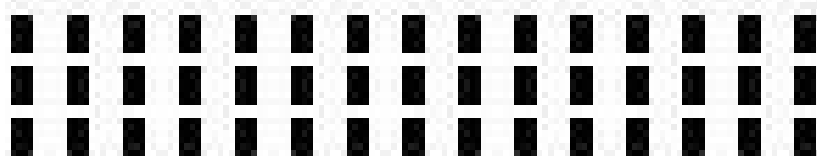
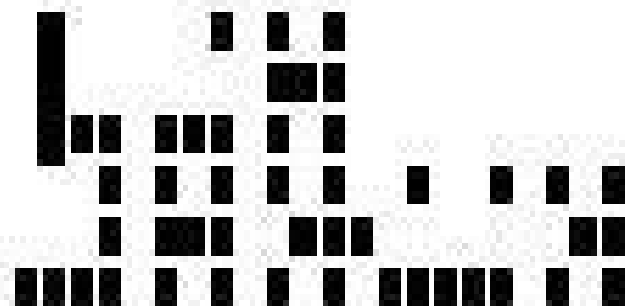


VISUAL BASIC



Кратко ръководство



visual studio 2008

	iii					lll	bb			iii					
vv	vv	sss	uu	uu	aa	aa	lll	bb	aa	aaa	sss	cccc			
vv	vv	iii	s	uu	uu	aa	aaa	lll	bbbbbb	aa	aaa	s	iii	cc	
v	v	iii	sss	uu	uu	aa	aaa	lll	bb	bb	aa	aaa	sss	iii	cc
		iii		s	uuuu	u	aaa	aa	lll	bbbbbb	aaa	aa	s	iii	cccc
			sss										sss		

VISUAL BASIC

Кратко ръководство

8 бита=1 байт :)

Програмна конструкция (statement) – тя е ред от кода на програмата на visual basic (vb)

[BASIC означава (основен) Beginners All-Purpose Symbolic Instruction Code]

една самосъдържаща се инструкция, изпълнявана от компилатора, която извършва полезна работа в приложението. Програмните конструкции могат да варират по дължина (някои съдържат само 1 ключова дума на vb) но всички те трябва да следват синтактичните правила, дефинирани и налагани от компилатора.

В VB Studio 2005 или 2008 програмните конструкции се състоят от ключови думи, св-ва, имена на обекти, променливи, числа, специални символи и др. стойности.

Ключова дума (keyword) – тя е запазена дума, която се разпознава от компилатора на vb и извършва полезна работа. Примерно ключовата дума End спира изпълнението на програмата. Ключовите думи работят заедно с обекти, св-ва, променливи и др. стойности. Оцветяват се в синьо в код едитора (Code Editor)

Променлива (Variable) – използва се за временно съхранение на данни в програмите.

Програмистът създава променливи посредством конструкцията Dim и след това ги използва за съхранение на резултати от изчисления, имена на файлове, входни данни...

В променливите могат да бъдат съхранявани числа, имена и стойности на свойствата.

Контрол (Control) – инструмент, който използвате за създаване на обекти в/у формите на VB

Контролите се избират от Toolbox и се използват за чертане на обекти с мишката в/у формите (поставянето на бутончета в/у формата)

Повечето контроли се използват за създаване на елементи от потребителския интерфейс като бутоните, прозорци за изображения, списъчни полета...

Обект (Object) – елемент от потребителския интерфейс, който се създава в/у формата на VB посредством контрол от Toolbox. **Във VB самата форма също е обект!!!**

Преместването, оразмеряването на обект става чрез настройка на свойствата(properties)

Обектите притежават нещо, наречено присъща функционалност – обектите знаят как да работят и могат да отговорят самостоятелно на определени ситуации (примерно списъчното поле знае как да се превърта:)

Клас (Class) – е плана или шаблона за един или повече обекти и дефинира какво прави обекта!

Класът дефинира какво може да прави обекта, но той самият не е обект!

Във VB.NET 2003 и 2005 може да се използват съществуващи класове на Visual Studio като System.Math и System.Windows.Forms.Form, а може да се изграждат и собствени класове и да наследяват св-ва, събития от тях. Наследяването дава възможност на 1 клас да придобие вече съществуващи характеристики на интерфейса и поведението от друг клас.

Пространство от имена (namespace) – е йерархия на библиотека от класове, организирани под уникално име като System.Windows или System.Diagnostics

За да имаме достъп до класовете и обектите от дадено пространство от имена, трябва да се постави конструкцията Imports в началото на програмния ни код.

Свойство (Property) – е стойност или характеристика притежавана от обект.

Например обектът бутон притежава свойството Text, което определя текста на бутона:)

както и и св-то Image, което задава пътя да файла с изображението в/у него.

В кода, форматът за задаване на свойство е:

Обект.Свойство = Стойност

където Обект е името на обекта

Свойството е характеристиката, която искаме да променим, а Стойност е новата стойност на свойството.

Например:

Button1.Text = "Hello"

(задаваме стойност "Hello" за св-то Text на обект Button1)

Процедури за събития (event procedure) – е блок от код, който се изпълнява, когато с някой обект се извърши манипулация в програмата. Когато се натисне бутона Button1 се изпълнява процедурата Button1_Click

Метод (method) – специална конструкция извършваща действие или услуга за даден обект в програма.

В кода означаването на методите е:

Обект.Метод (Стойност)

където Обект е име на обект, Метод е действието, което искаме да извършим, а Стойност е незадължителен

аргумент.

Например:

ListBox1.Items.Add("Chek")

използва метода Add , за да постави думата Check(провери,проверка) в списъчното поле ListBox1

Методите и Свойствата често се идентифицират по тяхната позиция в колекция или библиотека от обекти, така че се срещат дълги обръщения от типа на System.Drawing.Image.FromFile , което се чете като метода FromFile , който е член на класа Image , който е член на библиотеката от обекти System.Drawing

Създаване на link(връзка) в приложение:

от Toolbox си избираме контрола LinkLabel и после в кода се пише:

LinkLabel1.LinkVisited = True

System.Diagnostics.Process.Start ("http:// " " ")

Особеност на метода Process.Start е, че той може да се използва за стартиране на други Windows приложения.

Например:

System.Diagnostics.Process.Start ("IExplore.exe, "http:// " " ")

или

System.Diagnostics.Process.Start ("Winword.exe", "C:\файла дето ще се стартира.doc")

(без да е подчертано:)

За създаване на менюта се ползват контролите: MenuStrip и Toolstrip

Клавиши за достъп: за да добавите клавиш за достъп към елемент от меню активирате Menu Designer и въвеждате амперсанд (&) пред съответната буква в името на менюто. В името кликате още един път, за да се покаже курсора I и се въвежда & накрая натискате клавиша Enter

Клавишни комбинации за менюта: контролът MenuStrip дава възможност за присвояване на клавишни комбинации за менютата. Чрез свойството ShortcutKeys на контрола MenuStrip има възможност да се конфигурира тази настройка.

Използване на контроли за диалогови прозорци:

Стандартни диалогови прозорци в страници Dialogs и Printing на Toolbox

OpenFileDialog - набавя устройството, името на папката и името на съществуващ файл

SaveFileDialog - набавя устройството, името на папката и името на нов файл

FontDialog - дава възможност на потребителя да избере нов тип и стил на шрифта

ColorDialog - дава възможност на потребителя да избере цвят от палитрата

FolderBrowserDialog - дава възможност на потребителя да разглежда структурата от папки на компа

PrintDialog - дава възможност на потреб. Да задава опции за отпечатването

PrintPreviewDialog - отваря диалогов прозорец за предварителен преглед на печата

PageSetupDialog - потребителя контролира опциите за настройка на страници, като поле, размер на хартията и разположение.

Използване на променливи за съхранение на информацията:

Променливата(variable) както споменах е място за временно съхранение на данни е програмата. Те може да са колкото си искаме на брой:) Съдържат думи, числа, свойства.

Променливата трябва да се декларира!!!

Заделяне на място за променливи става с конструкцията Dim (dimension) (размер,измерване, измерение:)

Dim LastName As String

тази конструкция заделя място за променлива с име LastName , която ще съдържа текстова или още казано низова стойност.

Променлива за съхранение на входни данни: ползва се функцията InputBox

Ползване на променливи за извеждане на данни – полезна функция за визуализиране на изходни данни е

MsgBox (нея я знаете:)

ButtonClicked = MsgBox (Заглавие)

Работа с конкретни типове данни:

...тъпът данни String (низ) е полезен за управление на текст в програмата

тип данни	интервал	примерна употреба
-----------	----------	-------------------

Short	-32768 до 32767	Dim Ptici As Short Ptici = 12500
UShort	0 до 65535	Dim Dni As Ushort Dni = 55000
Integer	-2147483648 до 2147483647	Dim Nasekomi As Integer Nasekomi = 37500000
UInteger	0 до 4294967295	Dim Igrachki As Uinteger Igrachki = 3000000000
Long	-9223372036854775808 до 9223372036854775807	Dim WorldPop As Long WorldPop = 4800000004
ULong	0 до 18446744073709551615	Dim Stars As ULong Stars = 18000000000000000000
Single	-3.4028235E38 до 3.4028235E38	Dim Cenata As Single Cenata = 899.99
Double	-1.79769313486231E308 до 1.79769313486231E308	Dim Pi As Double Pi = 3.1415926535
Decimal	стойности до +/-79,228 x 1024	Dim Debt As Decimal Debt = 7600300.50
Byte	0 до 255 но без отрицателни номера	Dim RetKey As Byte RetKey = 13
SByte	-128 до 127	Dim NegVal As Sbyte NegVal = -20
Char	всеки Unicode символ в интервала 0-65535	Dim UnicodeChar As Char UnicodeChar = “A”
String	0 до към 2 милиарда 16-bit Unicode знака,символа(characters)	Dim Kuchence As String Kuchence = “karakachanka”
Boolean	True or False / Истина или Неистина	Dim Flag as Boolean Flag = True
Date	January 1, 0001, до December 31, 9999	Dim Data As Date Data = #3/1/2006#
Object	всякакъв тип може да бъде съхраняван в променлива от тип Object	Dim MyApp As Object MyApp = CreateObject (“Word.Application”)

Работа с операторите на Visual Basic:

Формулата е конструкция, която комбинира числа, променливи, оператори и ключови думи и има за цел да създаде нова стойност.

<u>Operators / Операторите</u>	<u>Description / Описание</u>
+	Addition / събиране
-	Subtraction / изваждане
*	Multiplication / умножение
/	Division / деление
\	Integer (whole number) division / целочислено деление
Mod	Remainder division / деление по модул, остатък от деление
^	Exponentiation (raising to a power) / степенуване
&	String concatenation (combination) / конкатенация на низове

Съкратени оператори (Shortcut Operators):

Operation / операция	Long-form syntax / синтаксис в дълга форма	Shortcut syntax / кратък синтаксис
Addition / събиране (+)	$X = X + 6$	$X += 6$
Subtraction / изваждане (-)	$X = X - 6$	$X -= 6$
Multiplication / умножение (*)	$X = X * 6$	$X *= 6$
Division / деление (/)	$X = X / 6$	$X /= 6$
Integer division / целоч. деление (\)	$X = X \backslash 6$	$X \backslash = 6$
Exponentiation / степенуване (^)	$X = X ^ 6$	$X ^ = 6$
String concatenation / конкатенация на низове (&)	$X = X \& \text{"ABC"}$	$X \& = \text{"ABC"}$

Използване на условни изрази:

Условният израз е част от цяла програмна конструкция, която задава въпрос за св-во, променлива като отговора е True или False

оператор за сравнение

значение

=	равно на
< >	различно от
>	по-голямо от
<	по-малко от
>=	по-голямо или равно на
<=	по-малко или равно на

Версии на Visual Basic. NET

Първата версия на Visual Basic .NET (Microsoft Visual Basic. NET 2002) е издадена през февруари 2002 година. Втората версия (Microsoft Visual Basic. NET 2003) става достъпна през март 2003. След дълъг период на развитие и работа излиза Microsoft Visual Basic 2005 г. в края на 2005 г. Visual Basic 2005 г. е вече толкова силно интегриран с Visual Studio, че тя е достъпна само като компонент в Visual Studio 2005 , който включва Visual C #, Visual C ++ и Visual J # компилаторите и други .NET инструменти за разработка, а по-късно и F# компилаторът...

Предисторията на Визуъл Бейсик (Visual Basic)

- Създаден е Visual Basic за Windows 1.0 (май 1991).
- Създаден е Visual Basic 1.0 за DOS през септември 1992.
- Самият език не е съвместим с Visual Basic за Windows, той е следваща версия на DOS-базираните компилатори на Microsoft — QuickBASIC и BASIC Professional Development System. Интерфейсът трудно може да се нарече графичен, защото използва допълнителните ASCII символи за симулиране на графичен интерфейс.
- Създаден е Visual Basic 2.0 през ноември 1992. Програмната среда е лесна за използване и е с подобрена скорост.
- Реализирана е версия 3.0 на Visual Basic през лятото на 1993 и е разделена на Стандартна и Професионална версии. VB3 включва енджин за бази от данни, който може да чете и пише бази от данни Access.
- Visual Basic 4.0 (август 1995) е първата версия, способна да създава както 32-битови, така и 16-битови приложения за Windows. Предоставя възможността за писане на класове във Visual Basic.
- С версия 5.0 (февруари 1997), Microsoft реализира Visual Basic изключително за 32-битови версии на Windows. Също така предоставя възможността за компилиране до native изпълним код, елиминирайки необходимостта от runtime библиотеки.
- Visual Basic 6.0 (лятото 1998) е подобрен в много отношения, включително разработката на уеб-базирани приложения. VB6 влиза във фазата без поддръжка от Microsoft през март 2008.
- Visual Basic.NET е стартиран през 2002 заедно с.NET Framework. Неговите езикови възможности са далеч по-богати от предишните версии, но е и по-сложен и повечето програми писани на старите VB трябва да се пренаписват, за да работят на VB.NET.
- Visual Basic.NET 2003 е стартиран през 2003 заедно с.NET Framework 1.1
- През 2004 Microsoft реализира бета версия на Visual Basic 2005 (кодовото име е Whidbey) и през 2005г излиза.
- В края на 2005 г. на пазара вече е Visual Basic 2005. Стартира рекламна кампания за безплатно регистриране и ползване без граници на Visual Basic 2005 Express Edition.

Visual Studio 2005 и 2008 се продават в няколко различни конфигурации: Standard Edition, Professional Edition, Team Suite, както и Express Edition, като най-добре си е да работите с Professional Edition :)

Примерите, които смятам да Ви илюстрирам в тази ел. книга ще са за програмна среда Visual Studio 2005 и Visual Studio 2008 Professional Edition ,те са си кажи речи съвместими...просто има развитие в усъвършенстването на дебъгера и още някои неща:) примерите може да бъдат и изпълнени на Visual Basic 2005 или 2008 Express Edition. Както се досещате тези среди се развиват непрекъснато и четейки тези страници вече може би има нова версия:)

Полезни линкове:

<http://brigante.sytes.net/vb.aspx>

[http://msdn.microsoft.com/bg-bg/vbasic/bb466226\(en-us\).aspx#top](http://msdn.microsoft.com/bg-bg/vbasic/bb466226(en-us).aspx#top)

учебни видео клипове

Програмната среда Visual Basic 2008 Express Edition със SP1 може да я намерите на този адрес:

<http://www.microsoft.com/express/product/default.aspx>

Тя е безплатна и след бърза регистрация ще може да я свалите и ползвате. Трябва да се регистрира след това с безплатен сериен номер от сайта на Майкрософт...иначе може да се ползва само 30 дни.

Програмната среда Visual Studio 2008 Professional Edition може да си свалите от тук:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=83c3a1ec-ed72-4a79-8961-25635db0192b&displaylang=en>

Visual Studio 2010 and .NET Framework 4 Beta 1 още за тях е рано:)

но може да ги видите тук:

<http://msdn.microsoft.com/en-us/vstudio/dd582936.aspx>

Това е следващото поколение програмна среда, която ще работи и с .NET Framework 4

Visual Studio Team System 2008 Team Suite може да свалите от тук:

<http://msdn.microsoft.com/en-us/evalcenter/bb655862.aspx>

Разбира се трябва да следите и за обновяванията на програмната среда:

[Microsoft Visual Studio 2008 Service Pack 1 \(Installer\)](#)

Трябва да имате предвид, че файловете са доста обемисти и да съобразите дали имате място на харда си. Заема повече от 1 GB този сървис пак:)

Когато инсталирате програмната среда Visual Studio 2008 Professional Edition ще бъдете запитани дали да инсталирате и библиотеката MSDN – тя е отделен компонент на пакета Visual Studio .NET и тя представлява пълен,обемист справочник, предназначен за програмистите...има техническа инфо в областта на програмирането, както и примерни сорс кодове, много документация, разни статии и т.н. Нея можете да си я инсталирате и в по-късен етап.

На преминаващите от Visual Basic 6 ще им е доста некомфортно в тази нова среда:)

Но са малко програмистите, които работят все още с Visual Basic 6

Може би някои работят на нея понеже имат стари готови проекти т.е. сорс кодове писани на Visual Basic 6 а понякога мигрирането от стар код към нов е изпълнено с доста неясни грешки показани от дебъгера и програмиста се чуди и мае какво да прави:)

Не е нужно да сте програмирали с Visual Basic 6 , за да правите това с новите среди...спокойно може да минете направо към Visual Studio 2008 Professional Edition или Visual Basic 2008 Express Edition...

Полезни връзки относно преминаването от стар проект писан на Visual Basic 6 към по новите среди:

<http://www.vbmigration.com>

http://www.artinsoft.com/pr_vbcompanion.aspx

<http://msdn.microsoft.com/en-us/library/aa480541.aspx>

<http://www.microsoft.com/downloads/details.aspx?FamilyId=10C491A2-FC67-4509-BC10-60C5C039A272&displaylang=en>

или може от сайта ми да си изтеглите книгата и инструмента на Майкрософт

<http://brigante.sytes.net/vb/VB Migrating%20Dec2005.zip>

Нарича се това преминаване - **Migration** или още code migration

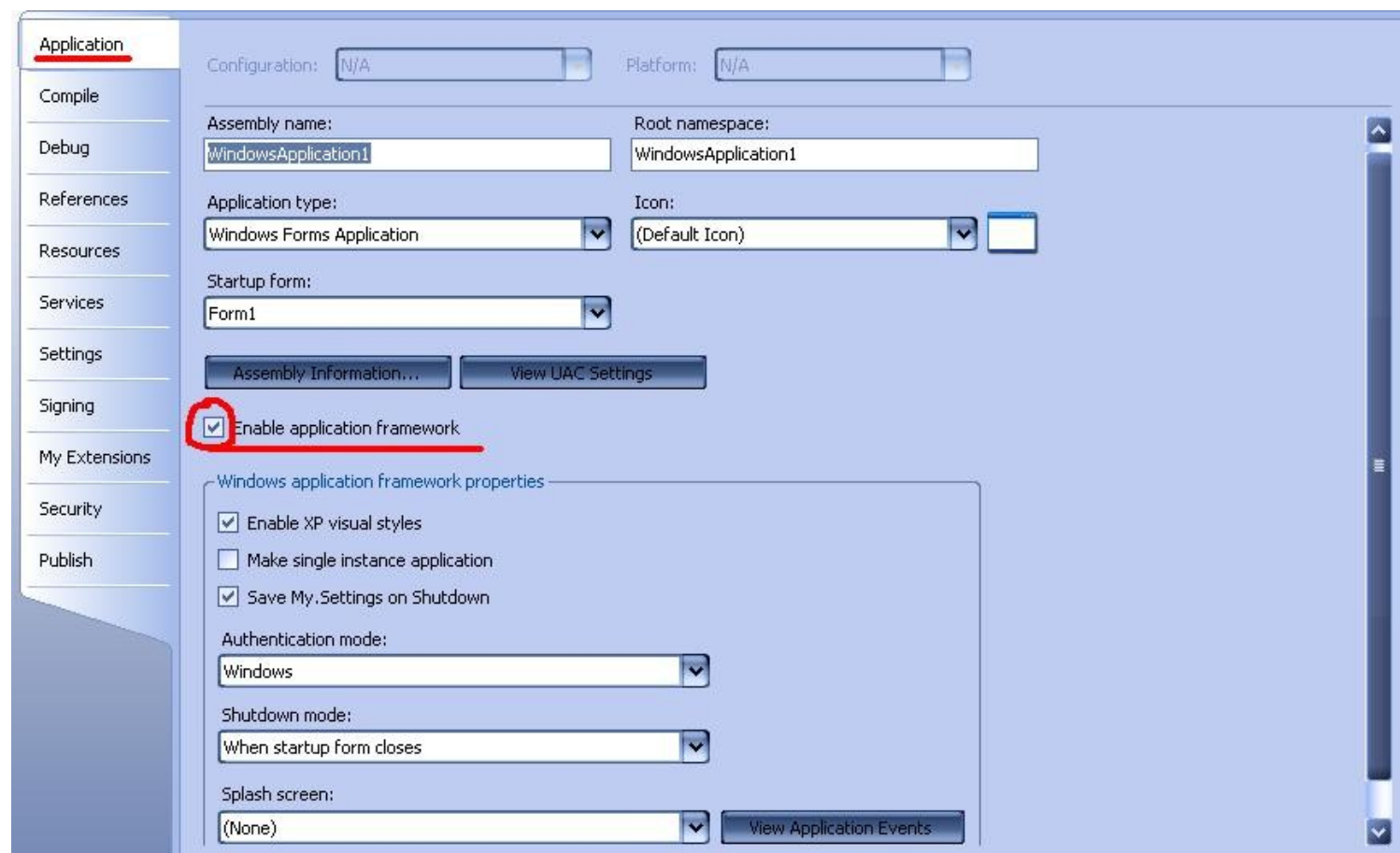
За надграждането или мигрирането от Visual Basic 6 към Visual Basic 2005 или 2008 можете да използвате инструмент, наречен Visual Basic Upgrade Wizard, който се стартира автоматично, когато отворите Visual Basic 6 програма в Visual Studio. Т.е осъществява се едно ъпгрейдване(надграждане) на програмен код.

Но този Upgrade Wizard на новата програмна среда не върши чудеса:) Принципно върши работа за някои контроли и форми, но при трансформирането на база от данни в стар проект ще си има доста грешки, които ще бъдат изведени и трудно ще дебъгнете:)

От Visual Basic 2005 към 2008 с лекота се конвертват проектите и грешките могат да са в разминаване на .net frameworks :) както и да е :)

Ако имате проект писан на 2005 и го отваряте с 2008 , и възникват грешки(може самият проект на чуждия автор да е бѣгнат:) пробвайте все пак да направите следното:

-отидете на Project от главното меню и след това на “името на проекта Properties” (най-отдолу е) кликнете и ще видите нещо подобно...



махнете отметката Eneble application framework
и опитайте отново да изградите проекта или само зеленката стрелкичка натиснете:)

Друг полезен линк относно преминаването от VB6 към VB .NET е:

<http://msdn.microsoft.com/en-us/vbasic/ms789185.aspx>

Visual Studio предлага нова начална страница или екран, който показва последно използваните проекти и Ви позволява да отваряте нови или съществуващи файлове, имате достъп до новини и статии свързани с програмирането.

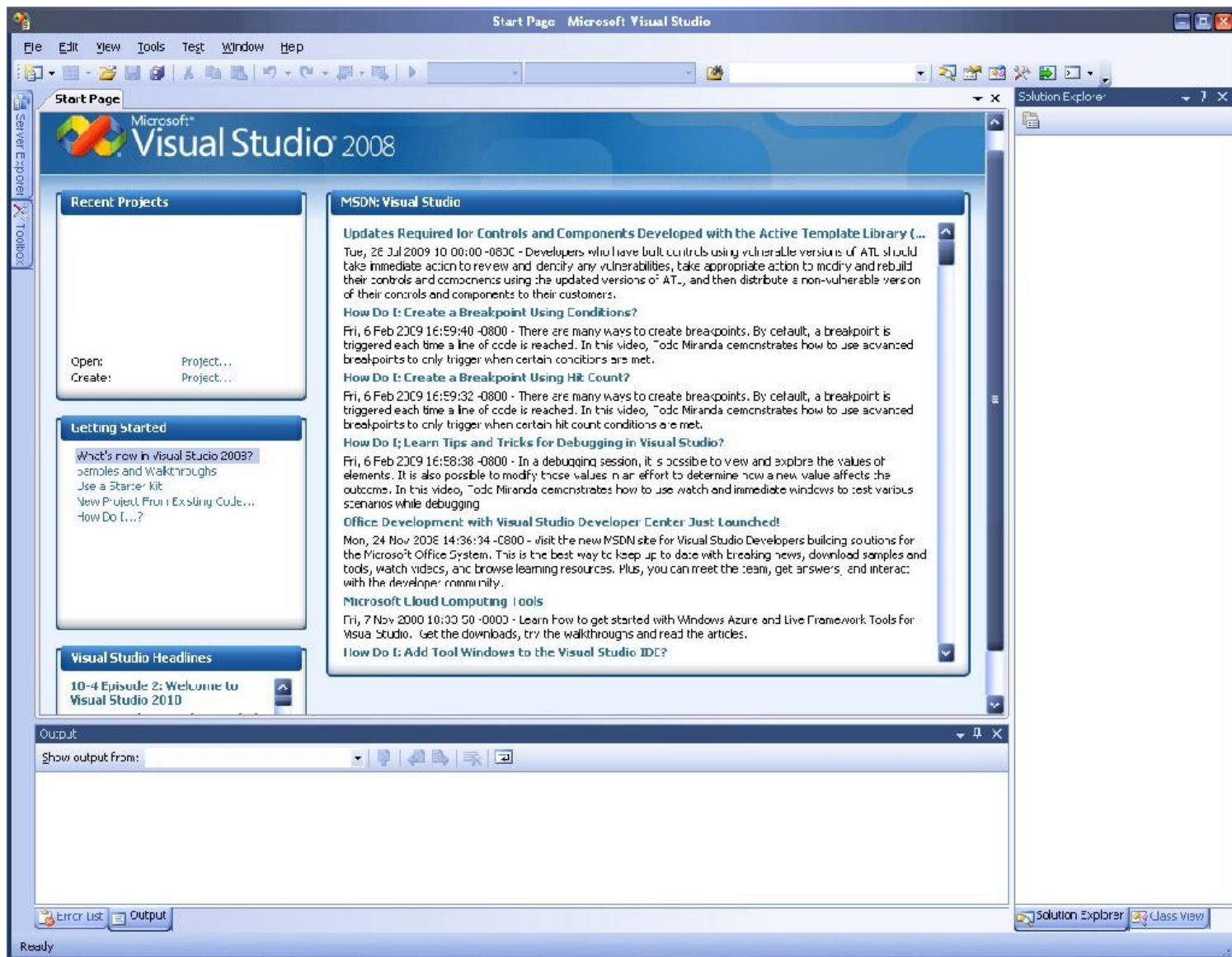
Забелязвате, че в стартовата страница си има браузър с rss msdn новинки:) кликайки там ще се отвори сайта и ще можете да четете най-новите неща...

Ако в главният прозорец, в който принципно се появява инфо като rss news не се появи msdn:news visual studio , то значи нямате нет или пък временно rss се обновява, или нещо се е бъргнало:) понеже често излиза съобщение, че няма налични новини (те са като линкове)

направете следното: идете на Tools → Options...

там на подраздел Startup и виждате линка на Майкрософт ... не го променяйте, а просто натиснете бутона OK :) би трябвало да се зареди емисията...

Ето как изглежда първоначално програмната среда Microsoft Visual Studio 2008 Professional Edition: ...тези програмни среди ги наричат integrated development environment (IDE)



И така да започнем с програмирането на Visual Basic с програмната среда Microsoft Visual Studio 2008 Professional Edition

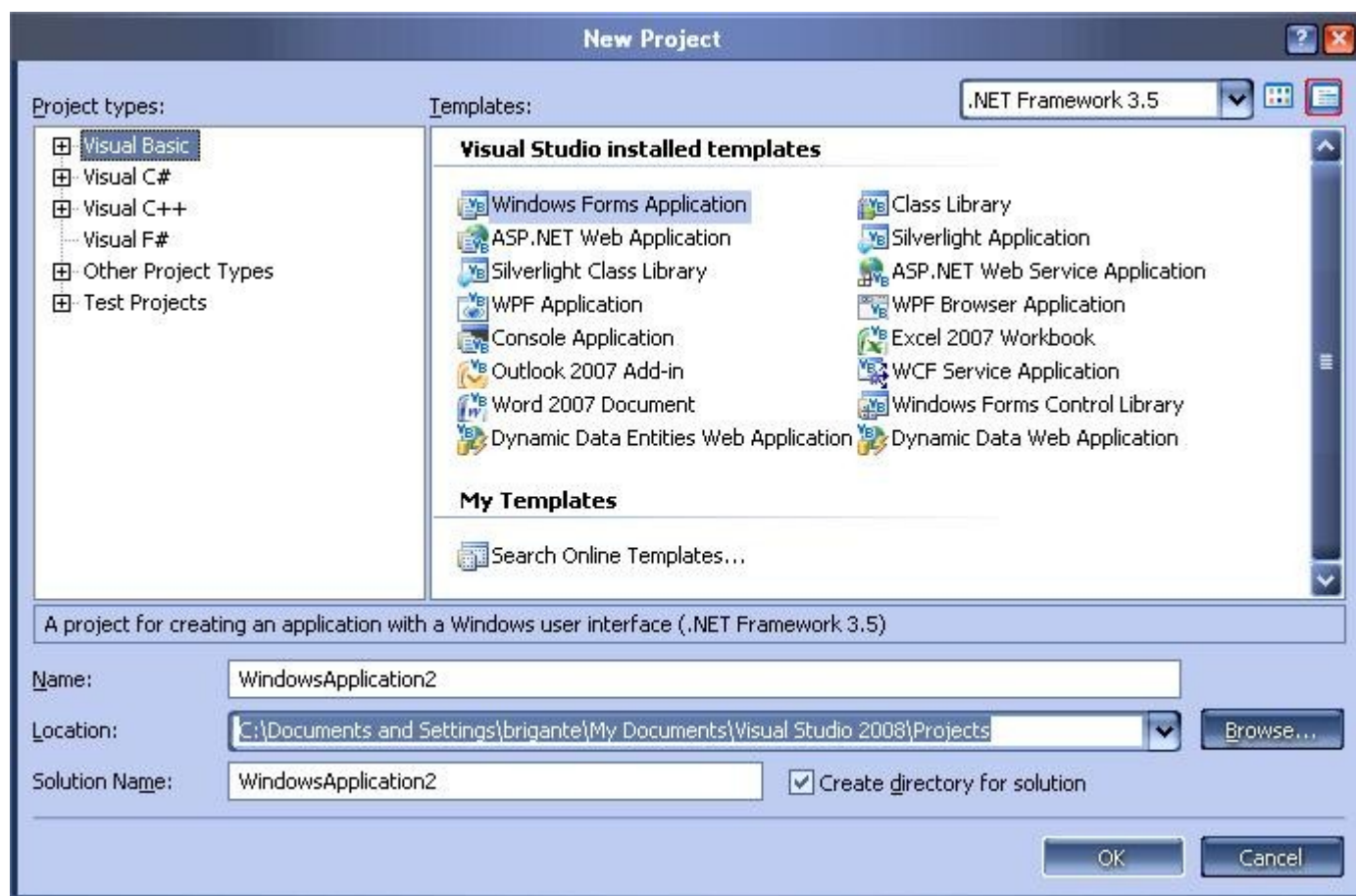
За да създадете нов проект на VB трябва от главното меню да кликнете на File и после да изберете New а от там ще изберете Project...

може също така и с клавишна комбинация: Ctrl+Shift+N

Новият проект може да бъде започнат и от полето Recent Projects още както е старт страницата на програмната среда. То се намира горе в левият край. Първоначално там няма да има никакви проекти:)

Open: Project
Create: Project

кликате на **Project** с/у **Create(Създай)** и ще се зареди диалогов прозорец **New Project**, който изглежда ето така:



Виждате полето с видовете възможни програмни езици...

Visual Basic

Visual C#

Visual C++

Visual F# този компилатор се добавя по-късно (за повече инфо посетете сайта ми

<http://brigante.sytes.net/fsharp.aspx>)

Вие може да нямате също така в програмната си среда и **Silverlight Application** (може да прочетете също така в сайта ми за силвърлайта:) <http://brigante.sytes.net/silverlight.aspx>

Създадените приложения се съхраняват в:

C:\Documents and Settings\името ви\My Documents\Visual Studio 2008\Projects\

Когато създадете проект и после искате да го унищожите,то просто го изтрийте от директорията...

C:\Documents and Settings\името\My Documents\Visual Studio 2008\Projects

и после няма да го има в Текущи проекти...

Ако имате някакъв чужд проект писан на VB и искате да го отворите направете следното:

От главното меню отидете на **File → Open → Project/Solution**

ще се зареди диалогов прозорец **Open Project** и си намерете файловете:)

търсете файла **.slt** (solution=решение) това е файла с цялостния проект и кликайки на него ще се зареди абсолютно всичко от проекта :)

може и да отворите даден проект писан на vb като кликнете само на **VB Project files**

.vbproj или **.vbpr**

това са разширенията (файловите формати) на vb приложенията...

Във Visual Studio програмите в процес на разработка са наречени проекти(Projects) или решения(Solutions), защото те съдържат много отделните компоненти, а не само един файл.

Solution Explorer може отначало да е в левия край,но за удобство може да се мести:)просто го хванете и завлачвайте и ще видите едни стрелки...поставяте го на част от стрелките...

В ляво горе има разделче наречено Recent Projects (текущи проекти=създадени вече проекти, които ще се заредят при кликане)

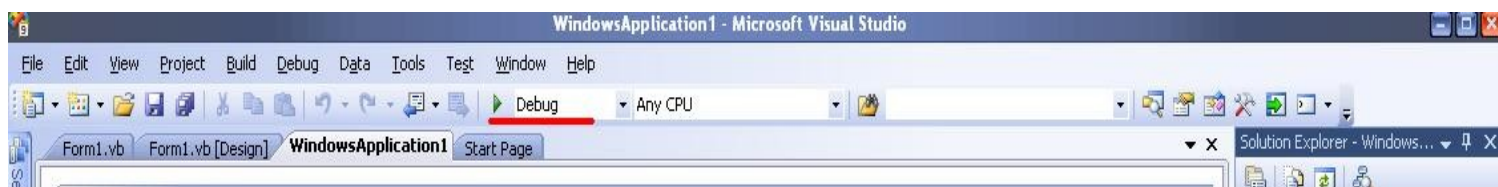
Под него се намира раздела Getting Started: там са част от помощните уроци от сайта на Майкрософт

Вече може би сте забелязали,че в тази нова програмна среда вместо така наречените Controls(контроли,инструменти) имаме така нареченото Toolbox (кутийка с инструменти:)
Оттам ще можете да си разполагате бутоните и др. контроли...

Можете да си нагласите Toolbox както Ви е удобно...най-добре е да е на AutoHide (автоматично да се скрива и при допир с мишката да се показват всички(е не всички) контроли:)

Поставянето на контролите става просто като кликнете веднъж в/у тях и после отидете с мишката до работната форма и кликнете там, и така контрола вече е добавен...

Ако се опитате да компилирате, да изградите приложението това става в режим Debug (създава се и такава директория в проекта Ви) Обикновено програмката ще е с по-голям размер, отколкото в крайния режим Release ... а той се задава от главното меню:



..там където по подразбиране е Debug отметнете на Release и създаденият изпълнителен файл ще е с по-малък размер...

Добра практика е постоянно като правите промени по проекта си да си го сейвате(съхранявате) и да си създавате копия на проектите:)

Построяване на изпълним файл на проекта: За да създадете изпълним файл на проекта, използвайте команда Build (Построй) / Build име_на_проекта...

Дебъгване на програмата: Дебъгерът се стартира с команда от менюто Debug /Start

Управление на прозорците в IDE

Един от основополагащите принципи на интегрираните среди за разработка се състои в това, че менютата, лентите с инструменти и прозорците на работната област (областта за редактиране, прозорецът на проекта, прозорецът за съобщения) са „вградени“ - т.е. се намират в рамките на главния прозорец на Visual Studio. NET Ако се скрият част от лентите с инструменти, на екрана се освобождава допълнително място, което може да се използва за прозорците за редактиране на файловете с изходния код на проекта...

По подразбиране прозорците за редактиране заемат цялата свободна работна област в главния прозорец.

Препоръки за управление на прозорците в IDE :

Работните параметри и показваните в прозореца на IDE компоненти се настройват от страниците на групата Environment от диалоговия прозорец Options, който се показва с команда Tools /Options

С командите от меню View се показват различните прозорци (прозорецът за ресурси, прозорецът за съобщения, прозорецът със списъка от задачи и т.н.)...

Всички отворени прозорци (припокриващи се страници) за редактиране са включени в списъка от меню Window(Прозорец)

Ако изберете някой от тях в списъка, ще го покажете на по-преден план от всички останали прозорци. Същото ще се случи и ако щракнете върху етикета на страницата над областта за редактиране.

Команда Window /Split (Раздели) разделя активния прозорец на подпрозорци. Това дава възможност да се преглеждат едновременно няколко фрагмента от файла, без да се увеличава броят на отворените прозорци за редактиране...

Разделителната линия в прозореца за редактиране може да се премества с мишката. Ако искате да затворите някой от подпрозорците, придвижете разделителната му линия до разположената по-горе от нея...

Система от менюта:

Ще опиша само най-често използваните от тях...

На кратко ще Ви запозная с главното меню в програмната среда Microsoft Visual Studio 2008

Има малки разлики от Visual Studio 2005

Меню File (Файл) се използва за общо управление на файлове и на областта на проекта
->

New (Нов) тази команда показва подменю с команди, с които се отварят диалогови прозорци за създаване на нови файлове, проекти и решения. Оттук се стартират различните помощници за разработка на проекти

Open (Отвори) командата показва подменю, което служи за отваряне на различни файлове, файловете, отворени с тази команда, не се включват в текущия проект...отваря и уеб проекти

Add New Item (Добави нов компонент) тази команда създава нов компонент от избрания тип и го добавя към проекта

Add Existing Item (Добави съществуващ компонент) тази команда добавя към проекта нов (вече съществуващ) компонент от указания файл

Add Project (Добави проект) командата създава нов проект от избрания тип в рамките на дадено решение, или добавя в дадено решение вече съществуващ, проект

Така беше в по-старата Visual Studio .NET , а сега в новата 2008 има само Add и подменю: New Project...

New Web Site...

Existing Project...

Existing Web Site...

т.е. добавя нов проект

нов уеб сайт

вече съществуващ проект

добавя вече съществуващ уебсайт

Close (Затвори) командата затваря активния прозорец за редактиране

Open Solution (Отвори решение, цялостен проект)

Open Project/Solution (Отвори проект/решение)

Close Solution (Затвори решение) съхранява включените в решението проекти...

Save (Съхрани)

Save as (Съхрани като)

Save all (Съхрани Всичко)

Export Template (експортиране, изнасяне на готов вече шаблон)

Page Setup (Структура на страница) с тази команда се задават параметрите на страницата при печат на файловете с изходния код

Print (Печат) командата разпечатва файла, който се намира в активния прозорец за редактиране

Recent Files (Наскоро отворени файлове) показва списък с последните отворени файлове.

Дължината на списъка се задава на страница Environment/General (Работна област/Общи) на диалоговия прозорец Options, който се отваря с команда Tools /Options

Recent Projects (Наскоро отворени проекти) показва списък с последните отворени проекти и решения... Дължината на списъка се задава на страница Environment/ General (Работна област/Общи) от диалоговия прозорец Options, който се отваря с команда Tools /Options

Exit (Изход)

Менюто Edit (Редактиране)

Меню Edit (Редактиране) съдържа команди за редактиране, търсене и замяна на текст в активния прозорец за редактиране

Undo (Отмени) ако се свъркали нещо(изтрили) в кода се поправете с тази команда...

Redo (Върни)

Cut (Изрежи)

Copy (Копирай)

Paste (Вмъкни)

Delete (Изтрий)

Select all (Избери всичко)

Find and replace (Намери и замени) подменю, което позволява да се търси и заменя зададен текстов фрагмент в прозореца за редактиране или в указано подмножество от файлове

Go To (Премини) командата извършва преход към зададен адрес, определение, референция или ред

Bookmarks (позиции за преход) с тази команда се създават специални позиции за преход, към които след това може да се премине с команда Go To (Премини)

Меню View (Изглед)

По-голямата част от командите в меню View показват различните прозорци за преглеждане на информация...

Code (Показва редакторът, където се пише програмния код)

Designer (Показва дизайн прозореца на Вашето приложение, формата)

Solution Explorer (Преглед на решенията) отваря прозореца за преглед на решенията в областта на проекта

Class View отваря прозореца за преглед на класовете от проекта в областта на проекта

Resource View (Показване на Ресурсите от проекта)

Тази команда отваря прозорец със същото име в областта на проекта, в който може да се прегледат и редактират ресурсите, използвани в текущия проект...и сорс от други чужди проекти...

Properties Menager или Window (Параметри) командата отваря прозореца, в който се показват параметрите на избрания компонент на проекта...показват се своите свойства...

Toolbox (Палитра с инструменти/контроли)

От Other Windows (Други прозорци) към Web Browser (Браузър) се отваря подменю с команди за управление на браузъра (за по-удобен достъп до справочната система)

Tasks List (Задачи) избор на информацията, която ще се показва в областта за съобщения

Toolbars (ленти с инструменти) за избор на показваните ленти с инструменти

Full Screen (Пълен екран) командата превключва текущия прозорец за редактиране в пълноекранен режим. С натискане на клавиша [Esc] можете да се върнете в нормален режим...

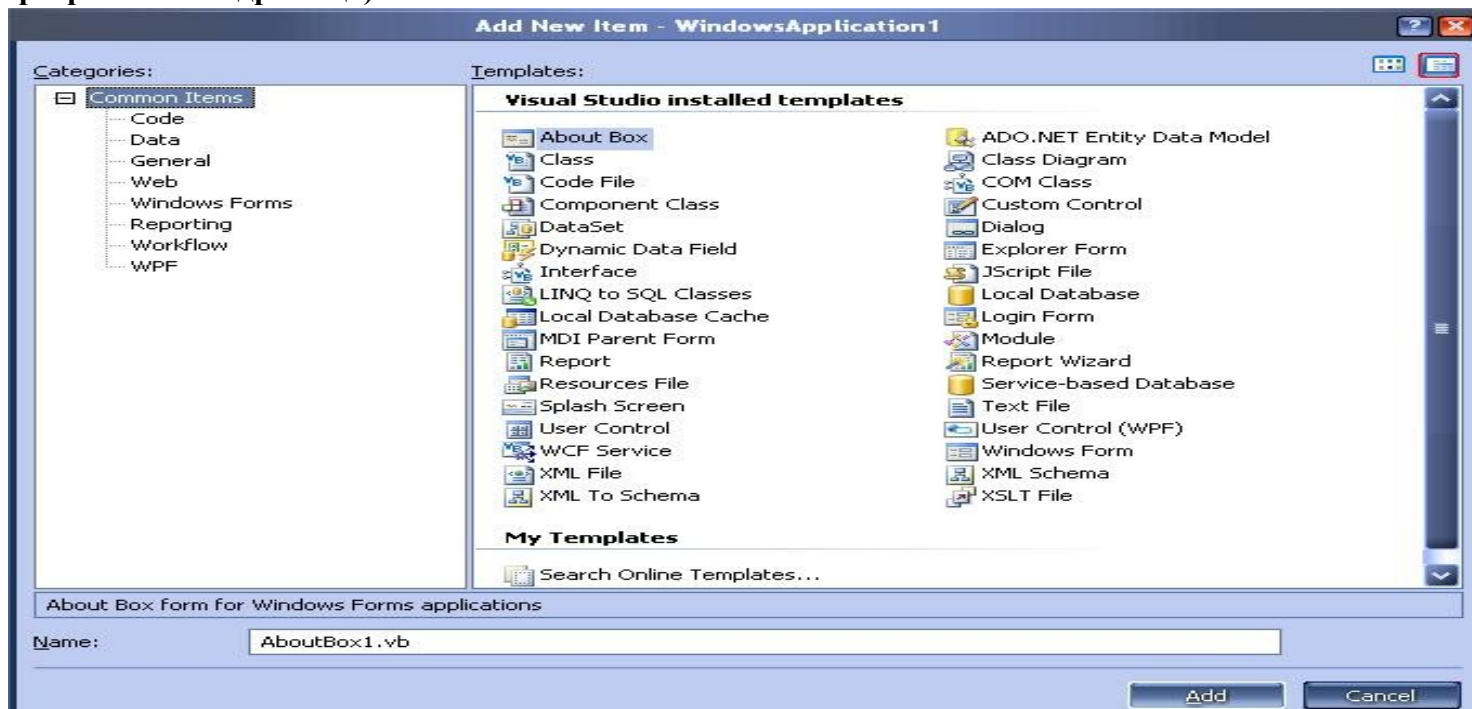
Следва двойка команди за навигация (преход към предишното и следващото съдържание на областта за редактиране)

Property Pages (Параметри на страницата) командата показва на екрана диалогов прозорец с информация за проекта, който е избран от областта за проекта

Меню Project (Проект)

Меню Project съдържа команди за създаване и управление на проекта.

Add Windows Form... (Добавя нови форми към проекта, нов клас, сплаш скрийн(при зареждане на даден прозорец или на главното приложение да се зарежда някаква картинка с текст на нея и прогрес лента и др. неща)



Add Class (Създай, добави клас) тази команда показва на екрана диалогов прозорец, от който може да се зададе нов клас, като за него се укажат базови класове и файлове с изходен код

Add Resource (Добави ресурс)

Add New Item (Добави нов компонент) с тази команда се създава и включва в проекта нов компонент от указания тип

Add Existing Item (Добави съществуващ компонент) командата добавя към проекта нов компонент от избрания файл

Add Reference (Добави референция)

Set As Startup Project (Направи начален проект по подразбиране)

име на проекта **Properties (Параметри)** командата показва на екрана диалогов прозорец, от който може да настроите параметрите на проекта...може например да зададете настройки за компилатора, свързващия редактор и дебъгера, да укажете параметрите на командния ред и т.н.

Меню Build (Построяване)

Меню Build съдържа функции за разработка на програмата, както и команди за стартиране на вградения дебъгер

Build solution (Построй решение) командата построява всички проекти, включени в решението. Ако след създаването на изпълнимия модул в някой от файловете с изходен код на проекта са били внесени изменения, поправеният изходен файл ще бъде компилиран отново. Изпълнимият модул също ще бъде свързан отново.

Rebuild Solution (Препострой решение) командата първо компилира всички изходни файлове от всички проекти от решението, след което свързва отново и изпълнимите модули на проектите.

Clean Solution (Почисти решение) командата премахва всички междинни и резултантни файлове в проектите на решението.

Build името_на_изпълним_файл (Построй името_на_изпълним_файл)... тази команда построява указания изпълним файл (на проект)...ако след създаването на изпълнимия модул в някой от файловете с изходен код на проекта са били внесени промени, поправеният изходен файл ще бъде компилиран отново, а изпълнимият модул - съответно свързан още веднъж.

Rebuild името_на_изпълним_файл (Препострой проекта) командата компилира всички файлове с изходен код в проекта, след което отново свързва и изпълнимия модул на проекта.

Clean името_на_изпълним_файл (Почисти проекта) тази команда премахва всички междинни и резултантни файлове за проекта.

Configuration Manager (Конфигурации) командата служи за конфигуриране настройките на проекта, които ще се използват при разработка на програми. С нея също може да се зададе и нова конфигурация за текущия проект. Настройките може да се променят в диалоговия прозорец име на проекта **Property Pages**, който се показва с команда **Project / Properties**

Меню Tools (Инструменти)

С командите от меню **Tools** се викат от **IDE** различни спомагателни програми, а също така може да се настройва интегрираната среда за разработка(програманата среда)

Connect to Database (Свържи се с база данни)

Customize тази команда показва диалогов прозорец със същото име, от който може да си организирате менютата, лентите с инструменти и клавиатурата по начин, който е най-удобен
Options командата отваря диалогов прозорец със същото име, от който може да се настройва интегрираната среда за разработка

Меню Debug

С командите от меню **Debug** се извършва проверка и оптимизация на програмите от проекта ... няма да обяснявам работата с дебъгера:)

Меню Help (Справочна система)

В меню **Help (Справочна система)** са включени командите на библиотеката **MSDN**

Може да получавате помощ онлайн, но може и локално да разглеждате помощните файлове(аз препоръчвам да отмените на онлайн)

Отделете време и разгледайте!

Меню Window(Прозорец)

Меню Window включва функции за управление на прозорците

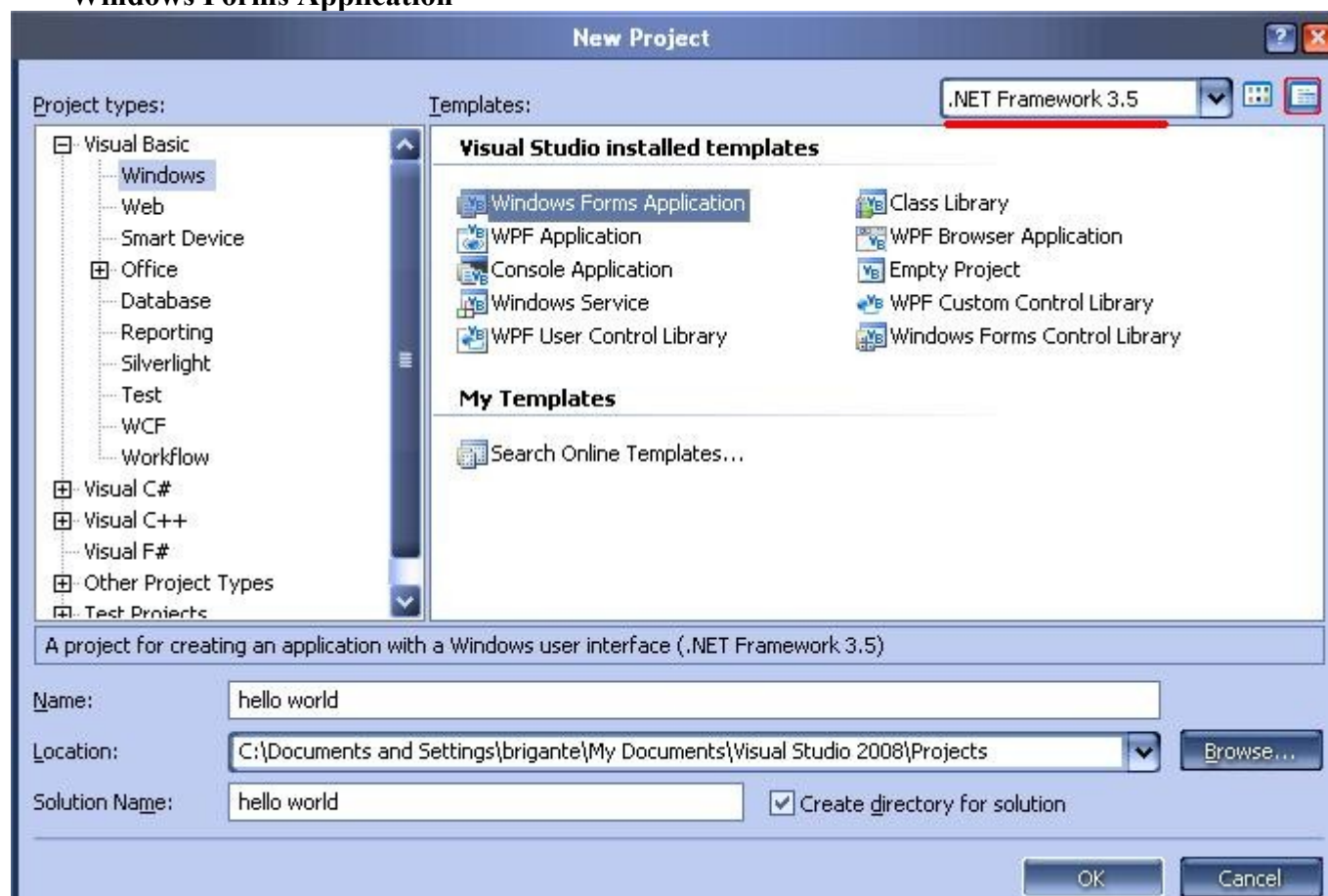
Ако примерно искате приложението да има леко прозрачен изглед (Opacity) може да направите това ето така: оразмерете(модефицирайте) си прозореца както искате като издърпвате долу краищата му...след това отидете на Properties(а ако не е видимо в програмната среда идете към главното меню и от View отидете на Other Windows и в подменюто намерете Properties Window и ще се покаже...добра практика е винаги да е показано в IDE) ... в Properties намерете къде е Opacity и намалете стойността (тя е на 100%)

сложете примерно 50-70 % ... компилирайте и забележете разликата в приложението си...

Създаване на приложение на VB в програмната среда Visual Studio 2008 Professional Edition

За да създадете проект първо стартирайте програмната среда:) след това от File изберете New → Project...

от показаният се диалогов прозорец естествено отмятаме на Visual Basic и избираме Windows Forms Application



името на проекта ще бъде озаглавен по подразбиране “WindowsApplication1”

можете да си промените името като го озаглавите както желаете и може да е и на български ез.

При мен е озаглавено hello world

Както виждате приложението ни работи под .NET Framework 3.5

Може да си работите и на другите версии:)

Кликнете на бутона OK и ще се зареди прокта в програмната среда.

Следващата стъпка е да си изградим потребителския интерфейс на приложението.

Това става като си оразмерим работния прозорец, издърпвайки го с мишката и си оразмеряваме прозорчето както искаме...също така добавяме и необходимите контроли (controls)

Тях ги вземаме от Toolbox

Контролите се добавят към работното поле(работен прозорец,диалогов прозорец) като кликнете един път върху избрания контрол и после в самото работно поле или два пъти в/у контрола и той ще се появи в работния прозорец.След като се е появил си го оразмерявате като с мишката го издърпвате по краищата и си го разполагате където искате:)

Първата ни програмка ще се казва “Hello World”
ще ползваме MSGBOX

Добавете си едно бутонче към приложението. Това става като от Toolbox намерите контрола Button
Кликнете два пъти в/у него или го засурнете към работния прозорец.

След това си го преместете където желаете. Кликнете два пъти в/у бутона, който току що поставихте.
Ще се зареди, активира програмният редактор.

Активирането на текстовия програмен редактор, за да можем да въвеждаме код в програмна среда Visual Studio 2008 и другите среди, става като се кликне два пъти в/у самата форма(работното ни поле,работен прозорец) или още може да се активира като кликнете 2 пъти в/у някой от контролите, които сте разположили!!!

(в код едитора на Visual Studio може да се въведат повече от 65000 символа)

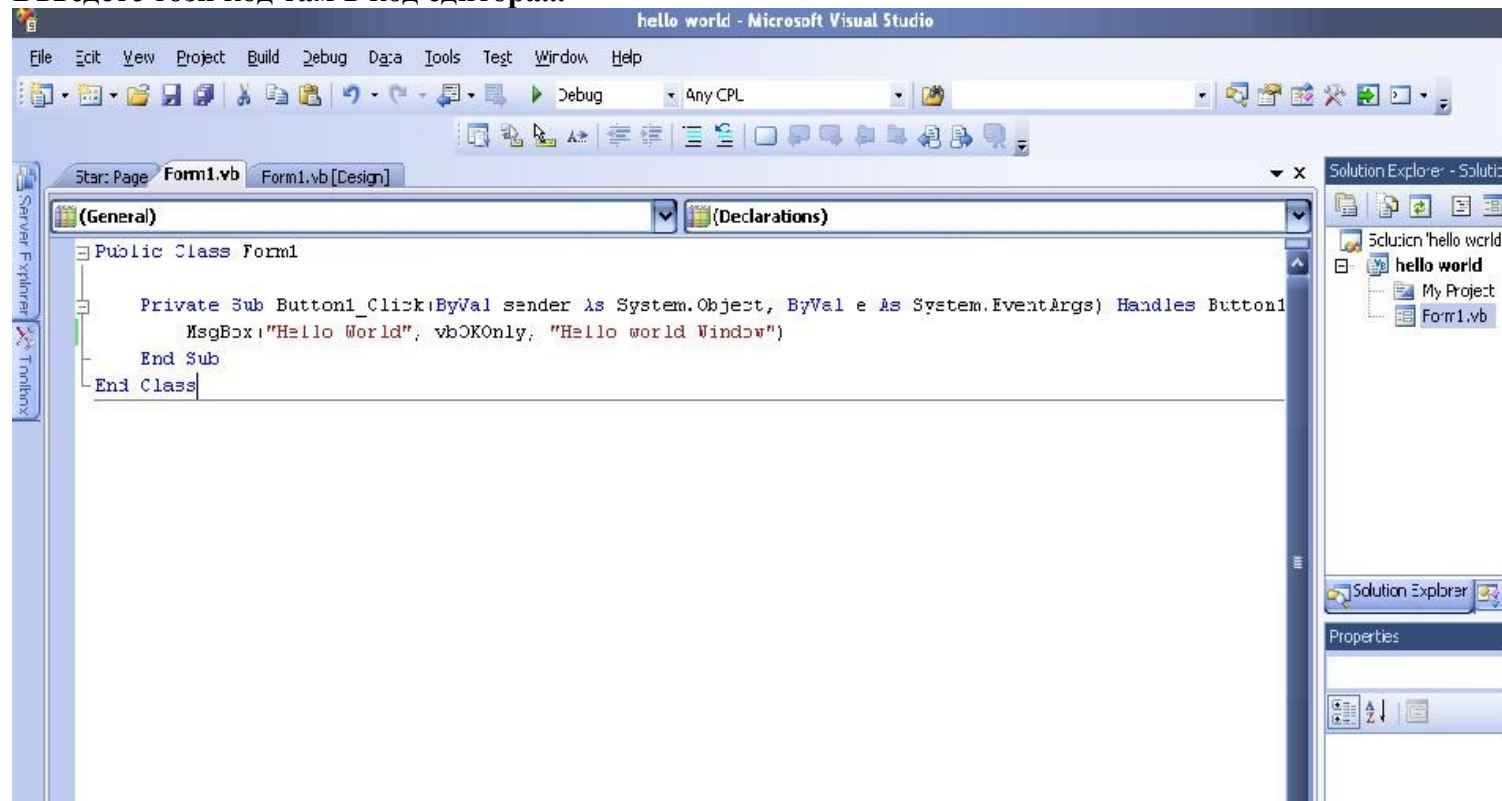
Ще забележите,че програмната среда вече е генерирала доста програмен код:) и на нас единствено ни остава да си добавим код за самите контроли...в случая за контролът бутон

Приложението ни има само един контрол:)

Програмната логика за контрола бутон (Button) е следната:

```
MsgBox("Hello World", vbOKOnly, "Hello world Window")
```

Въведете този код там в код едитора...



При мен изглежда ето така...

За да преминете от код едитора към дизайн изгледа просто кликнете на таба Form.vb [Design]

Сега си съхранете проекта(сейвайте си го) Това става като натиснете бутончето с дискетките...Правете това често при по-големи проекти!

Остава да тестваме нашата първа програмка:) Натиснете бутона с малката зелена стрелкичка.
Start Debugging



Би трябвало приложението да изглежда така, когато го дебъгваме...

Кликайки на бутона, ще се появи и малкият диалогов прозорец с надписа Hello World

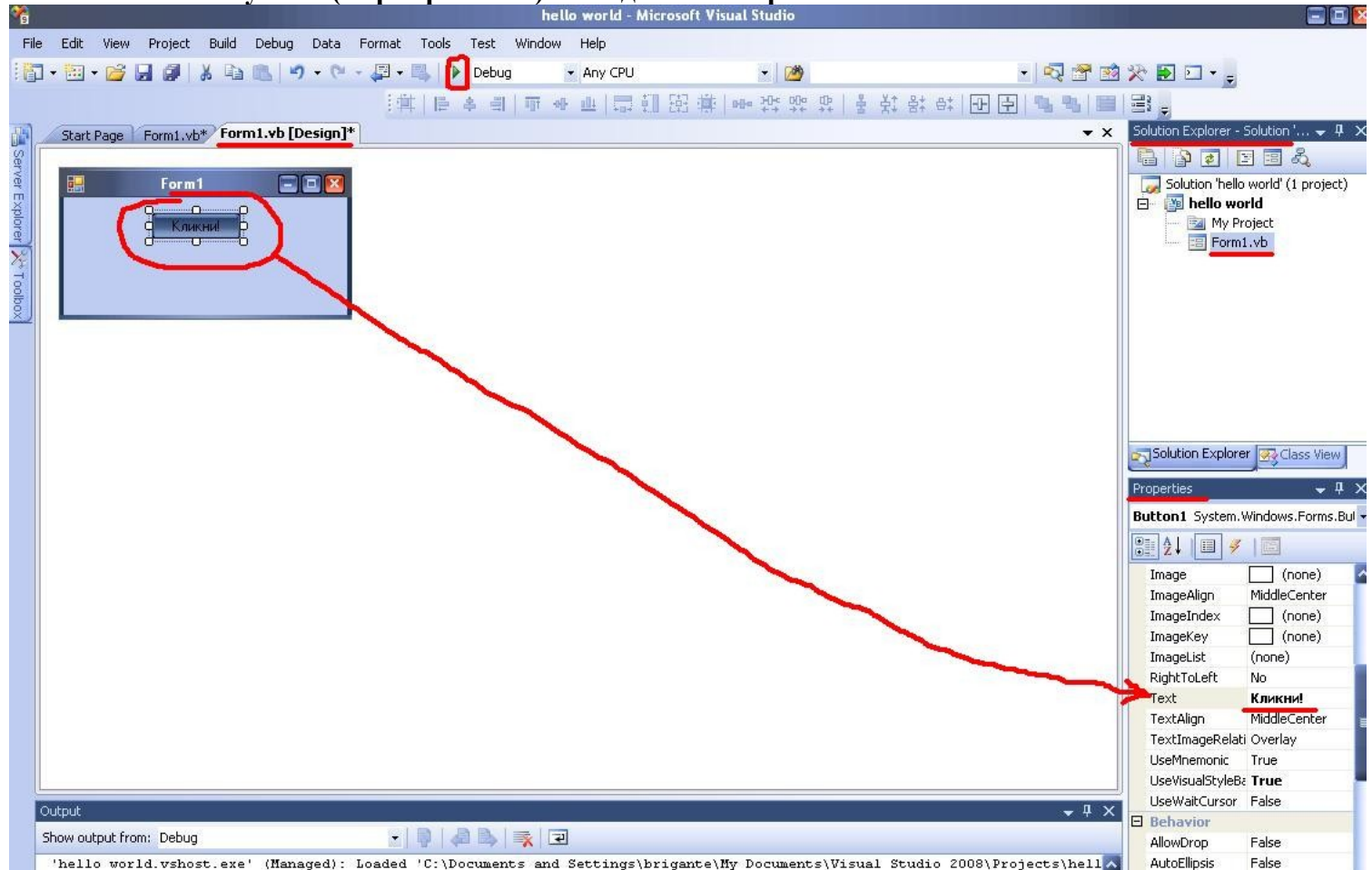
Сега да направим някои подобрения по нашият проект:)

Минете пак в работен режим в програмната среда...

Ще променим надписа на бутона:

По подразбиране той е озаглавен Button1

Кликнете 1 път в/у него (маркирайте го) и отидете на Properties



Аз съм си го озаглавил “Кликни!”

Просто промених текста (Text) от св-вата (Properties)

Съхранете проекта и наново дебъгвайте...

Остава да изградите приложението си:) от Build -> Build Solution

C:\Documents and Settings\brigante\My Documents\Visual Studio 2008\Projects\hello world\hello world\bin\Debug\hello world.exe

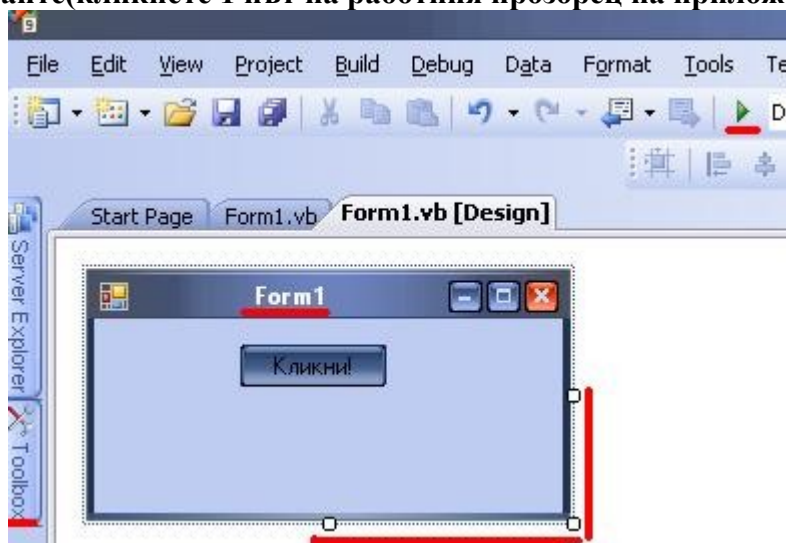
...и аз в Output-а получих инфо, че проекта ми е изграден правилно...

Ако раздела в програмната среда Output не го виждате, то отидете на главното меню и от View изберете Output (Ctrl+Alt+O)

А сега да понаправим още малко промени в нашето приложение...

Ще сменим името на прозорчето т.е. вече няма да е Form1 ,а ще сложим наше си име:)

За целта просто маркирайте(кликнете 1 път на работния прозорец на приложението)

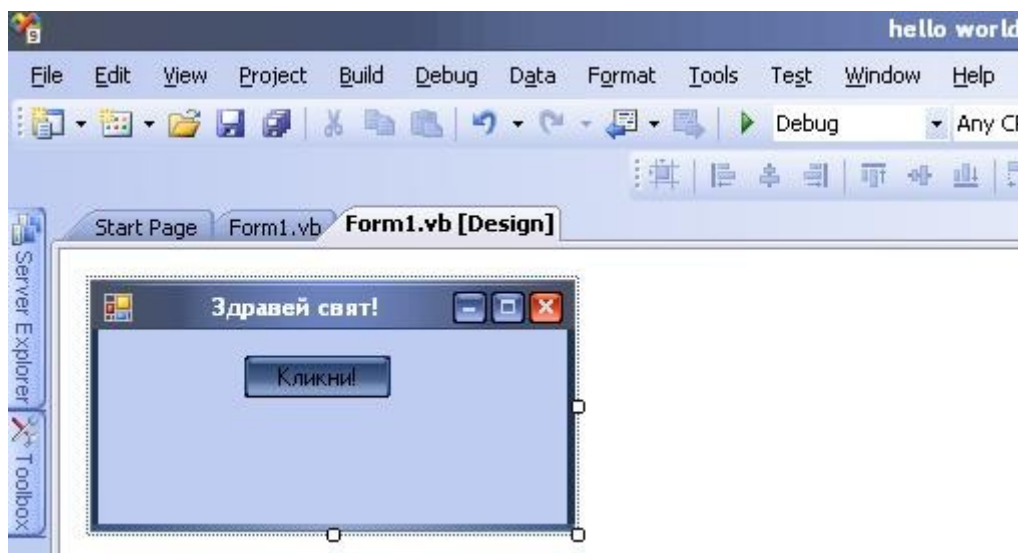


...и ще видите тези 3 малки бели “квадратчета” около рамката...това означава, че вече сме маркирали целият работен прозорец на приложението ни...и вече можем да му променяме св-вата от Properties ... също така можем да си го оразмерим както желаем(да го направим по-голям, по-малък)

Сега от Properties(Своиства) намерете Text и променете името Form1 може да напишете каквото си искате на бълг. ез.

След като го напишете нещо там отидете на формата и кликнете един път в/у нея(работният прозорец на приложението) и ще забележите как името се променя:)

Не забравяйте да си съхранявате проекта!



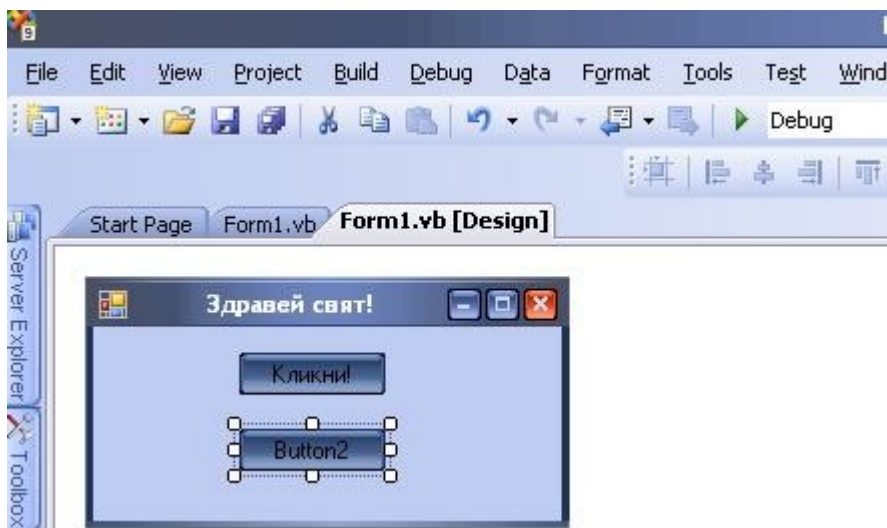
...ето при мен вече е променено заглавието на формата...

А сега да поставим и втори контрол в нашето приложение(програмка)

Пак ще е контрол бутон (Button) , но той ще ни е за Изход в приложението ни т.е. ще си го именуваме “Изход” и при натискането му, потребителят ще изключва програмата...вместо да натиска на малкото червено бутонче...

От Toolbox си издърпайте бутонче:) после си го поставете от долу под бутона “Клики!” т.е. правим си нов потребителски интерфейс на приложението...

При мен приложението с новите промени изглежда ето така:



Сега, за да го активирате този нов контрол, кликнете два пъти в/у него и ще се отвори код едитора...

Там ще напишем програмната логика за контрола Button2

И понеже този бутон ще ни служи за изход от програмата, то ние ще променим и името му...

От Properties отидете на Text и изтрийте надписа Button2 и си напишете "Изход"

След това кликнете където и да е по работният прозорец(формата) на приложението...

Сега остава да въведем функцията Close() в код едитора(редактора) за този контрол Button2

...би трябвало целият сорс код да изглежда ето така:

Public Class Form1

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    MsgBox("Hello World", vbOKOnly, "Hello world Window")
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Close()
End Sub
End Class
```

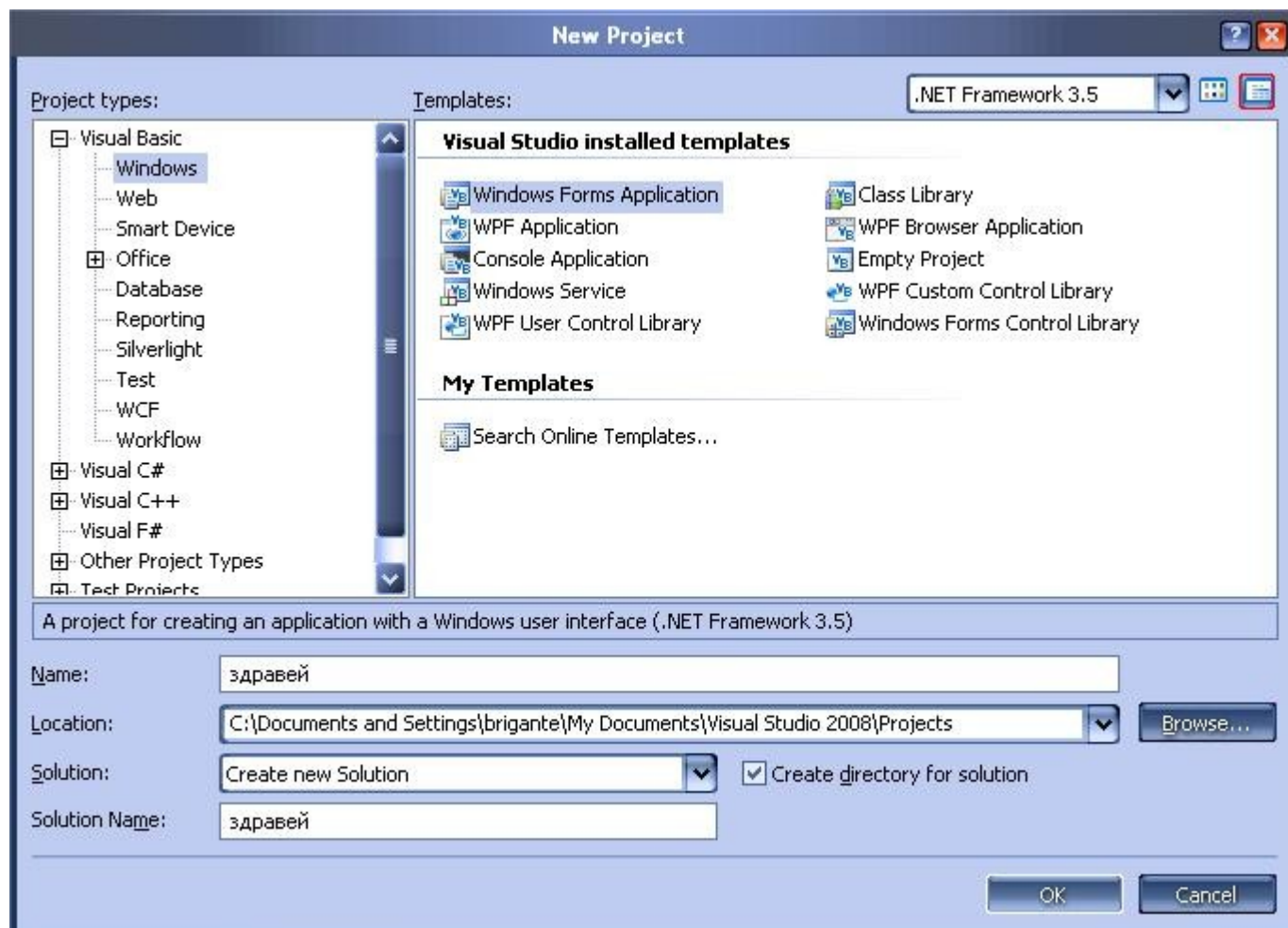
Съхранете си проекта и го тествайте! След това го изградете пак: Build->Build Solution



Ще направим нова програмка, която също е Hello World , но ще ползваме още един контрол. Малко по-различна ще е.
Просто ще Ви покажа и друг начин ...

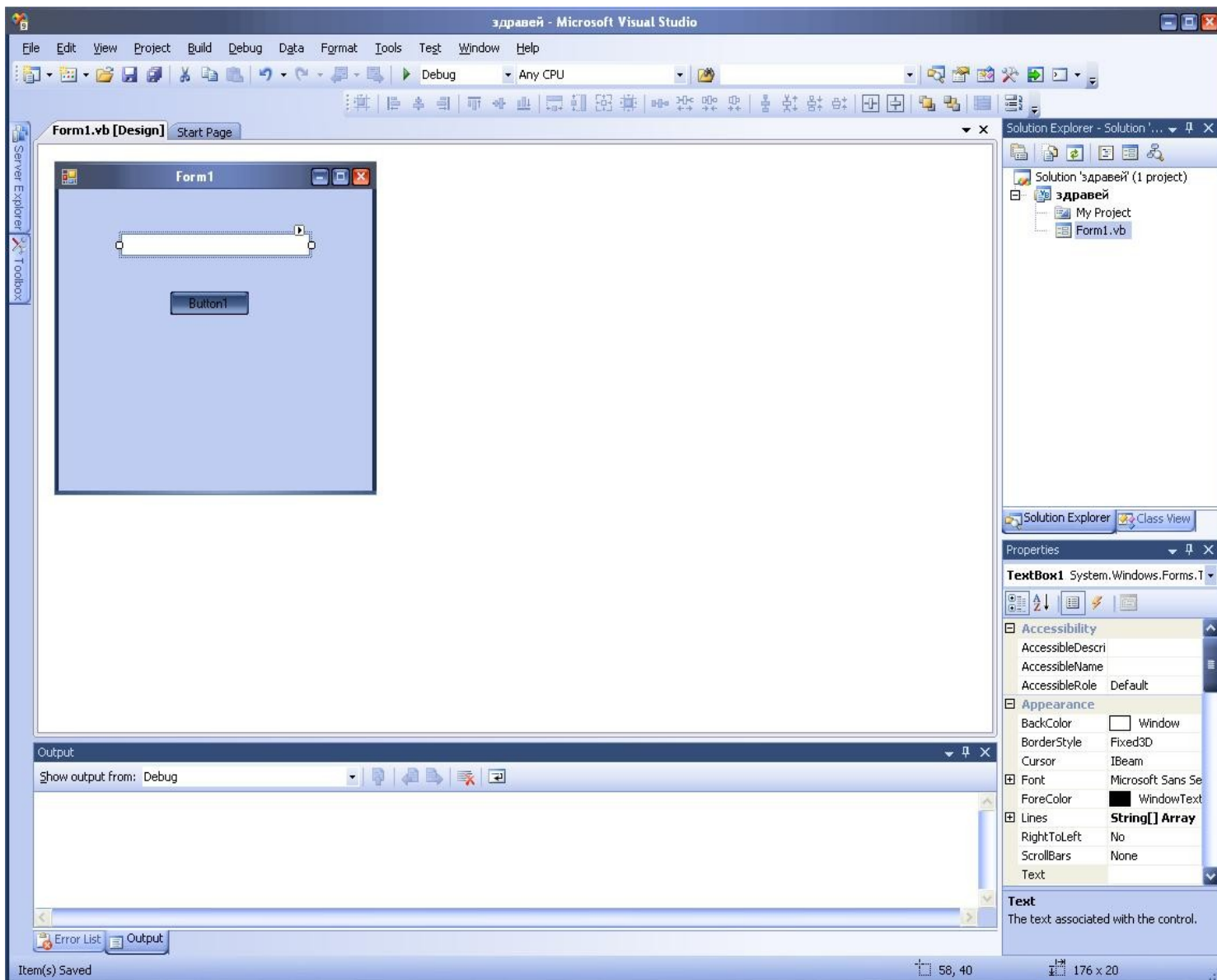
Трябва да знаете, че в информатиката всяко нещо може да се прави по много и различни начини.

Отидете в главното меню на File -> New -> Project
и си създайте нов проект...



наименовал съм си проекта “здравей”
кликате на бутона ОК и ще се зареди новият проект:

И започваме изграждането на потребителският интерфейс (User Interface) на приложението (Application) ни:



Добавяте си 2 контрола:

първият е TextBox от Toolbox се взема:)

вторият е Button пак от Toolbox се взема:)

и си ги разполагате както искате(правите потребителския интерфейс на приложението си)

Не забравяйте да ги активирате тези 2 контрола... това ставаше като кликате в/у тях...и се отваря код едатора...а за да отидете отново в дизайн изгледа си кликате на таба Form1.vb [Design] и сте отново в работната си форма ...

А сега да променим името на контрола бутон (Button)

От Properties намерете Text и си го озаглавете примерно “Клигни тук!”

Можем също така да направим и доста промени по контрола TextBox

За целта маркирайте самият контрол TextBox

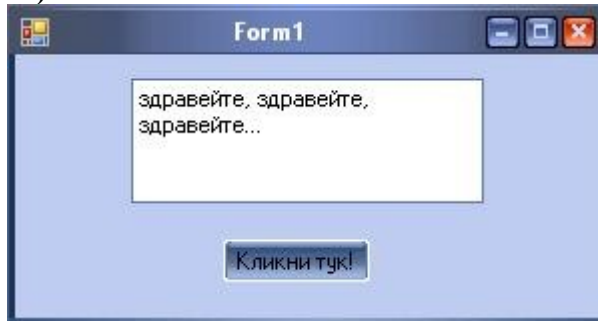
Сигурно забелязвате как той има една малка стрелкичка...

От там се променя възможността за многоредово изписване на някакъв текст (низ=string)

Ето как съм го променил:



или по този начин с бълг. текст :)



А сега може да се опитате да промените името на приложението (да не е Form1)

Опитайте се да добавите и нов контрол бутон, който ще служи за затваряне на приложението...

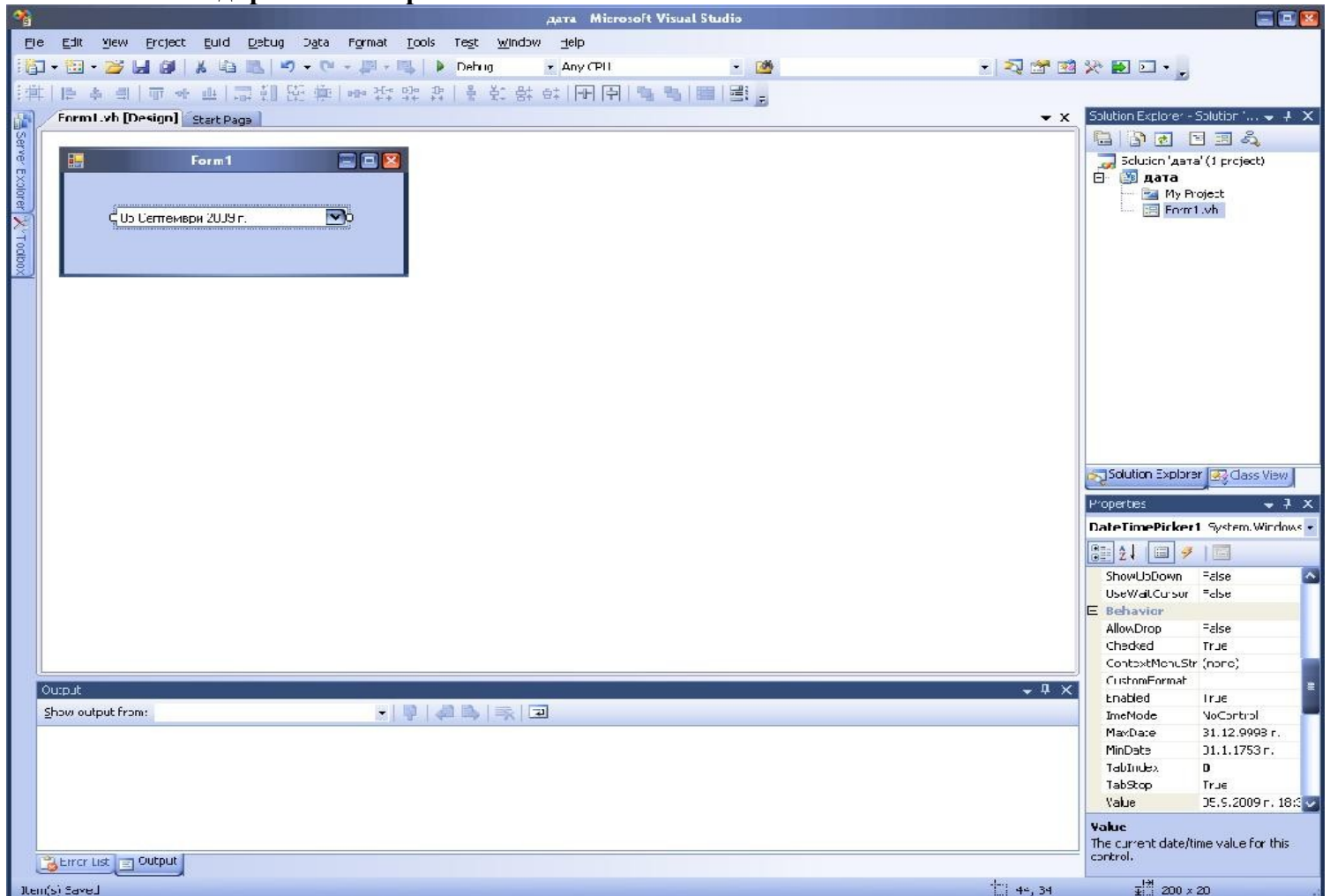
Сега ще пореботим и с друг контрол и това е контролът DateTimePicker

Отидете на File и кликнете на Close Solution...

...ще се затвори настоящият проект и ще се зареди пак началната страница,стартовата(Start Page), както е когато стартираме програмната среда...и си създайте нов проект...

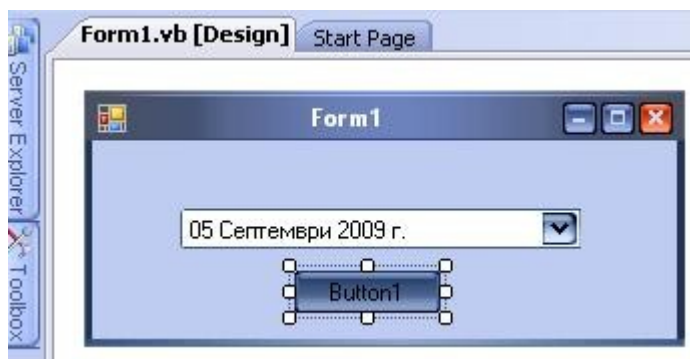
Може да си го именувате "DateTimePicker" или "дата", или както желаете...

От Toolbox си издърпайте контрола DateTimePicker



Веднага, когато бъде поставен този контрол ще се появи и точната дата, която се показва на операционната система.

Ще ползваме в новото ни приложение още един контрол и това е бутон контрола:) От Toolbox си го вземете и поставете в работната форма.



Преименувайте го на "Покажи!" Text от Properties



Съхранете си проекта (Save All...)

Може сега да тествате приложението:) Разбира се то още не е завършено, но това не пречи да тествате

Не забравяйте, че трябва да активирате всички контроли!

Активирането ставаше като се кликне 2 пъти в/у контрола...и се отваря код едитора...

Вече сме изградили горе-долу потребителският интерфейс на програмката и следва да се добави програмната логика:

Тъй като ще последва някакво събитие при кликането на бутона "Покажи!" , то е ясно, че именно на него ние ще трябва да пишем програмния код...

Кликнете 2 пъти в/у бутон контрола и в код едитора въведете следния код:

```
MsgBox("Датата е: " & DateTimePicker1.Text)
MsgBox("Ден в годината: " & _
DateTimePicker1.Value.DayOfYear.ToString())
```

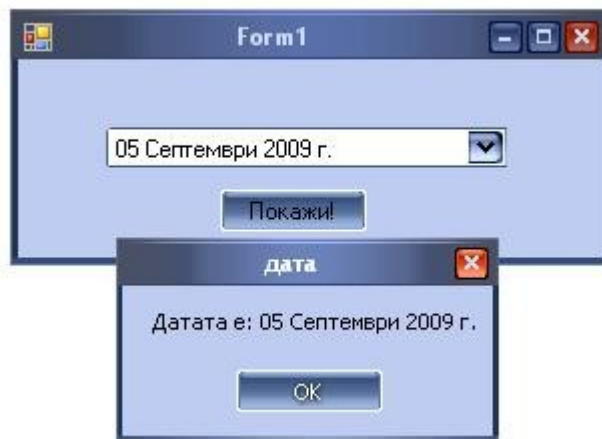
Както забелязвате ползваме 2 MsgBox функции => ще имаме две малки прозорчета със съобщения:) низови конкатенации(string concatenation) или конкатенация на низове:) т.е. имаме тук слепване

За слепване на низове (конкатенация) се използва символът &

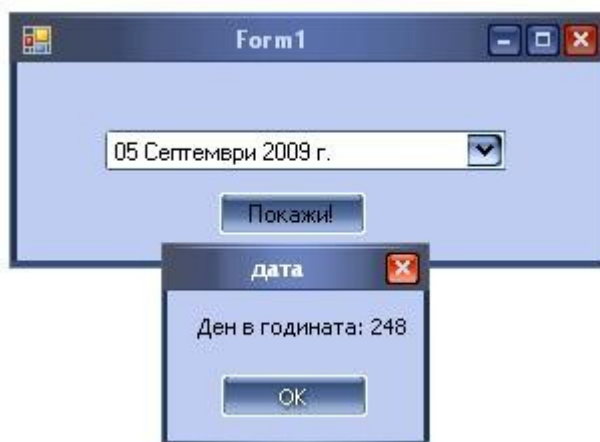
Едното ще извежда датата и годината:) те са си от самият контрол DateTimePicker но после като се кликне на бутончето ОК ще се покаже 2-то малко прозорче със съобщение, което ни показва колко дни са изминали от началото на годината Value.DayOfYear се грижи за всичко това

Програмната конструкция (statement) DateTimePicker1.Value.DayOfYear.ToString() просто прави изчислението като отброява от 1 януари до показаната в момента дата:) ползваме и ToString метод (method) именно той превръща цифровият резултат в този низ и т.н.

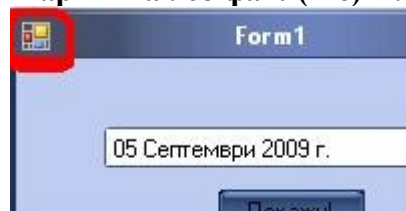
Това е крайният резултат:)



при натискане на бутона ОК се показва 2-то прозорче със съобщението за изминалите дни...



Могат още доста неща да се променят в това нашето семпло приложение:
примерно можем да добавим наша си картинка .ico файл(file) т.е. да променим ето тази:



тази си е на Майкрософт и е такава по подразбиране от самата програмна среда...

За да си поставите ваша картинка трябва да маркирате работната форма(просто кликате един път в/у работният прозорец на приложението) и от свойствата (Properties) да потърсите Icon от раздела Window Style

кликайки там ще се появи едно малко квадратче и го натиснете...остава да намерите от харда подходящ .ico файл (може да си свалите от сайта ми <http://brigante.sytes.net/ico.aspx>)

Така приложението Ви ще си има свой си облик:)

Сега ще пореботим с един друг контрол и това е LinkLabel контролът.

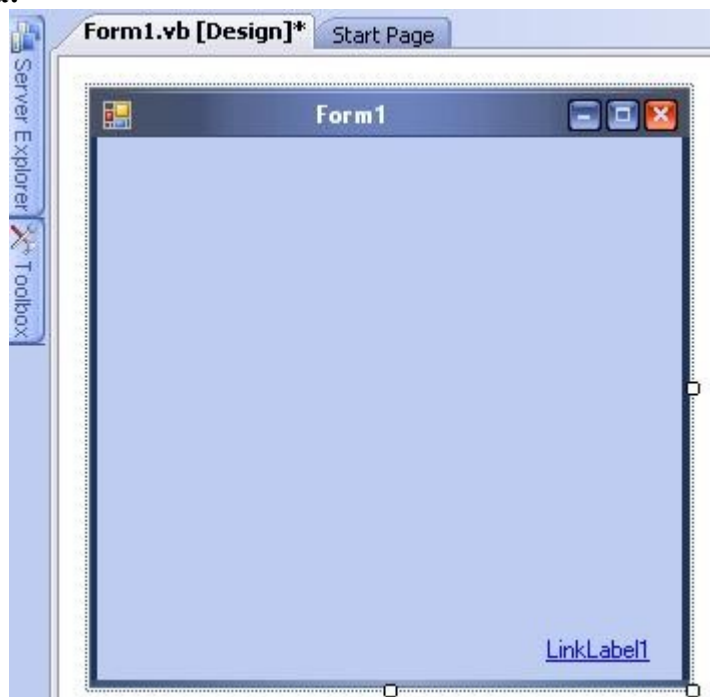
В това упражнение ще се научите как да поставяте link (връзка) в приложението си и при кликане на този линк ще се отваря браузъра по подразбиране и ще отвежда към даден сайт...

За целта от File си създайте нов проект и си го озаглавете примерно “link” или “линк”

От Toolbox си издърпайте контрола LinkLabel ... забележете, че има и контрол Label , така че не ги бъркайте:) ще ползваме доста често контрола Label , когато просто пишем пояснителен текст в нашите приложения.

С мишката си позиционирайте контрола LinkLabel някъде из прозореца, където желаете...

При мен изглежда ето така:



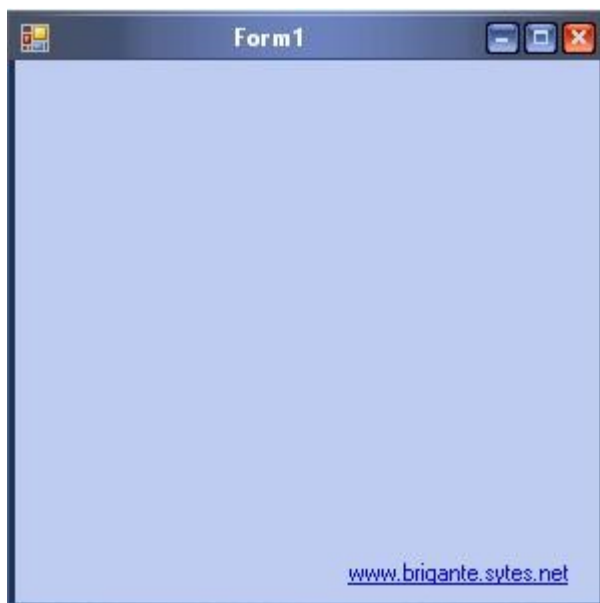
...виждате, че съм си го позиционирал долу в дясно:)

Сега остава да променим текста (да не е LinkLabel1 ,а името на сайта)

Кликнете 1 път в/у контрола и отидете в свойствата т.е. в Properties :) и потърсете Text напишете любимият си сайт или своя си:) и пак позиционирайте контрола...

Не забравяйте да си съхранявате проекта!

Следва Start Debugging :) Спирането на дебъгването ства като просто затворите тестовото си приложение от малкото червено бутонче с хиксче x :) разбира се може и от главното меню Debug и след това Stop Debugging



Разбира се, когато кликате на линка няма да се отвори никакъв браузер та камо ли сайт:)

Самата програмна среда не добавя програмен код в приложението ни...не е както, когато правим уеб сайт с някой редактор (например Dreamweaver 8 или друг:) и редакторът на код да си ни помага и улеснява:) разбира се и тук във Visual Studio си има доста улеснения за програмиста:)

Надявам се, че не сте забравили, че трябва да активирате контрола:) Това ставаше като се кликне 2 пъти в/у него...и ще видите следният програмен код, генериран разбира се от IDE-то

```
Public Class Form1
```

```
Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel1.LinkClicked
```

```
End Sub  
End Class
```

следва да напишем процедурата за събитие или програмната логика :) абе програмният код за самият контрол:) за LinkLabel1_LinkClicked

```
' промяна на цвета, след като сме натиснали линка и му слагаме булева ст-ст истина True  
LinkLabel1.LinkVisited = True  
' ползваме метода Process.Start , за да се отвори браузера по подразбиране дето е:)
```

```
System.Diagnostics.Process.Start  
("http://www.brigante.sytes.net")
```

знакът (') апостроф се ползва за коментар на кода във визуъл бейсик и се игнорира от компилатора ...

т.е. кода трябва да изглежда ето така:

```
Public Class Form1
```

```
Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel1.LinkClicked
```

```
LinkLabel1.LinkVisited = True  
System.Diagnostics.Process.Start _  
("http://www.brigante.sytes.net")  
End Sub  
End Class
```

При така написаният код като се кликне на линка и ще се отвори браузъра по подразбиране, който сте си избрали в системата, но може да направим някои промени в кода ни и да указваме точно с кой браузър да се отваря:) ето го и промененият код:

```
System.Diagnostics.Process.Start("IExplore.exe", _  
"http://www.brigante.sytes.net")
```

т.е. ние вече указваме кой точно браузър да се активира:) разбира се може да си посочите да е с Opera или друг...

В началото на тази ел. книга бях написал малко за този контрол LinkLabel и поставянето на връзка...

С този метод Start , който ни е в кода можете да стартирате и други приложения като Word, Exel и други вече изградени програми...

примерче:

```
System.Diagnostics.Process.Start("Winword.exe", _  
"с:\име на файла.doc")
```

...така като се кликне в/у надписа в приложението Ви и ще се зареди “име на файла.doc” разбира се в програмата Word на Майкрософт :)

Ами това е относно този контрол LinkLabel

Следващото приложение, което ще направим е семпло браузърче...

Създайте си нов проект и го наменувайте “browser” или на бг. ез. “браузър”

от File после New Project и в New Project диалоговият прозорец си избирате пак Windows Forms Application именувате си проекта и кликате на бутона OK

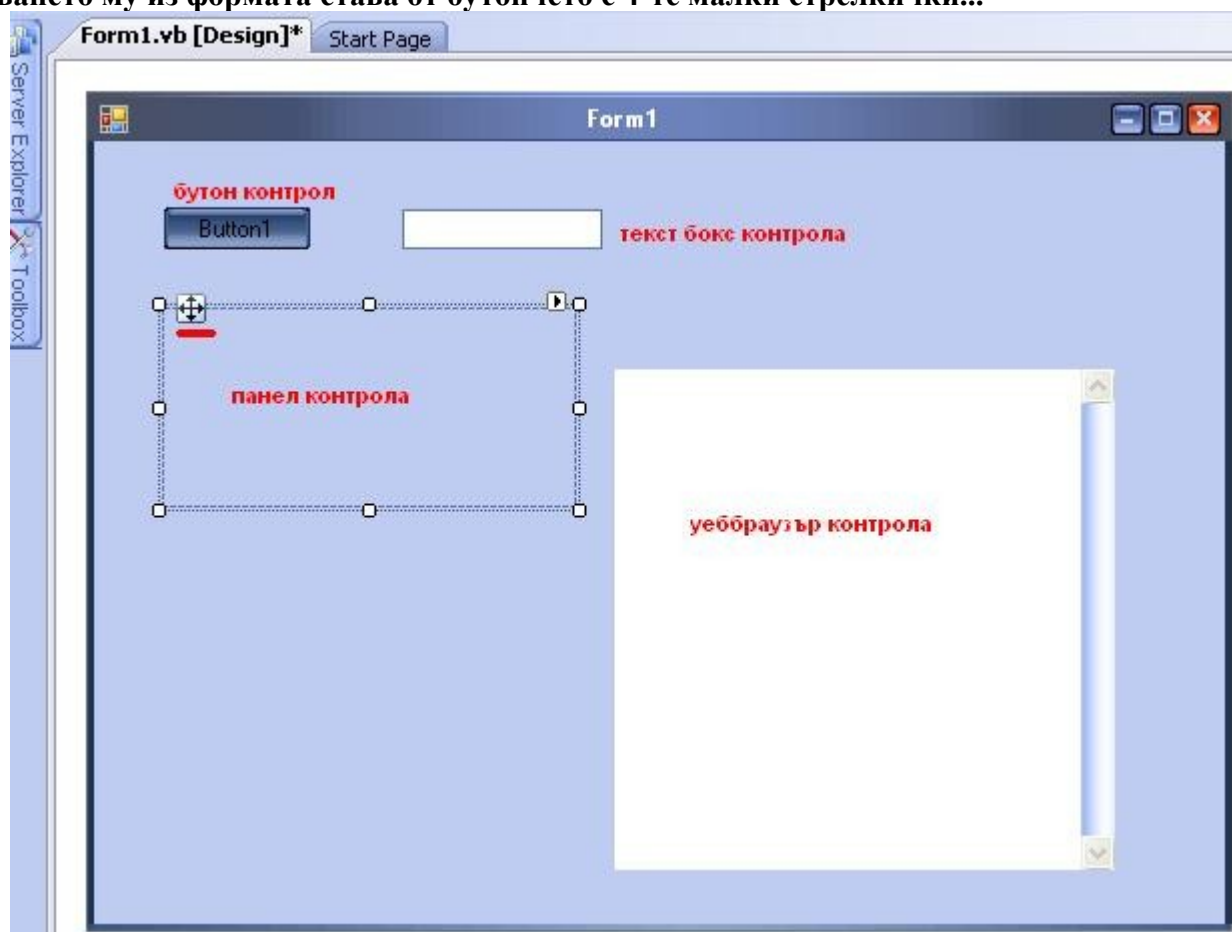
Започваме да си създаваме потребителският интрефейс (User Interface)

Ще са ни необходими 4 контрола. Това са: Button , WebBrowser , Panel , TextBox контролите ...

бутон контрола вече Ви е познат:) TextBox контрола също (служи за въвеждане и извеждане на низ, на текст)

WebBrowser (уеббраузър) контролът сам говори за себе си:) все едно отваряте сайт с Internet Explorer и няма нужда да се пише дълъг програмен код...просто си добавяте контрола

Panel контрола се ползва доста често в програмите:) той придава един вид разграничение измежду поставените в приложението контроли, служи за групиране на други контроли в приложението... преместването му из формата става от бутончето с 4-те малки стрелкички...

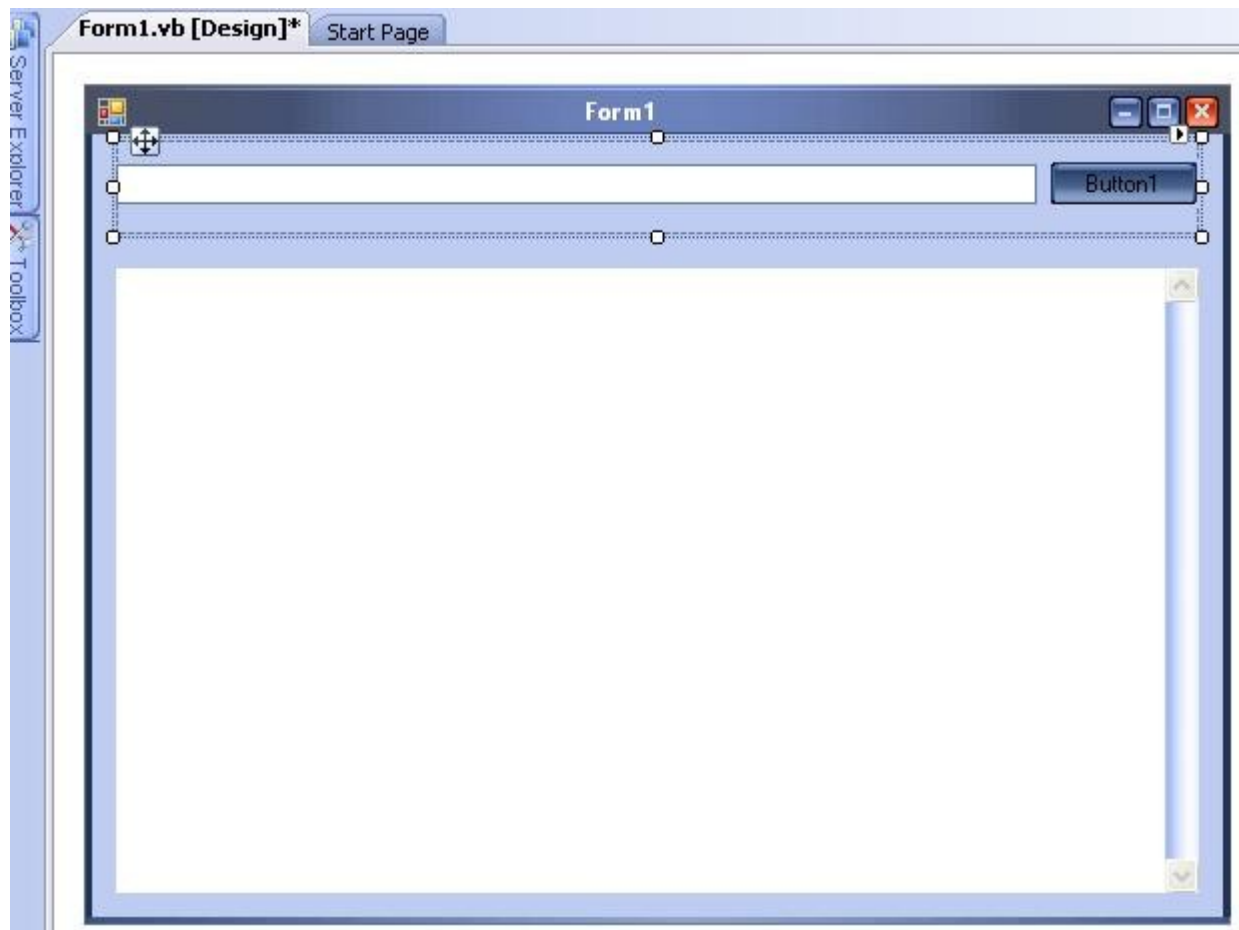


...при мен засега изглежда ето така потребителският инетефейс и разбира се няма да остане така:)

Активирайте всички контроли! Кликахме 2 пъти в/у всеки един ...

Оразмерете си контролите като с мишката ги издърпвате в краищата...

Опитайте се приложението Ви да добие следният изглед:



Съхранявайте си проекта!

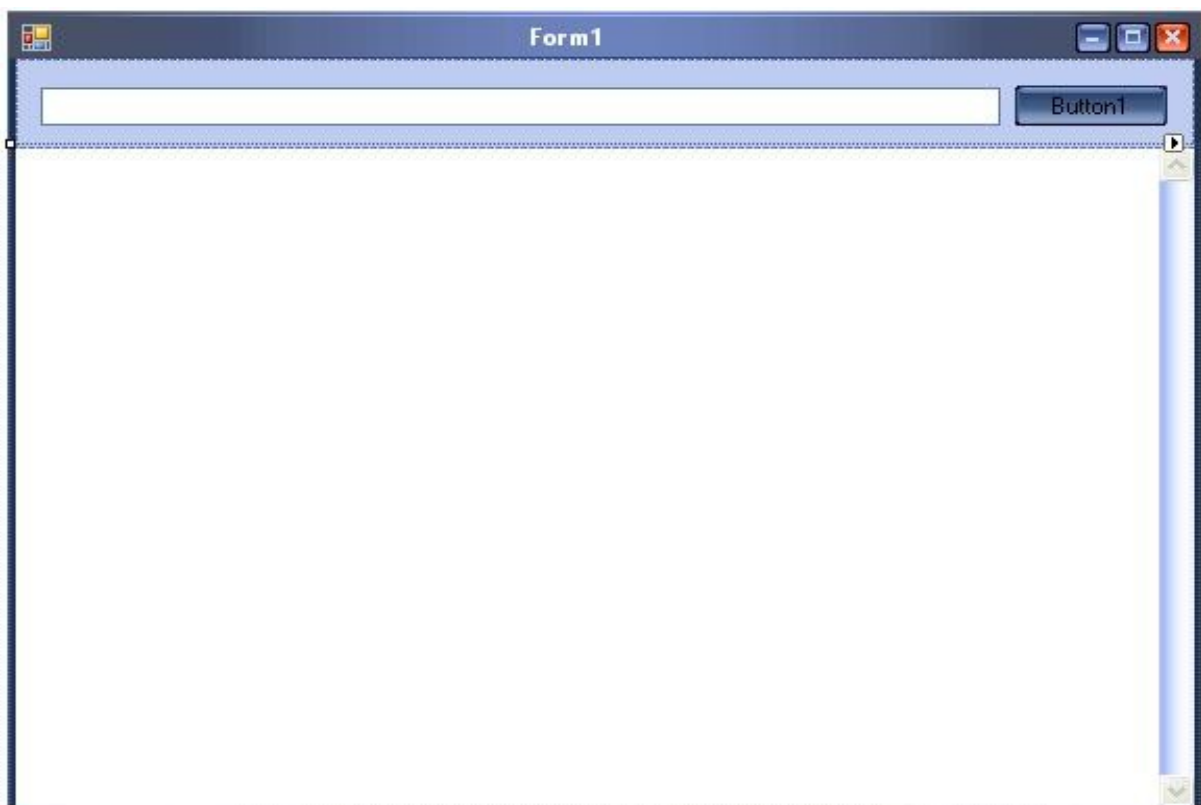
Сега ще придадем някои св-ва на контролите.

Маркирайте контрола Panel и отидете в Properties

Там потърсете Dock и сложете позицията да е на Top т.е. да се позиционира и разпъне изцяло отгоре...

Маркирайте контрола WebBrowser от работната ни форма и отидете да му придадете и на него някои св-ва:) Пак потърсете Dock св-вото и го разпънете на цяло т.е. изберате Fill

Пак си съхранете проекта! :) След всичко направено до тук приложението ни трябва да изглежда ето така:



Може сега ако искате да си дебъгнете (F5) и разбира се все още нищо няма да се случва при натискане на бутона, който всъщност сега трябва да го преименуваме от Properties Маркирате си го и потърсете Text в Properties прозорчето:) Аз си го кръщавам:) “Покажи!” може и “Зареди!” или както си искате:) и кликайте някъде из формата, за да се извърши промяната:) С това ние завършваме изграждането на потребителския интерфейс и ни остава да направим програмната логика...

За бутон контрола (кликнете 2 пъти в/у него) ще напишем следния код: Забележете, че винаги при кликането в/у даден контрол бива позиционирани в код едитора точно, където трябва да се пише програмният код...

WebBrowser1.Navigate(Textbox1.Text)

т.е. целият код в едитора трябва да е следния:

Public Class Form1

```
Private Sub Panel1_Paint(ByVal sender As System.Object, ByVal e As System.Windows.Forms.PaintEventArgs) Handles Panel1.Paint
End Sub
```

```
Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    WebBrowser1.Navigate(TextBox1.Text)
End Sub
```

```
Private Sub WebBrowser1_DocumentCompleted(ByVal sender As System.Object, ByVal e As System.Windows.Forms.WebBrowserDocumentCompletedEventArgs) Handles WebBrowser1.DocumentCompleted
End Sub
```

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
End Sub
End Class
```

аз кликах и в/у формата и затова имам

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

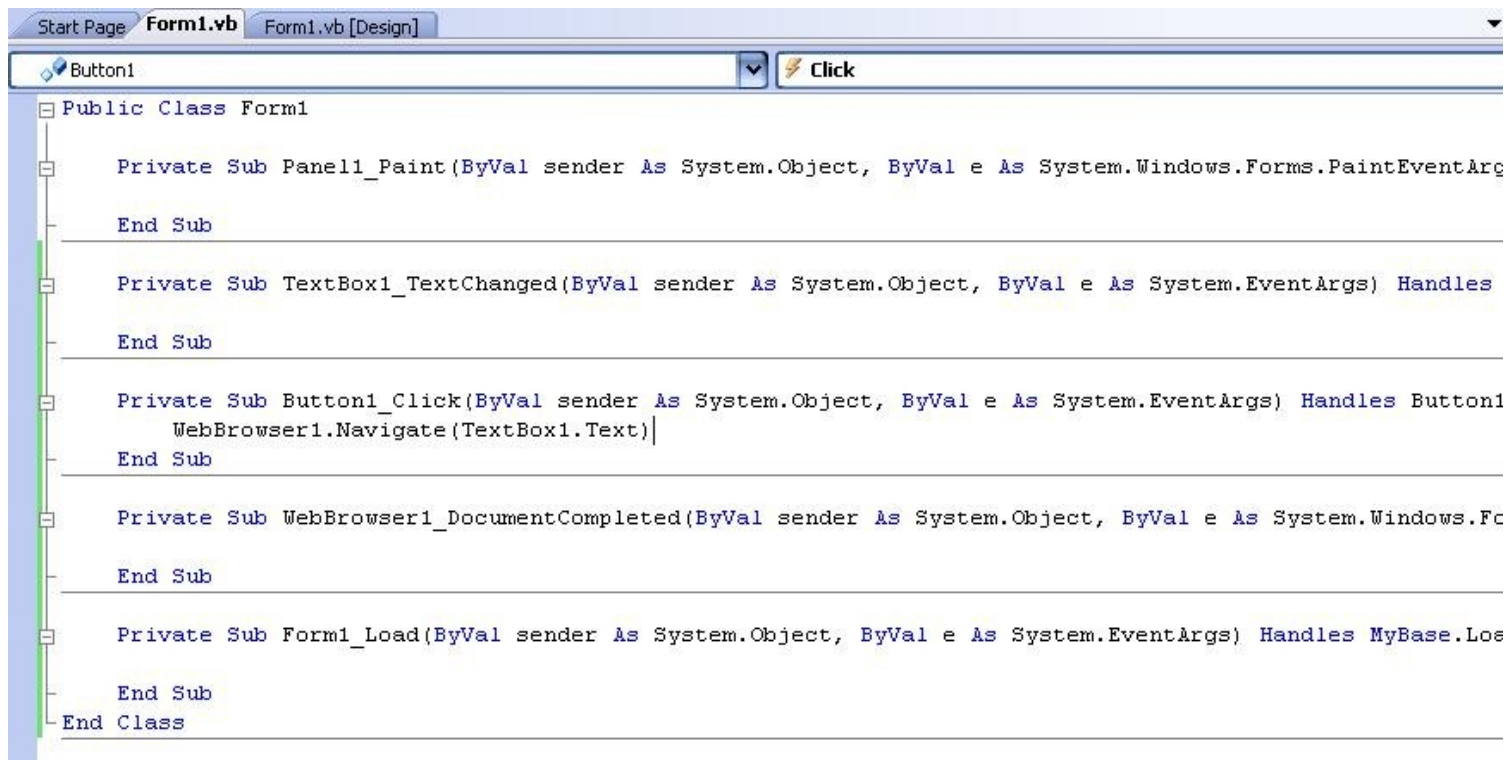
може и без него:) както и да е...

Този кратък код, който написахме за бутон контрола всъщност “казва” на програмата ни да изпълни написаното в TextBox1 и да мине да се изведе към WebBrowser1 контрола , като това става с метода Navigate

Забележете, че контролите се номерират т.е. не са WebBrowser ,а WebBrowser1 и ако добавим още един път такъв контрол WebBrowser , то ще стане WebBrowser2 и т.н.

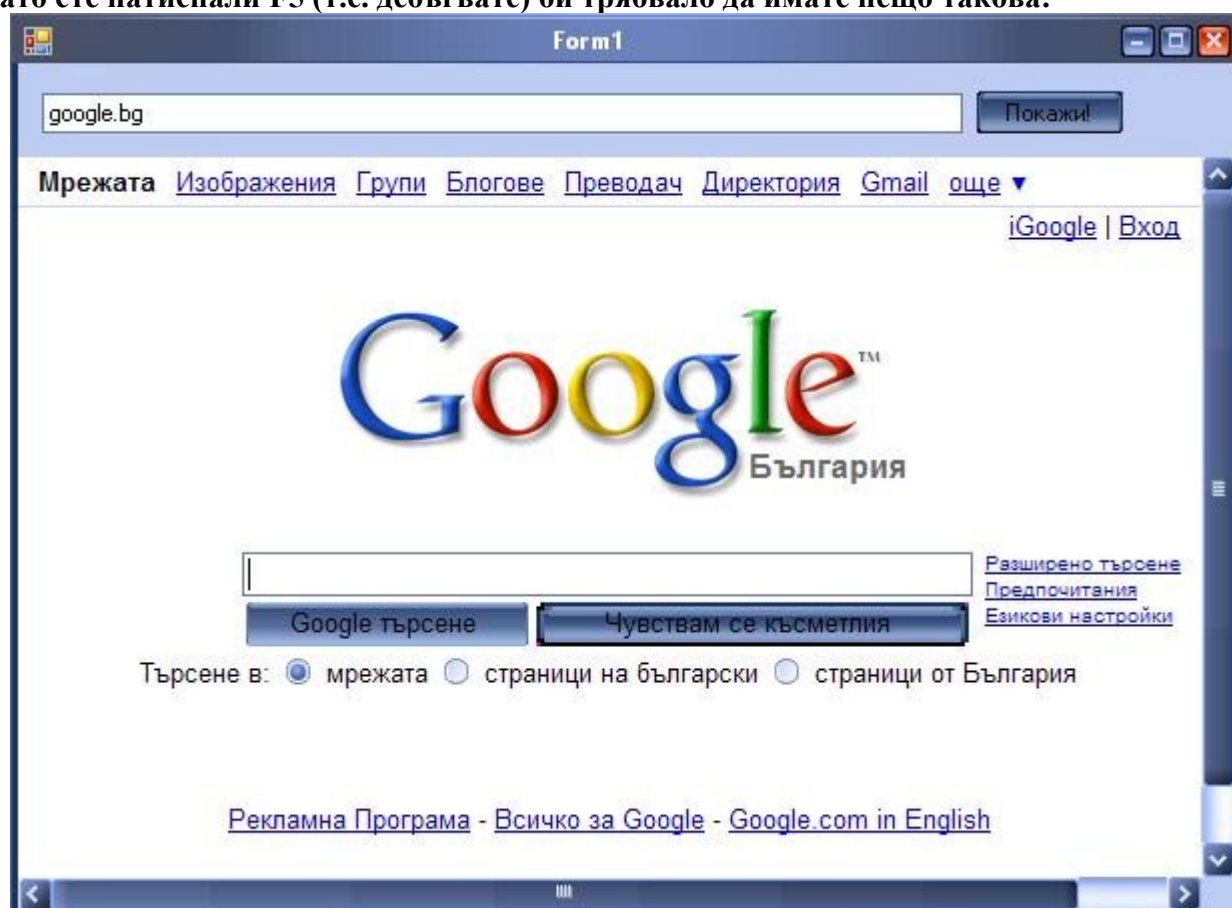
Не е нужно да пишем повече програмен код за това наше приложение:)

Извършва се някакво действие, когато се клика на бутона и затова само там ние трябва да сложим код...



така трябва да изглежда кода в редактора...

А след като сте натиснали F5 (т.е. дебъгвате) би трябвало да имате нещо такова:



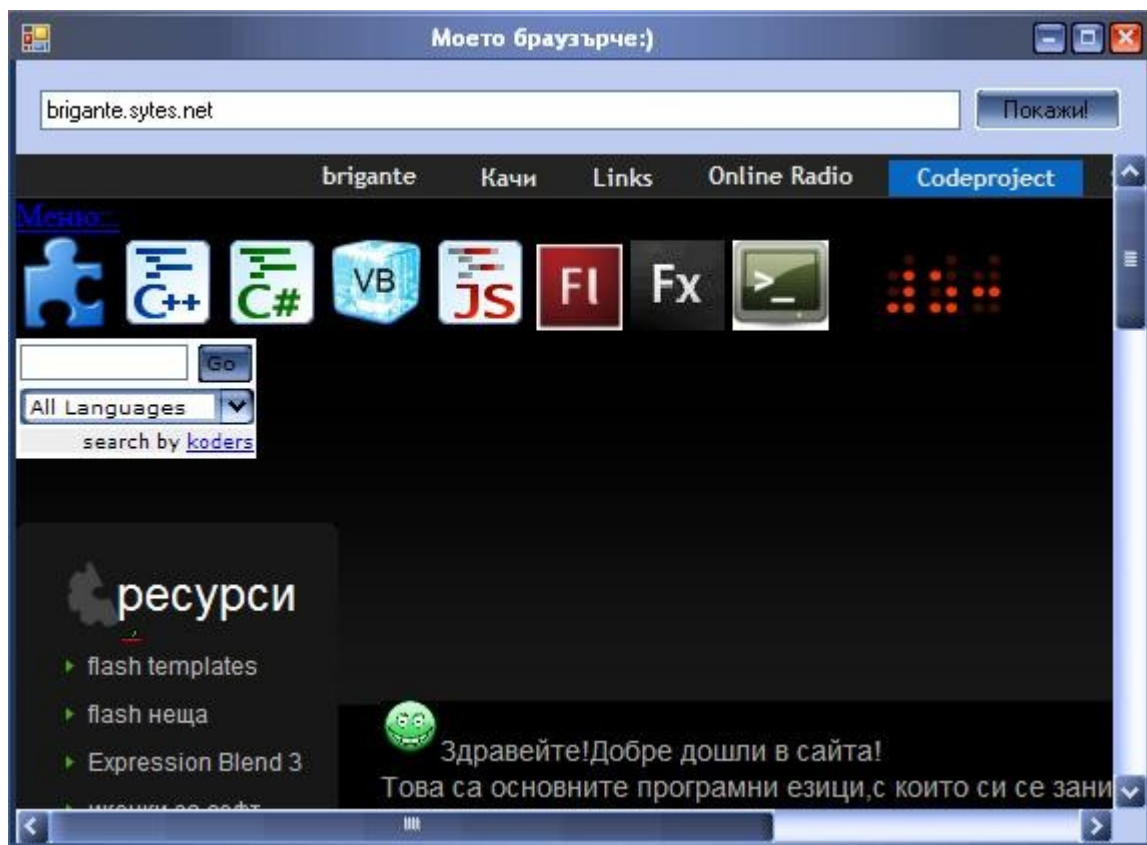
Разбира се със самото стартирането на приложението няма да се зареди веднага страницата на гугъл т.е. нямаме Home Page от IExplorerа:) , а трябва да се въведе сайта горе в TextBox контрола и да се натисне бутона “Покажи!” или както сте си го именували.

За да спрете дебъгването просто кликнете на Close бутона от приложението Ви или на Debug от главното меню и кликнете на Stop Debugging

Може да си смените името на формата (да не е Form1)

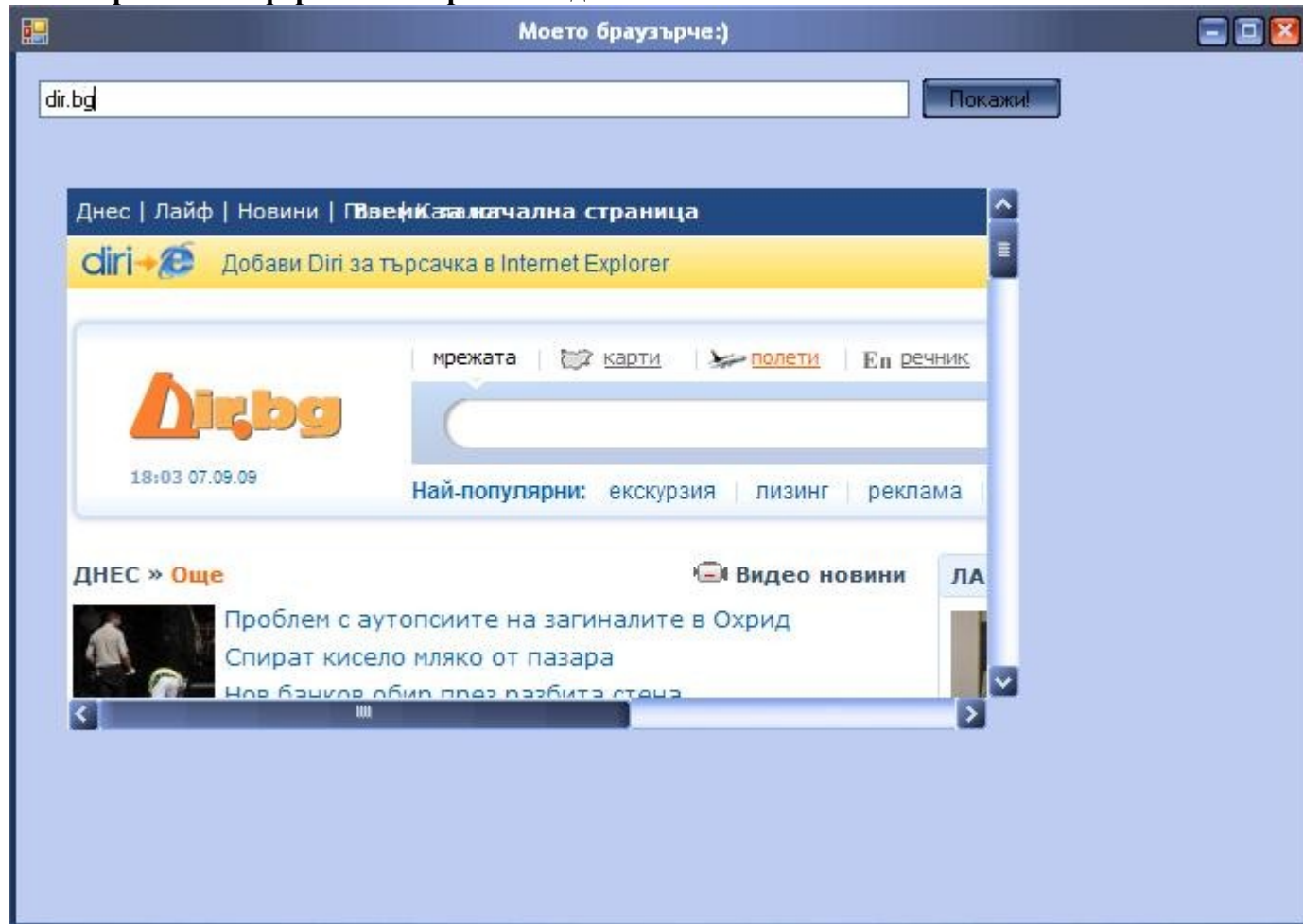
Примерно може да си го кръстите “Моето браузърче” или друго...

За целта маркирате цялата работна форма и отивате на св-ва и търсите Text и сменяте текста...



И приложението вече е с др. име:)

Когато е стартирано приложението и се кликне на малкото бутонче Maximize , приложението ще се разпъне по целият дисплей и именно затова придадохме св-вото Dock Fill на контрола WebBrowser , за да не стане така, че когато е на голям екран уеббраузър контрола да си е мъничък какъвто сте го оставили в работната форма... т.е. при Dock да е None



При Dock със св-то None ще изглежда така:) и точно затова му придаваме Fill ,за да си следва прозореца при каквото и да е оразмеряване от потребителя на приложението ни...

...някой ден => продължение:)