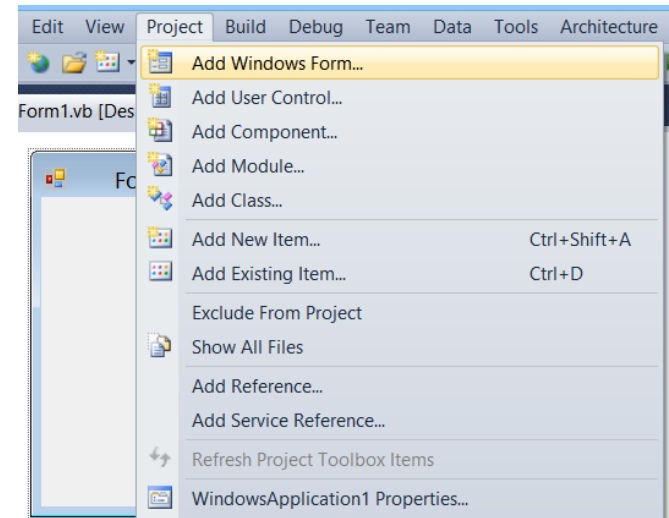
Επιλέγοντας **File=>New Project** και μετά **Windows Forms Application** της γλώσσας **Visual Basic** το Visual Studio 2010 μας δίνει αυτόματα το επόμενο περιβάλλον εργασίας.

Διακρίνουμε τη μονάδα κώδικα "**Form1.vb**" (σε κατάσταση σχεδίασης) και το παράθυρο "**Toolbox**" με τα αντικείμενα - χειριστήρια (**controls**) που μπορούμε να εισάγουμε στη φόρμα με όνομα "**Form1**".



Πως εργαζόμαστε με φόρμες (2)

- Όπως έχουμε αναφέρει οι φόρμες χρησιμεύουν για τη διασύνδεση του χρήστη με την εφαρμογή. Μια εφαρμογή μπορεί να χρειαστεί μια ή περισσότερες φόρμες για να καλύψει τις απαιτήσεις διασύνδεσης με τον χρήστη.
- Μπορούμε να προσθέσουμε όσες φόρμες θέλουμε από το μενού **Project/Add Windows Form**
- Ο προγραμματιστής μπορεί να κάνει διάφορες ρυθμίσεις στις ιδιότητες μιας φόρμας χρησιμοποιώντας το παράθυρο «**Properties**»



Μερικές Ιδιότητες Φόρμας (1)

Ιδιότητα	Περιγραφή
(Name)	Αποτελεί το όνομα της φόρμας ή καλύτερα το όνομα της κλάσης που είναι τύπου φόρμας. Εξορισμού ονομάζεται FormX, όπου X είναι ένας αριθμός
AcceptButton	Προσδιορίζει ποιο χειριστήριο τύπου «Button» που ήση έχει τοποθετηθεί στη φόρμα θα πρέπει να ενεργοποιηθεί αν ο χρήστης πατήσει το πλήκτρο «Enter».
AutoScroll	Αν δώσουμε την τιμή True, η φόρμα αυτόματα προσθέτει μπάρες κύλισης για να δείξει περιεχόμενα της φόρμας όταν αυτά δεν φαίνονται
BackColor	Το χρώμα που φαίνεται στο υπόβαθρο της φόρμας
BackgroundImage	Μαζί με την ιδιότητα BackgroundImageLayout μπορούμε να τοποθετήσουμε ένα γραφικό στο υπόβαθρο της φόρμας
FormBorderStyle	Εξορισμού τιμή είναι το «Sizable» που επιτρέπει στη χρήστη να αλλάξει μέγεθος στη φόρμα. Με το «None» εξαφανίζονται οι μπάρες τίτλου και τα borders. Κάτι τέτοιο είναι χρήσιμο σε μια οθόνη καλωσορίσματος
Icon	Καθορίζει το εικονίδιο που θα εμφανίζεται στο πάνω αριστερό άκρο της φόρμας

Μερικές Ιδιότητες Φόρμας (2)

Ιδιότητα	Περιγραφή
IsMdiContainer	Ενεργοποιεί τη δυνατότητα «multiple document interface» της φόρμας. Αυτό επιτρέπει στη φόρμα να μπορεί να περιέχει πολλαπλές άλλες φόρμες
Size	Προσδιορίζει το μέγεθος της φόρμας με δύο τιμές (υπο-ιδιότητες) πλάτος – ύψος.
ShowInTaskbar	Προσδιορίζει αν θα εμφανιστεί στην μπάρα κατάστασης του συστήματος
StartPosition	Προσδιορίζει τη θέση που θα εμφανιστεί την πρώτη φορά στην οθόνη η φόρμα
Tag	Μπορούμε να βάλουμε οποιοδήποτε δεδομένο θέλουμε
Text	Αποτελεί την επικεφαλίδα τίτλου της φόρμας
TopMost	Αν δώσουμε τιμή true η φόρμα θα εμφανιστεί μπροστά από κάθε άλλη
WindowState	Προσδιορίζει την τρέχουσα κατάσταση του παραθύρου

Μερικές Μέθοδοι Φόρμας

Μέθοδος	Περιγραφή
Show	Εμφανίζει τη φόρμα στο χρήστη
Showdialog	Εμφανίζει τη φόρμα στο χρήστη ως παράθυρο διαλόγου
Hide	Αποκρύπτει τη φόρμα από τον χρήστη
Focus	Η φόρμα αποκτά εστίαση
Close	Κλείνει η φόρμα

Οι ιδιότητες και οι κλήσεις σε μεθόδους μπορεί να δοθούν με κώδικα όπως:
(όπου MyForm, MyForm1, MyForm2 είναι μεταβλητές αντικειμένου τύπου Form)

```
MyForm.Text = "Yannis"
```

```
MyForm.Hide()
```

```
MyForm2.Text = MyForm1.Text
```

```
Me.BackColor = Color.Azure
```

```
MyForm.Close()
```

Συμβάντα & Χειριστές συμβάντων

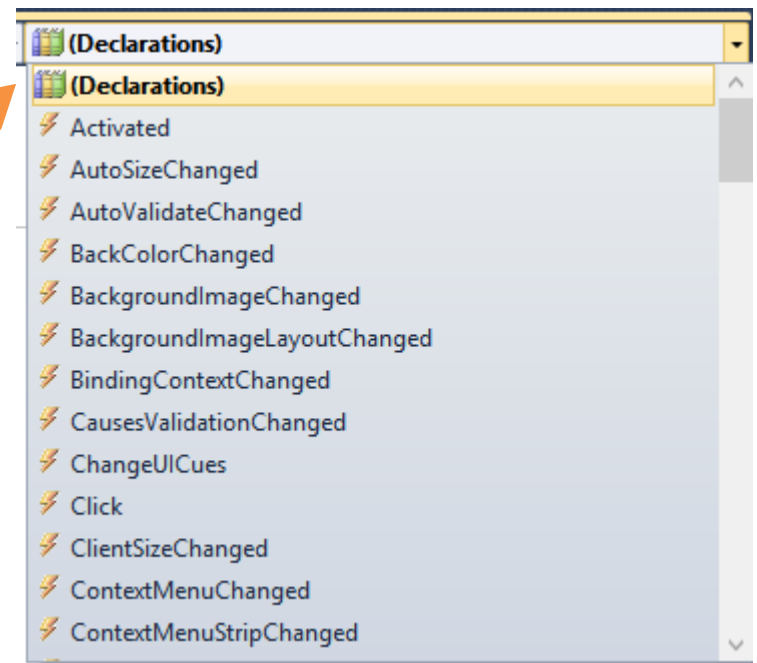
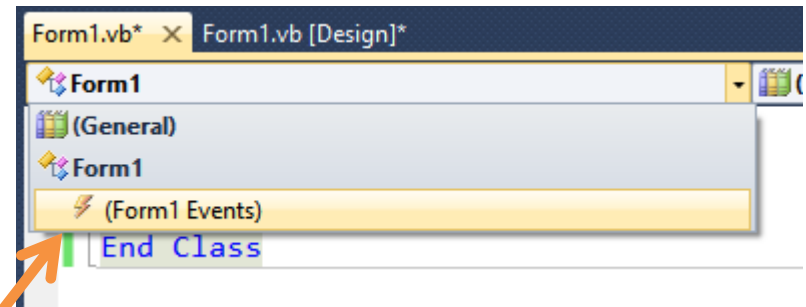
- Μια σημαντική λειτουργία των αντικειμένων, όπως έχουμε αναφέρει, είναι η ικανότητά τους να "**αντιδρούν**" σε "**συμβάντα**" (**events**) που προκαλούνται από τον χρήστη, κώδικα προγράμματος ή το ίδιο το σύστημα.
- Στον προγραμματισμό αυτή η «αντίδραση» σημαίνει την **εκτέλεση** κάποιου τμήματος κώδικα δηλαδή κάποιας διαδικασίας. Αυτό επιτυγχάνεται με τις διαδικασίες που ονομάζουμε "**χειριστές συμβάντων**" (**event handlers**). Οι διαδικασίες αυτές εντάσσονται στις μεθόδους του αντικειμένου και είναι μέθοδοι τύπου Sub με εμβέλεια ιδιωτική (private) (**γιατί?**) με μόνη διαφορά ότι **εκτελούνται αυτόματα** μόλις προκληθεί το συμβάν για το οποίο φτιάχτηκαν να αντιδρούν.
- Το λειτουργικό σύστημα "**αντιλαμβάνεται**" τα συμβάντα που προκαλούνται π.χ. από έναν χρήστη όταν κάνει "**κλικ**" με το ποντίκι του σε μια φόρμα ή σε ένα αντικείμενο textbox, κ.ο.κ. και επιπλέον **διαβιβάζει τον έλεγχο** στον κώδικα **χειριστή του συμβάντος** του συγκεκριμένου αντικειμένου (αν υπάρχει).

Μερικά Ενδεικτικά Συμβάντα

- Συμβάντα που αφορούν τις ενέργειες που κάνουμε με το ποντίκι μας
 - Click, MouseClick, MouseDown, MouseHover, MouseLeave, MouseMove, MouseUp
- Συμβάντα που αφορούν ενέργειες που κάνουμε με το πληκτρολόγιο
 - KeyDown, KeyUp, KeyPress,
- Άλλα χρησιμοποιούμενα συμβάντα:
 - Activated, Deactivate, Closed, Load, GotFocus, LostFocus,

Γράφοντας κώδικα σε χειριστές συμβάντων φόρμας

- Για να δούμε τα συμβάντα στα οποία μπορεί να «αντιδράσει» μια φόρμα ανοίγουμε το παράθυρο της ενότητας κώδικα της φόρμας μας (**View Code**) και επιλέγουμε «**Form1 Events**» από το πάνω αριστερό πλαίσιο. Στη συνέχεια από το **δεξί πλαίσιο** (declarations) μπορούμε να δούμε όλα τα συμβάντα.



Ο χειριστής συμβάντος “Click” της Φόρμας

```
(General) (Declarations)
Public Class Form1
    Private Sub Form1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Click
        MsgBox("Έκανες κλικ στο αντικείμενο φόρμα!!")
    End Sub
End Class
```

- Επιλέγοντας από τα συμβάντα που μπορεί να αντιδράσει ένα αντικείμενο φόρμας το «Click» συμβάν θα δούμε τη διαδικασία

Private Sub Form1_Click

- Το πιο σημαντικό τμήμα της παραπάνω διαδικασίας είναι η δήλωση **Handles Me.Click** όπου δηλώνεται ότι η συγκεκριμένη διαδικασία διαχειρίζεται το Click συμβάν του αντικειμένου φόρμας (**Me**). Αυτό σημαίνει ότι όταν προκληθεί το συγκεκριμένο συμβάν (δηλαδή ο χρήστης κάνει κλικ με το ποντίκι του πάνω από την επιφάνεια της φόρμας) θα εκτελεστεί ο κώδικας της διαδικασίας **Form1_Click**
- Δοκιμάστε να τρέξετε τον παραπάνω κώδικα και δείτε την συγκεκριμένη συμπεριφορά. Με τον ίδιο τρόπο μπορείτε να δοκιμάσετε και άλλα συμβάντα

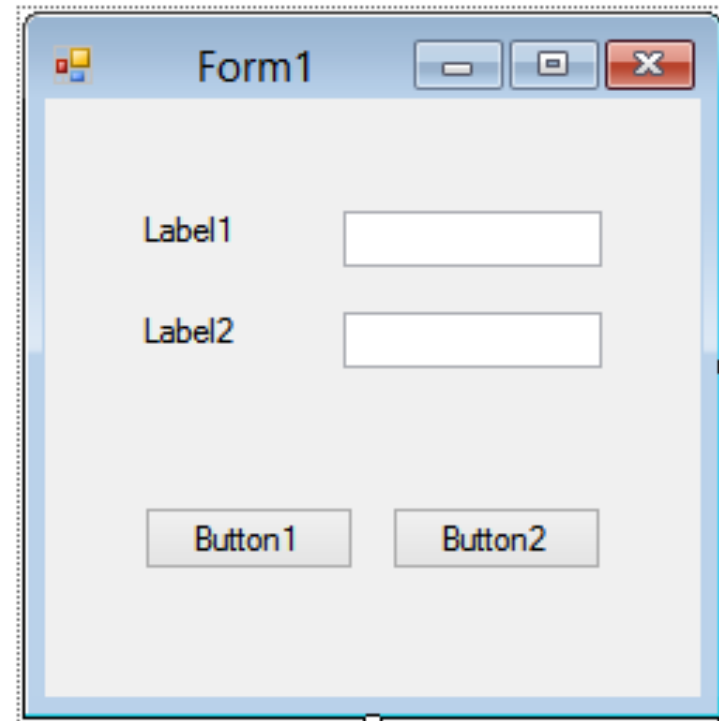
Ο χειριστής συμβάντος “Load” της Φόρμας

```
(General) (Declarations)
Public Class Form1
    Private Sub Form1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Click
        MsgBox("Έκανες κλικ στο αντικείμενο φόρμα!!")
    End Sub
    Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        Me.Text = "Form Loaded"
        Me.BackColor = Color.Cyan
    End Sub
End Class
```

- Ένα άλλο παράδειγμα συμβάντος για ένα αντικείμενο φόρμας είναι το «**Load**» συμβάν. Το συγκεκριμένο συμβάν «πυροδοτείται» μόλις φορτωθεί η φόρμα στη μνήμη του Η/Υ και πριν εμφανιστεί.
- Με ένα τέτοιο συμβάν μπορούμε να διαχειριστούμε τις αρχικές τιμές που θέλουμε να έχει η φόρμα μας μόλις φορτωθεί για πρώτη φορά
- Δοκιμάστε να τρέξετε τον παραπάνω κώδικα και δείτε την συμπεριφορά του

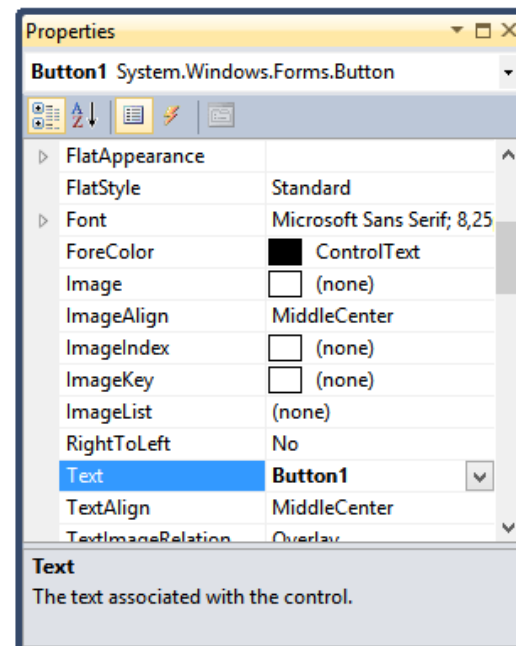
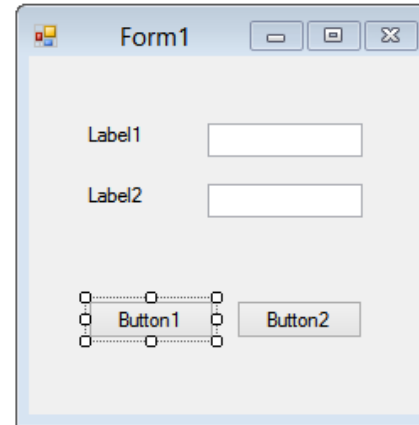
Εισαγωγή αντικειμένων – χειριστηρίων σε φόρμα

- Η εισαγωγή χειριστηρίων στη φόρμα μας γίνεται κυρίως σε κατάσταση σχεδίασης με τον εξής τρόπο:
- Επιλέγουμε αντικείμενο από το παράθυρο "**Toolbox**" & από την κατηγορία χειριστηρίων που μας ενδιαφέρει π.χ. "**Common Controls**" και το σύρουμε μέσα στο χώρο σχεδίασης της φόρμας.
- Στη διπλανή εικόνα έχουμε σχεδιάσει μέσα στη φόρμα από 2 αντικείμενα των παρακάτω τύπων: Label, Textbox και Button



Πως εργαζόμαστε με τα χειριστήρια

- Με παρόμοιο τρόπο εργαζόμαστε και με τα αντικείμενα – χειριστήρια που μπορούμε να εισάγουμε σε μια φόρμα.
- Έχοντας ενεργοποιηθεί το παράθυρο «Properties» όταν επιλέγουμε ένα χειριστήριο εμφανίζονται οι ιδιότητές του στο συγκεκριμένο παράθυρο



Πως εργαζόμαστε με τα χειριστήρια

- Οι ιδιότητες και οι κλήσεις σε μεθόδους που υποστηρίζει ένα χειριστήριο μπορεί να δοθούν και με κώδικα όπως:

```
Button1.Text = "Αποθήκευση"
```

```
Button2.Text = "Άκυρο"
```

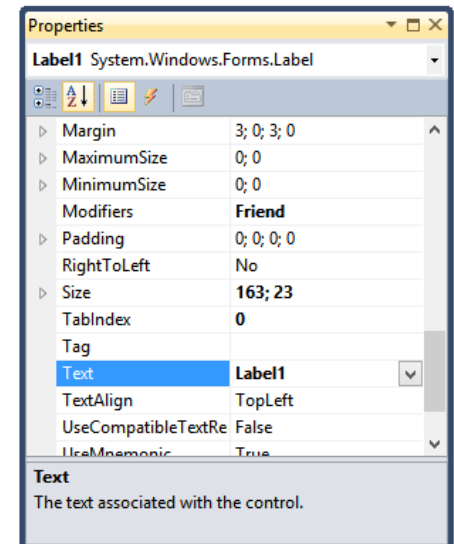
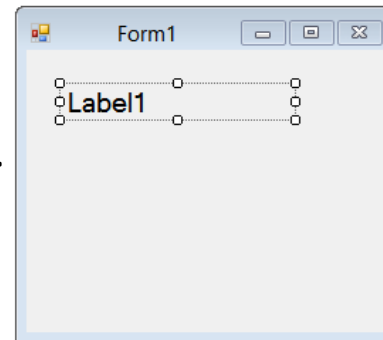
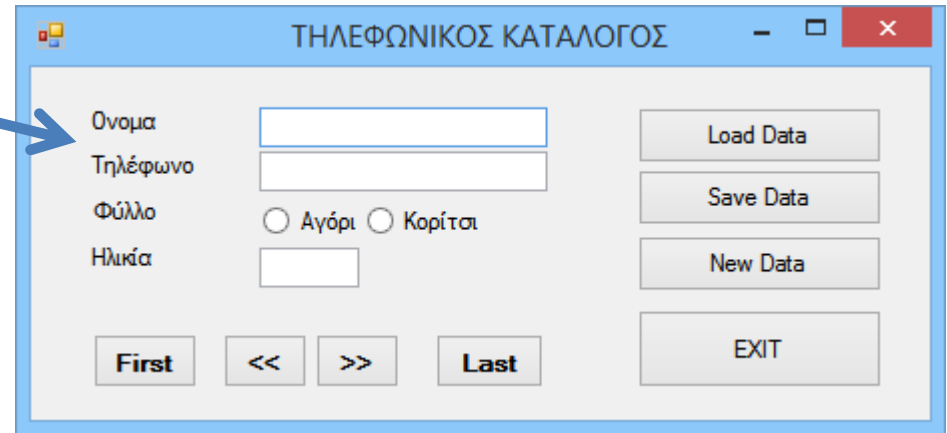
```
Button1.BackColor=Color.Aqua
```

```
Button2.Hide()
```

```
Button1.Focus()
```

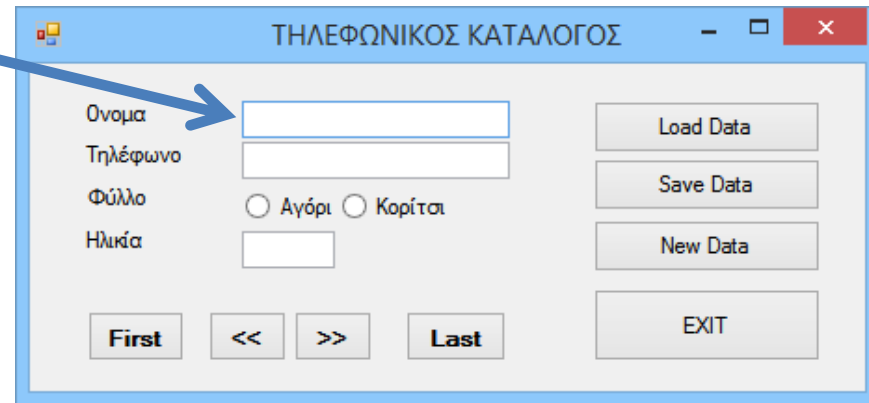
Το χειριστήριο Label

- Το **Label** (ετικέτα) χρησιμοποιείται για την εμφάνιση κειμένου ή εικόνων που δεν μπορούν να τροποποιηθούν από το χρήστη.
- Συνήθως χρησιμοποιείται για να προσδιορίσει άλλα αντικείμενα σε μια φόρμα - να παρέχει στην ουσία μια περιγραφή του τι θα κάνει ένα διπλανό αντικείμενο - χειριστήριο (πχ TextBox).

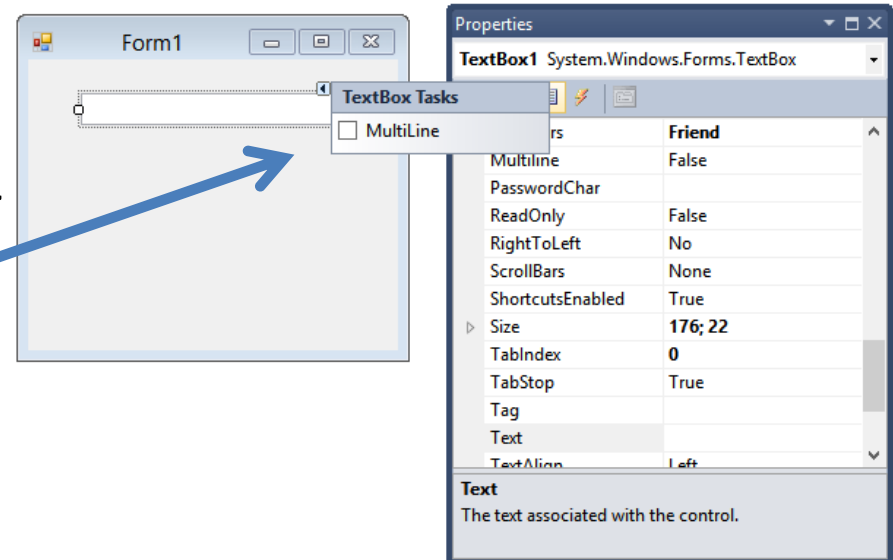


Το χειριστήριο TextBox

- Το **TextBox** (πλαίσιο κειμένου) χρησιμοποιείται για να πάρουμε είσοδο από το χρήστη ή να εμφανίσουμε ένα κείμενο. Το συγκεκριμένο χειριστήριο χρησιμοποιείται γενικά για επεξεργάσιμο κείμενο, αν και μπορεί να εμφανίσει κείμενο μόνο για ανάγνωση.

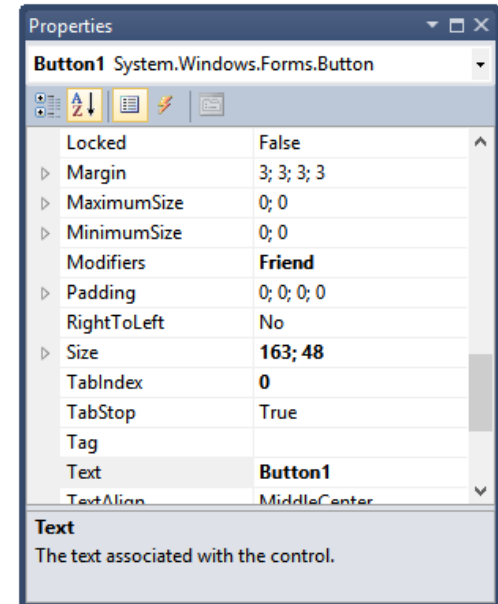
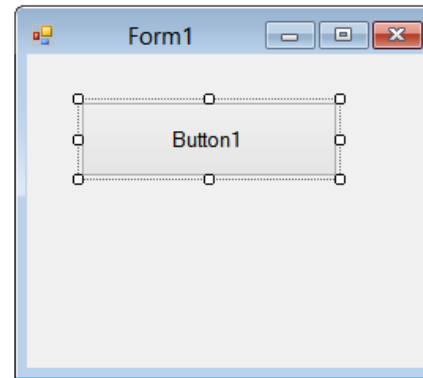
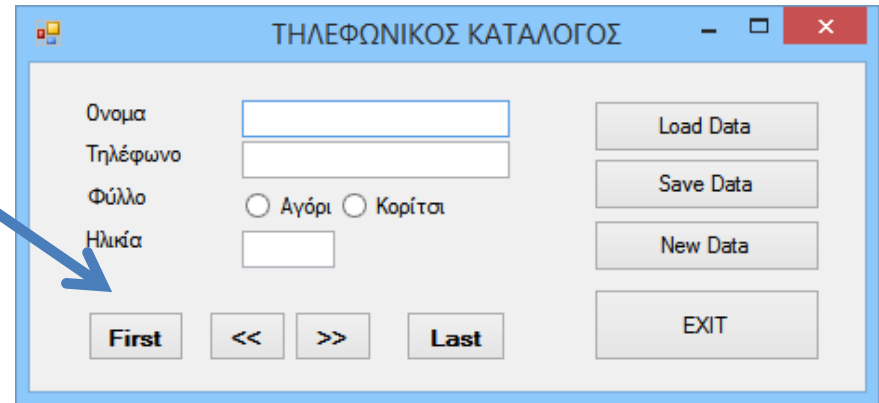


- Τα πλαίσια κειμένου μπορεί να εμφανίζονται σε πολλαπλές γραμμές και μπορούμε να προσθέσουμε βασική μορφοποίηση.



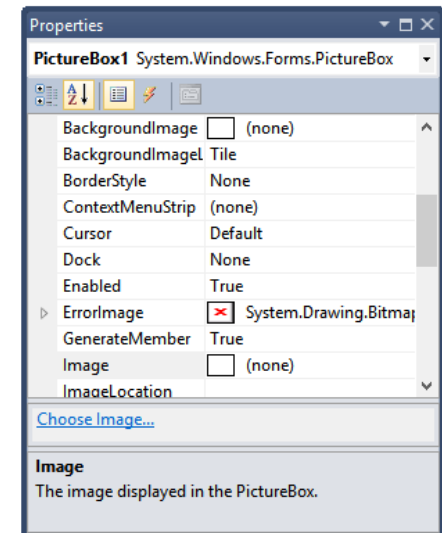
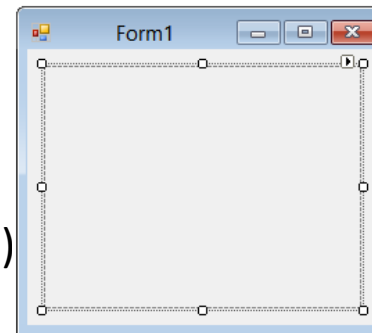
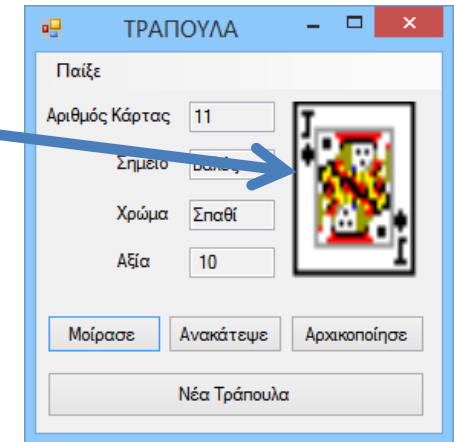
Το χειριστήριο Button

- Το **button** (πλήκτρο διαταγής) είναι ένα αντικείμενο - χειριστήριο που μας επιτρέπει να κάνουμε επάνω του κλικ και να ενεργοποιούμε την εκτέλεση κάποιας εργασίας της εφαρμογής μας



Το χειριστήριο PictureBox

- Το **PictureBox** (πλήκτρο διαταγής) είναι ένα αντικείμενο - χειριστήριο που μας επιτρέπει να εισάγουμε εικόνες στην εφαρμογή μας
- Για να εισάγουμε μια εικόνα στο αντικείμενο PictureBox χρησιμοποιούμε την ιδιότητα **image**.
- Άλλες χρήσιμες ιδιότητες του PictureBox είναι το **AutoSize** που προσδιορίζει αν θα επεκταθεί το PictureBox να καλύψει την εικόνα καθώς και το **SizeMode** που μπορεί να χρησιμοποιηθεί για να επεκταθεί η εικόνα ή να κεντραριστεί ή να γίνει zoom στο PictureBox.
- Είναι σημαντικό να γνωρίζουμε ότι πριν εισάγουμε μια εικόνα θα πρέπει να εισάγουμε το αρχείο της ως Resource (πόρος) στην εφαρμογή μας.
- Αυτό το κάνουμε μια φορά και το επαναχρησιμοποιούμε όσες φορές θέλουμε.



Τέλος Ενότητας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

