

Uvod u C# i .NET



UVOD

Uvod

Uvod u C# i .NET Framework, i Prva grafička aplikacija u visual C#

'U okviru ove lekcije, bavićemo se sledećim pitanjima: zašto koristiti C#, poređenjem C# sa drugim jezicima, u čemu je razlika između C# i Visual C#, zatim okruženjima koja se koriste kod programiranja u C#.

U ovoj lekciji cilj je da se napravi prva grafička aplikacijau Visual C#. Ovo je jednostavna aplikacija ali postavlja fundament za mnogo složenije aplikacije. Takođe prvi put ćemo koristiti ključne alate kao što su Toolbox i Property-editor.

Ključna pitanja:

Zašto koristiti Visual C#?

Koja okruženja i okviri se koriste za Visual C#?

Kako se kreira grafička aplikacija?

Za šta služi Toolbox?

Za šta služi Property-editor?

Osobine C#

01

OSObine C#

C# je lak za učenje i moćan za upotrebu.

Microsoft Visual C# je s jedne strane vrlo moćan a sa druge strane jednostavan za učenje programski jezik. Koristi se za razvoj aplikacija, pri čemu se upotrebljava radni okvir Microsoft .NET. Jezik C# koristi najbolje osobine jezika C++ i **Microsoft Visual Basic** jezika. Jezik C# 2.0 pojavio se 2005. godine, i tada je uveden generički koncept, dok prva verzija C# se pojavila 2001. god. Jezik C# 3.0 je dostupan u okviru Microsoft Visual Studija 2008, i on obuhvata **Language Integrated Query**. Razvojno okruženje Visual Studio 2008 čini jezik C# brzim tj. elegantnim za krišćenje, omogućujući brzo i efikasno pravljenje raznih aplikacija, uključujući aplikacije sa bazama podataka i aplikacijama na Internetu.

C# je programski jezik koji omogućuje da se na najbolji način prave programi za operativni sistem Windows i za Internet. Može se postaviti pitanje : zašto koristiti C# ? Odgovor je da C# , tj. C-*sharp* poboljšana verzija jezika C/C++. Odnosno, C# je baziran na C/C++, i osim toga to je verzija koja je „oštra“ tj. moćnija, kao i elegantnija, *sharper-smarter*, od samog originala tj. od C/C++. C# je relativno „mlad“ jezik, i to mu omogućuje da se bazira na modernom i najnovijem konceptu kako da programeri napišu programe koji su pouzdani i efektivni. Takodje, u potpunosti je napravljen i usaglašen za upotrebu na Internetu. Kad se kaže usaglašenost sa erom Interneta, to znači da omogućuje ne samo da jedan program se izvršava za jednu osobu na njenjegovom radnom stolu, već i izvršavanje programa tako da komunicira sa drugim kompjuterima na nekoj lokalnoj mreži računara, ili da komunicira sa računarima širom sveta preko Interneta.

Visual Studio 2008 ili 2010 je programsko okruženje koje sadrži niz vizuelnih alata, koji omogućuju da se prave brzo i efikasno C# projekti. Mogu se praviti čak i projekti gde se kombinuju dva programska jezika. C# ima značajnu ulogu u arhitekturi **Microsoft .NET Framework**-a. Slično je jezik C igrao veliku ulogu u razvoju UNIX-a. Ako već poznajete C/C++ ili jezik Java, onda će te zaključiti da je sintaksa jezika C# vrlo slična sa sintaksom ovih jezika.

POREDJENJE C#

C# je napravljen da veoma liči na C/C++, ali za razliku od C/C++, C# je lak tj. brz za učenje.

C# se pojavio 2001. god. To je dakle relativno nov jezik, ali je napravljen da bude sličan sa C/C++, tako da ga poznavaooci C/C++ lako uče i razumeju. Jezik Java je takodje baziran na C/C++, pa poznavaooci Java takodje mogu lako da nauče C#. C# je elegantan, jednostavan, moćan, i potpuno objektno-orijentisan, i predstavlja vrlo dobar izbor za nove programere. S druge strane, C/C++ je izvanredan i izuzetno moćan jezik koji koriste iskusni programeri, i ogroman broj programa je napisan u tom jeziku, ali ima reputaciju da je relativno težak za učenje tj da je relativno komplikovan. Npr. upotreba tzv. lokatora tj pokazivača, *pointers*, u C/C++ je komplikovana. C# je napravljen da veoma liči na C/C++, ali za razliku od C/C++, C# je lak tj. brz za učenje. Naime, kod C#, memorijski menadžment je automatski, lokatori tj pokazivači, *pointers*, se retko koriste, i ne koriste se “zaglavni fajlovi” ili “makroi”, *header files* ili *#define macros*.

Takođe, *Visual C#*, Vizuelni C#, koristi vizuelne alate, *the visual development tools*. Npr. vizuelni alati u razvojnom okruženju *Visual Studio*-u omogućuje da se naprave raznolike web-stranice kao i desktop-stranice, i to jednostavno i brzo pomoću tehnike: *drag-and-drop*. Tj. *Visual C#* kombinuje efikasnost i produktivnost jezika *Visual Basic* sa C-stilom tj. stilom jezika C/C++. Kad kažemo *Visual C#*, to je znači jezik C# kombinovan sa vizuelnim alatima, npr. alatima u *Visual Studio*-u. A *Visual Studio* je razvojno okruženje koji omogućuje olakšani i elegantniji i brži razvoj složenih programa u jeziku C#. Ali u suštini *Visual C#* je jezik C#, uz dodatne vizuelne alate. *Visual C#* je specijalizovan za pravljenje tzv. „formi“, gde se pod „formom“ podrazumeva neki „prozor“ na ekranu gde se prikazuju odredjeni podaci, odredjena struktura podataka, u odredjenom unapred definisanom i pogodno dizajniranom formatu. Na tu „formu“ mogu se postaviti razni objekti, npr. objekti **Button**, tj. „dugme“, **Label**, tj. „nalepnica“, **TextBox**, tj. „tekstboks“, itd.

JEZIK C#

C# je potpuno objektno-orijentisan jezik.

I u jeziku C#, kao I u C/C++ ili Java, mogu se:

- Definisati promenljive
- Mogu se koristiti aritmetički operatori
- Mogu se definisati metode tj. funkcije i koristiti argumenti metoda tj. funkcija
- Takođe if-iskazi i while-petlje su deo jezika C#. Isto tako,
- u C# se koriste izuzeci, exceptions, da bi se procesirale greške u programu
- Konačno, C# je potpuno objektno-orijentisan jezik, gde se mogu koristiti svi principi objektno-orijentisanog programiranja,
- npr. nasledje klasa, sakrivanje podataka, konstruktor-metode, itd

VISUAL C#

.NET je okvir, u mnogim oblicima sličan sa Java bibliotekama

Dakle, Visual C# je znači jezik C# kombinovan sa vizuelnim alatom, npr. razvojnim vizuelnim alatom Visual Studio 2008 ili 2010 ili 2012. Mogu se nabrojati sledeći razlozi zašto je korisno programirati tj. pisati programe u Visual C# :Visual C# je potpuno objektno-orijentisani jezik, Omogućuje lako i brzo programiranje moćnih Web-aplikacija, Visual C# se puno koristi od strane kompanije Microsoft, što ga čini pogodnim jezikom za operativni sistem Windows koji je od iste kompanije.

Glavno okruženje za C# je Visual C# .NET (Microsoft), koja u stvari ima nekoliko verzija. Npr.:Visual C# Standard Edition. To je verzija koja je idealna za početnike. Tu je sve što treba za pisanje programa za operativni sistem Windows. Dok okruženje Visual Studio .NET obuhvata jednu pojačanu verziju Visual C# (u odnosu na Visual C# Standard Edition) , i ovo okruženje takodje sadrži druge jezike, npr. C++, ili Visual Basic, ili čak sadrži J# koji liči na Javu. Pri tome, verzija:

Visual Studio .NET Professional Edition, je kompletno razvojno okruženje. Postoje još jače verzije od ove, i to su *Visual Studio .NET Enterprise Developer*, kao i *Visual Studio .NET Enterprise Architect*. Ove tri profesionalne verzije su komercijalne, dakle nisu besplatne već moraju da se kupe i tako obezbedi licenca za legalizovan rad.

.NET

C# je i jezik i istovremeno i objektno-orijentisana biblioteka.

Microsoft kompaniji nije odgovaralo da koristi Javu pod licencom, pa je morao da smisli kako da konkuriše Javi. Osim toga, Java iako je moćan jezik, ima nedostatak da se praktično zahteva da kompletan program mora biti u Javi, ako želite da iskoristite Javu u potpunosti. Međutim, Microsoft je imao milione i milione linija izvornog koda u drugim jezicima od Jave. Tako da je morao da uvede nešto što će potpomagati istovremenu upotrebu različitih jezika. Uveli su .NET, *dot net*.

.NET je okvir, u mnogim oblicima sličan sa Java bibliotekama, jer C# je vrlo sličan sa jezikom Java. Java je istovremeno i jezik i biblioteka programa. Isto tako, C# je i jezik i istovremeno i objektno-orijentisana biblioteka koja pojednostavljuje programiranje, u svim oblastima, od Web baza podataka do Windows dijalog-boksa.

VISUAL C# FORMA

Pod „formom“ se podrazumeva neki „prozor“ na ekranu gde se prikazuju odredjeni podaci, u odredjenom unapred definisanom i pogodno dizajniranom formatu.

Visual C# je specijalizovan za pravljenje tzv. „formi“, gde se pod „formom“ podrazumeva neki „prozor“ na ekranu gde se prikazuju odredjeni podaci, odredjena struktura podataka, u odredjenom unapred definisanom i pogodno dizajniranom formatu. Na tu „formu“ mogu se postaviti razni objekti, npr. objekti

Button, tj. „dugme“,

Label, tj. „nalepnica“,

TextBox, tj. „tekstboks“,

ListBox,

CheckBox,

PictureBox

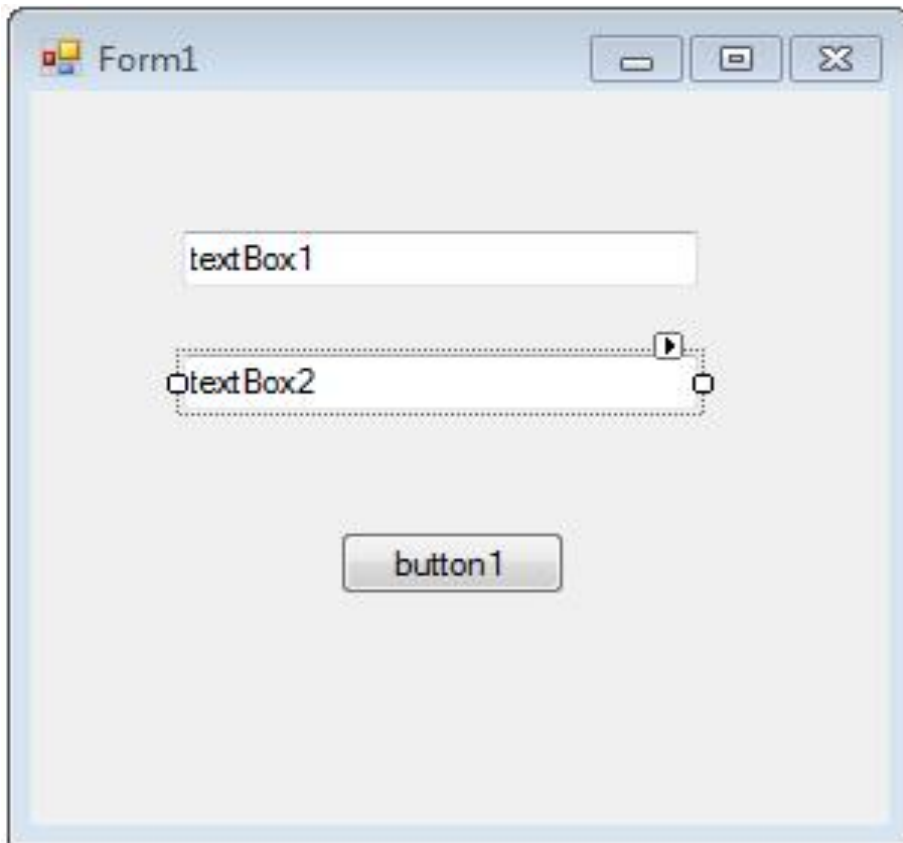
DataGrid,

RadioButton,

itd.

FORMA SA DVA „TEKSTBOKSA“ I JEDNIM „DUGMETOM“

Na slici je forma sa dva „tekstboks“a i jednim „dugmetom“



UPOTREBA VISUAL C#-FORME

Na formi su u gornjem desnom uglu dugmići za smanjivanje, povećavanje i zatvaranje forme (Minimize, Maximize, Close), jer „forma“ j ima osobine „prozora“.

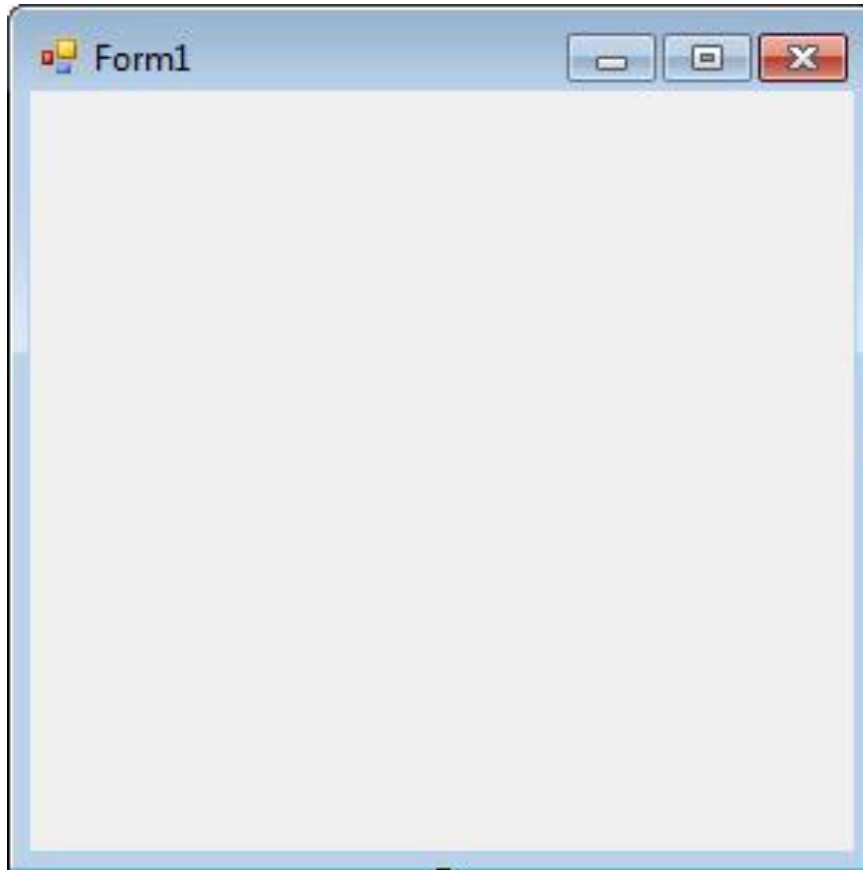
Većina grafičkih aplikacija u Visual C# su bazirani na tzv. „formi“, tj. C# - formi tj. Visual C# - formi. Forma u jeziku Visual C# označava jedan specijalni „prozor“, tj. „platno“ za prikazivanje određenih podataka, tj. gde se mogu postavljati i editovati razni objekti, koje smo pomenuli npr. **Label**, **Button**, **TextBox**, itd. Neka Forma ima karakteristike, *properties*, kao i ostali objekti u Visual C#.

Ako se klikne bilo gde na pozadini „forme“, onda na **Properties Window** se pojavljuju *properties* tj. karakteristike od Forme, i to: Icon, Text, BackColor, itd. Npr. osobina BackColor označava boju u pozadini „forme“.

Na formi su u gornjem desnom uglu dugmići za smanjivanje, povećavanje i zatvaranje forme (Minimize, Maximize, Close), jer „forma“ j ima osobine „prozora“, kao i svi ostali prozori u Windows operativnom sistemu. U gornjem levom uglu je ikonica, mala slika, koja se pojavljuje umesto „forme“ kada se „forma“ minimizira. Posle ikonice, u vrhu „forme“, nalazi se tekst koji predstavlja naslov „forme“. Dole je data skica „forme“. Koristićemo ovu skicu ponegde, jer je zgodna za editovanje. Na vrhu skice nalazi se ime forme, kao i sa desne strane 3 ikonice koje označavaju Min, Max, i **Close**, dok ikonica krajnje levo označava ikonicu „forme“.

VISUAL C# FORMA

Slika prikazuje Visual C# formu



Visual Studio

02

VISUAL STUDIO

Okruženje Visual Studio obuhvata nekoliko jezika, C++ i C# i J#, ali svi oni koriste iste vizuelne alate.

Pomoću okruženja za Visual C#, npr. Visual C# Standard Edition, vrši se kompilacija neke aplikacije C# tj C# - programa. Pri tome, vrši se kompilacija za **.NET Framework**, a to je okvir koje pruža pristup velikom nizu korisnih komponenti i funkcija. Ove funkcije omogućuju optimizaciju upotrebe memorije, i sigurnost i pouzdanost izvršavanja programa napisanih u C#. Da bi se izvršavala neka aplikacija tj. program u C#, mora se instalirati okvir **.NET Framework**, i ovo je besplatno za distribuciju. Obično, u zanje vreme, je **.NET Framework** uključeno u operativni sistem Windows.

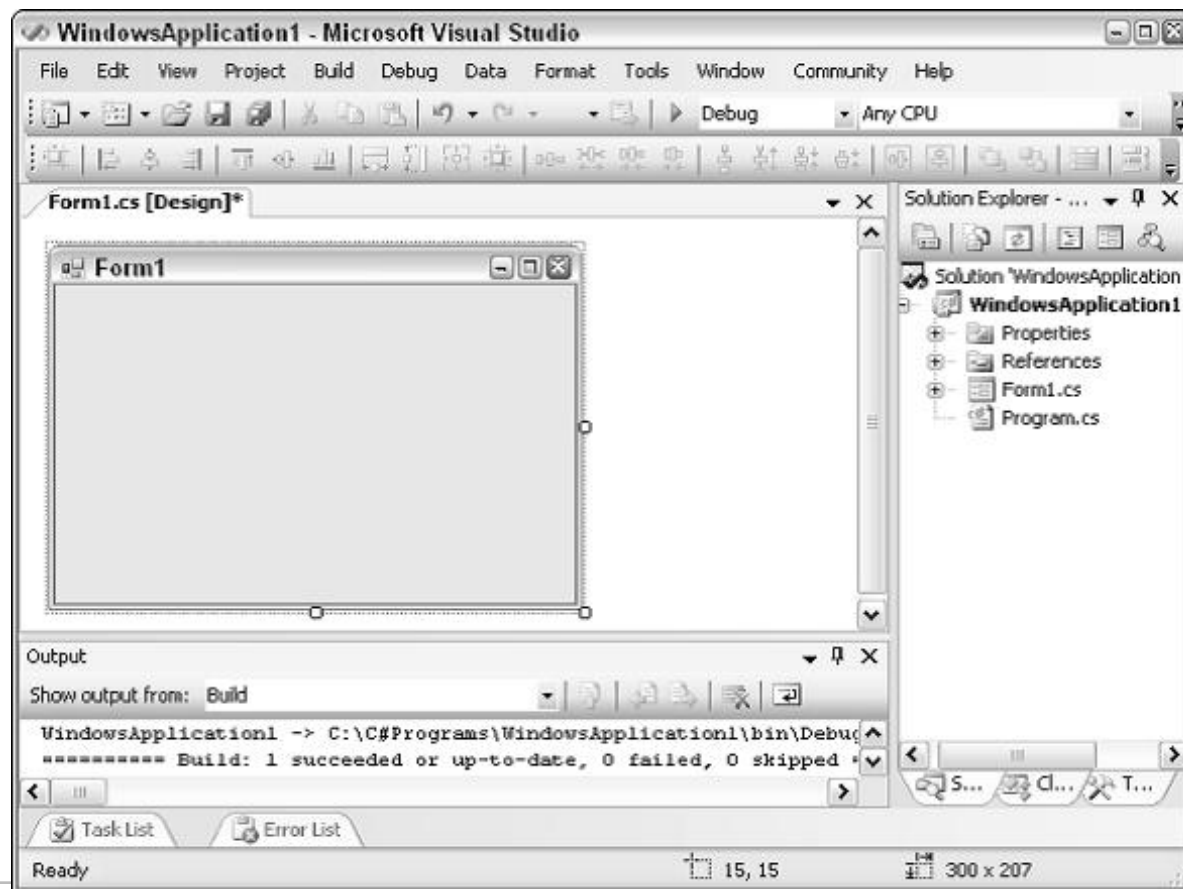
Okruženje Visual Studio obuhvata nekoliko jezika, C++ i C# i J#, ali svi oni koriste iste vizuelne alate. I ako želite neku profesionalnu verziju Visual C#, onda morate da kupite Visual Studio. Dakle, okruženje *Visual Studio* nije besplatno. Ako koristite Visual Studio 2010 Standard ili Visual Studio 2010 Professional, treba uraditi sledeće da se startuje Visual Studio 2010: kliknuti **Start button**, onda uperiti miš na **All Programs**, i onda na *Microsoft Visual Studio 2010* grupu programa. Onda kliknuti *Microsoft Visual Studio 2010*. Tada Visual Studio 2010 se otvara, kao što je dole prikazano. Prvi Microsoft vizuelni jezik bio je Visual Basic. Nazvan je Visual jer koristi GUI. Danas, Microsoft je okupio sve jezike u jedno okruženje, Visual Studio 2010 tj. 2012. U stvari ranije je bio to Visual Studio, a sada je Visual Studio .NET, da bi se naglasila veza između Visual Studio-a i .NET-a.

Razvojno okruženje *Visual C# Express Edition* je odlično za početnike, jer je besplatno, i izvanredno. Naime, to je integrisano razvojno okruženje, IDE, **Integrated Development Environment**, koje obuhvata editor, kao i skup ostalih alata za razvoj projekta tj. aplikacije u Visual C#. Inače, IDE za C++ je obično frustrirajuće, kada se poredi sa IDE za **Visual C# Express**. *Visual C# Express Edition* omogućuje pravljenje programa u C# na lak i brz način, pomoću tehnike „selektovanja i prevlačenja“, *drag-and-drop*. A takođe vrši se automatsko kodiranje nekih monotonih blokova programskog koda, umesto programera. Visual C# Express Edition 2008 može se besplatno preuzeti sa Interneta, sa adrese:

www.microsoft.com/express/download.

SCREENSHOT PRVE STRANE IDE SA PRIKAZANOM „FORMOM“

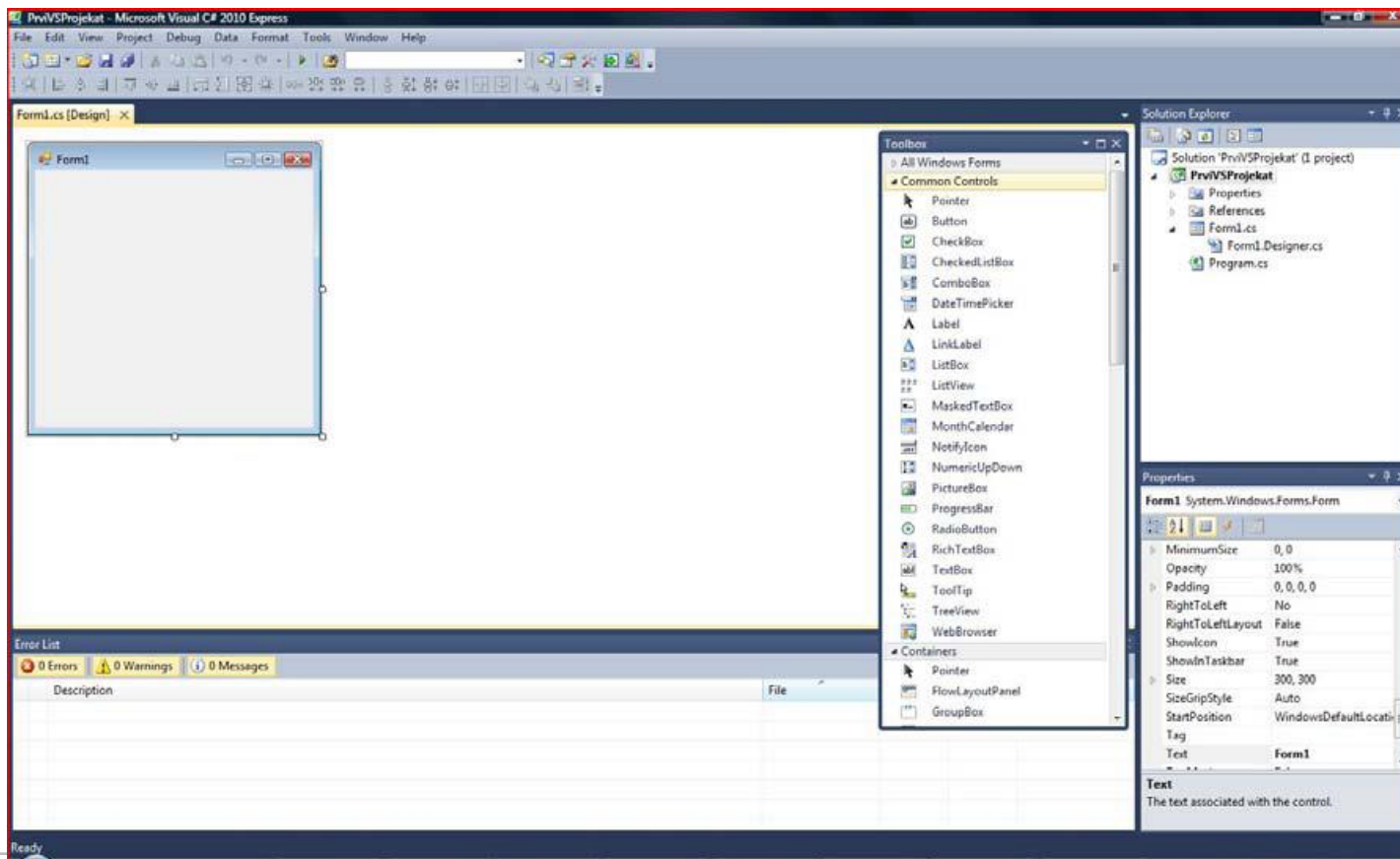
Slika prikazuje screenshot prve strane IDE (Visual Studio 2005) sa prikazanom „formom“.



Slika 1: Visual studio

SCREENSHOT PRVE STRANE IDE VISUALC# 2010 EXPRESS SA PRIKAZANOM „FORMOM“

Slika ilustruje screenshot prve strane IDE VisualC# 2010 Express sa prikazanom „formom“



INSTALISANJE VISUAL C#

*Pri instalaciji, može biti zahtevano da se instališe **Windows Component Update**.*

Instalisanje Visual C#, sastoji se u instalisanju okruženja za Visual C#, i to, ili osnovne verzije **Visual C# .NET** (Visual Studio 2010 Standard), ili kompletnije verzije **Visual Studio .NET** (**Visual Studio 2010 Profesional**). Da bi se to uradilo, npr. da se instališe Visual C# .NET, potrebno je izvršiti instalacioni program, *setup*, tj.

Visual C# .NET Setup

I pri tome pratiti uputstva koja se dobijaju od tog instalacionog programa.

Pri tome, može biti zahtevano da se instališe

Windows Component Update,

A to je skup sistemskih fajlova koji se koriste od strane ne samo Visual C#, nego i od strane **Windows Software**.

C# WEB SERVER

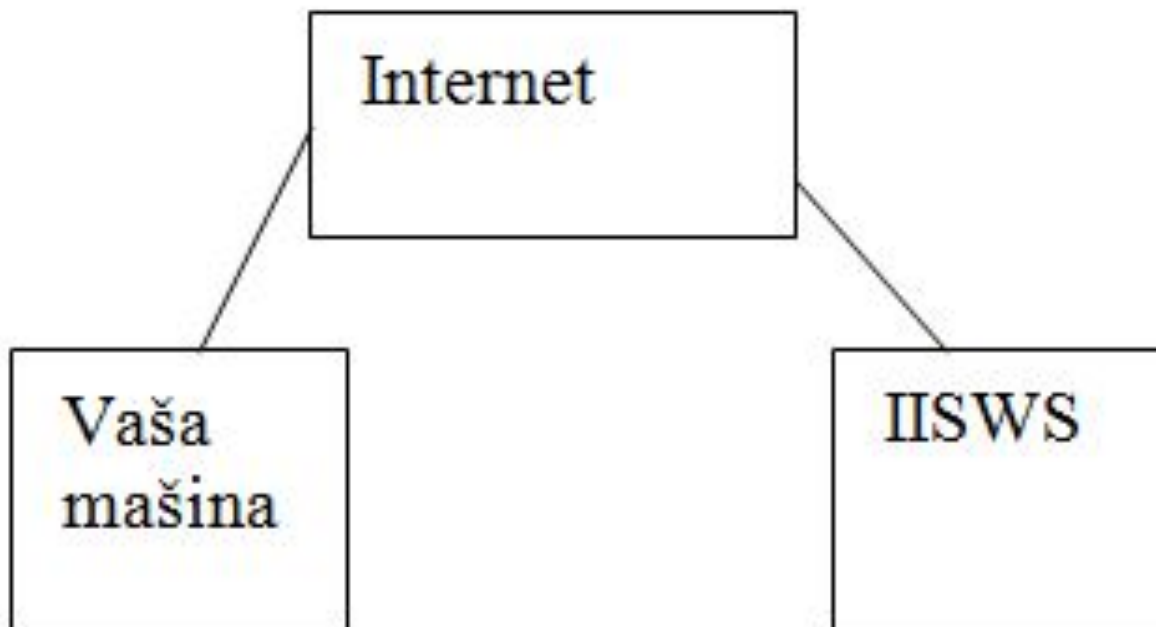
I bez Web-servera može se koristiti Visual C#, ukoliko ne želite da kreirate Web-aplikacije.

Pomoću Visual C#, mogu se praviti Dinamičke Web-aplikacije, ali ove aplikacije zahtevaju da se ima neki **Web-server**. Tako da postoji opcija, kod instalisanja Visual C#, da se instalise na sopstvenoj mašini jedan Web-server. Medjutim, i bez Web-servera može se koristiti Visual C#, ukoliko ne želite da kreirate Web-aplikacije. Tako, da se **Web-server** može instalisati kasnije.

Takođe, kod instalisanja Visual C#, može se izvršiti selekcija komponenti koje su potrebne korisniku, npr. s obzirom da dokumentacija može da zahteva puno prostora, može se ona izostaviti po želji. Ukoliko ste na nekoj mreži, možda ima **Internet Information Services Web Server**, IISWS, na nekoj drugoj mašini a ne vašoj, pa se umesto da instalishete Web server na vašoj mašini možete odlučiti da koristite **Web server** IISWS.

WEB SERVER IISWS

Na slici je ilustrovan Web server IISWS.



Visual Studio alati

03

VISUAL STUDIO ALATI

Ima niz alata u okviru Visual Studio-a.

U Wikipediji može se naći opis Visual Studio-a. u okviru tog opisa navode se alati koji su raspoloživi u Visual Studio-u:

Kodni editor, Dizajner forme, Klasni dizajner, Web dizajner, itd

From Wikipedia, the free encyclopedia:

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight.

Visual Studio includes a code editor supporting IntelliSense as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building GUI applications, web designer, class designer, and database schema designer.

Visual Studio supports different programming languages by means of language services, which allow the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C/C++ (via Visual C++), VB.NET (via Visual Basic .NET), C# (via Visual C#), and F# (as of Visual Studio 2010^[6]). Support for other languages such as M, Python, and Ruby among others is available via language services installed separately. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. Individual language-specific versions of Visual Studio also exist which provide more limited language services to the user: Microsoft Visual Basic, Visual J#, Visual C#, and Visual C++.

WINDOWS FORMS DESIGNER

The Windows Forms designer is used to build GUI applications using Windows Forms.

Windows Forms Designer: The Windows Forms designer is used to build GUI applications using Windows Forms. Layout can be controlled by housing the controls inside other containers or locking them to the side of the form. Controls that display data (like textbox, list box, grid view, etc.) can be bound to data sources like databases or queries. Data-bound controls can be created by dragging items from the Data Sources window onto a design surface.^[30] The UI is linked with code using an event-driven programming model. The designer generates either C# or VB.NET code for the application.

Dot-NET okvir

04

DOT-NET OKVIR

It includes a large library and provides language interoperability.

U Wikipediji se može naći opis .NET okvira. Takodje veza izmedju .NET i Visual Studio-a. Evo tabele gde se vidi koje verzije Visual Studio-a podržavaju koje verzije .NET okvira:

	Podržana .NET verzija
Visual Studio 2008	2.0, 3.0, 3.5
Visual Studio 2010	2.0, 3.0, 3.5, 4.0
Visual Studio 2012	2.0, 3.0, 3.5, 4.0, 4.5

Okvir .NET podržava C#, C++, J#, itd.

From Wikipedia, the free encyclopedia:

The **.NET Framework** (pronounced *dot net*) is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large library and provides language interoperability (each language can use code written in other languages) across several programming languages. Programs written for the .NET Framework execute in a software environment (as contrasted to hardware environment), known as the Common Language Runtime (CLR), an application virtual machine that provides services such as security, memory management, and exception handling. The class library and the CLR together constitute the .NET Framework.

The .NET Framework's Base Class Library provides user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. Programmers produce software by combining their own source code with the .NET Framework and other libraries. The .NET Framework is intended to be used by most new applications created for the Windows platform. Microsoft also produces an integrated development environment largely for .NET software called Visual Studio.

Visual Studio IDE

05

VISUAL STUDIO IDE

Učenje kako koristiti IDE, zahteva puno vremena, slično koliko i učenje kako koristiti C#.

Okruženje Visual Studio .NET predstavlja u stvari jednu aplikaciju -tzv. **Integrated Development Environment**, IDE, tj. Integrisano razvojno okruženje. Šta je IDE, tj. kako ga možemo opisati? Dakle, IDE, to je razvojno okruženje koje sadrži

-editor za pisanje programskog koda,

-i seriju integrisanih alata,

npr. alati za debugovanje, zatim, alati za pridruživanje podataka programu tj. aplikaciji, pa alati za organizovanje 'projekata' tj. programa, itd. Učenje kako koristiti IDE, zahteva puno vremena, slično koliko i učenje kako koristiti C#. U stvari, aktiviranjem Visual Studio .NET, može se postaviti na početak tj. na prvu stranu IDE. Prva strana IDE sastoji se od

-centralne radne površine tj. centralnog prozora, a na vrhu se nalaze **menu-bar** i **toolbars**,

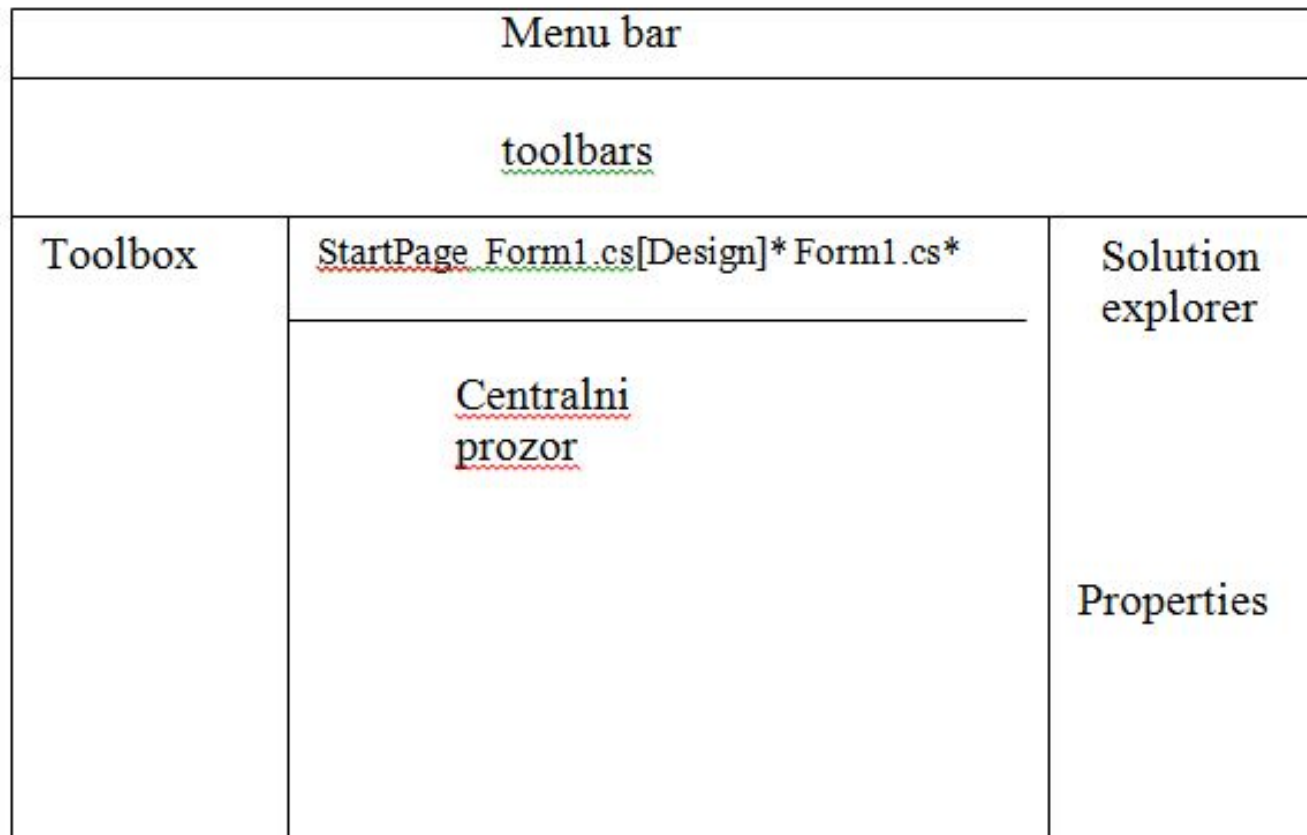
- a sa leve i desne strane **tool windows**

Tool window, tj. alatni prozor, može se skratiti ili sakriti kompletno.

Konačno, centralni prozor ima opcije tj. kontrolne t dugmiće na samom vrhu, i to, prvo početna strana tj. dugme **StartPage** (započinjanje novog projekta), zatim kodirajući editor, **code editor**, j. text editor (dugme **Form1.cs***), pa konačno , i **Form editor** tj. (dugme **Form1.cs[Design]***). Desni prozor je **Properties-prozor**, a levi prozor je **Toolbox-prozor**. Na desnoj strani se takodje nalazi prozor **Solution explorer**. Dole je skicirano kako može da izgleda prva strana IDE. Medjutim, postoji puno različitih načina da se formira prva strana IDE. Takodje, postoje različite verzije Visual C# I Visual Studio-a. Tako da prva satrana IDE može da izgleda na više različitih načina, i slika na Vašem ekranu kada budete koristili Visual Studio može da izgleda različito. Ovo je posle dole ilustrovano pomoću *screenshot*-ova. Screenshot daje detaljan izgled stranice IDE, ali je ponekad korisno koristiti skice umesto *screenshot*-ova radi bolje preglednosti, jer detaljan screenshot sadrži puno detalja koji nisu uvek od interesa. Takodje, *screenshot* može da zauzima značajno više prostora od skice.

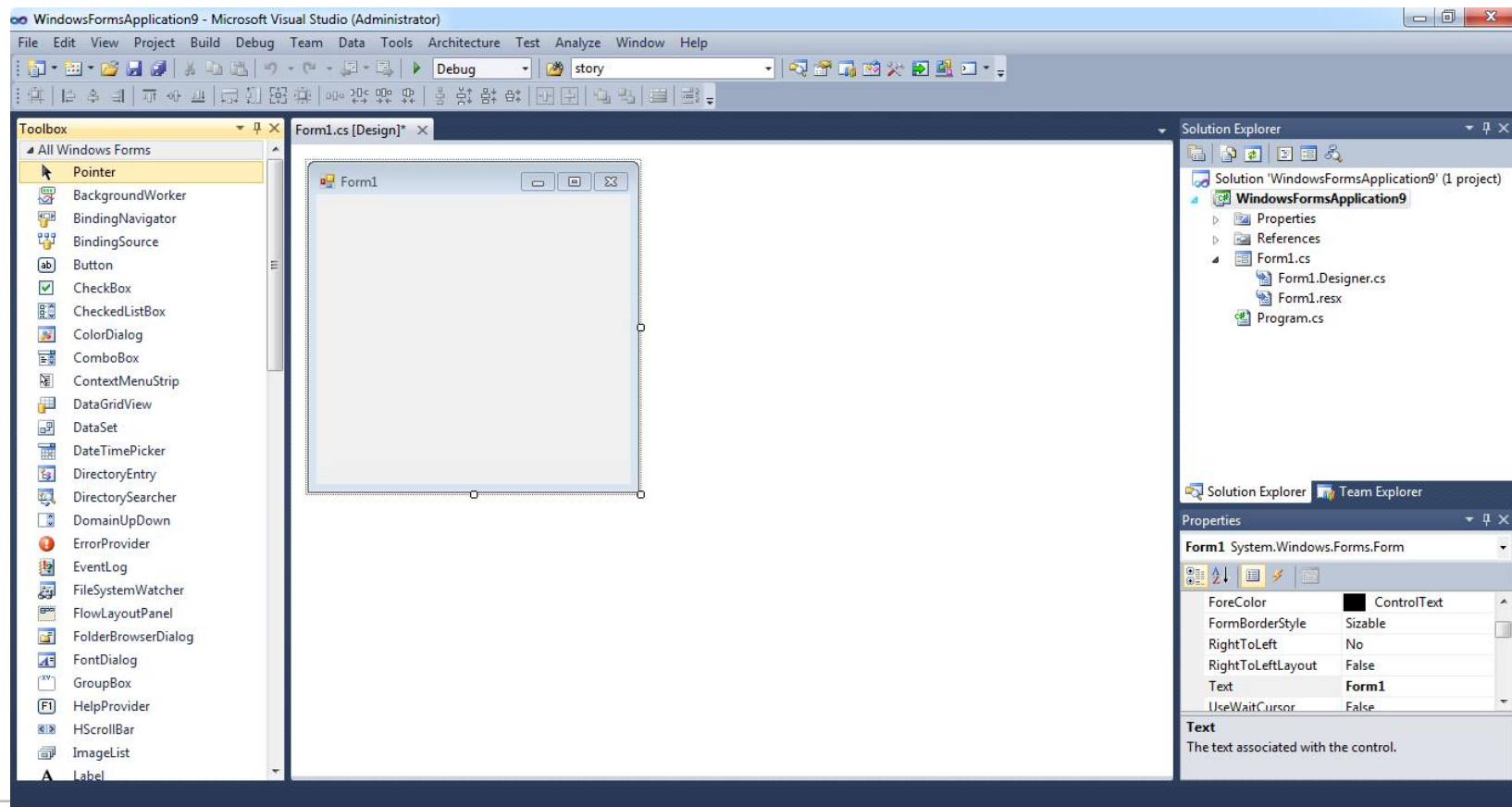
SKICA PRVE STRANE IDE

Slika: Skica prve strane IDE



SCREENSHOT PRVE STRANE IDE SA NESAKRIVENIM TOOLBOX I PROPERTIES EDITOR

Slika: Screenshot prve strane IDE (Visual Studio 2010) sa nesakrivenim Toolbox i Properties Editor



FORMA-DIZAJNER

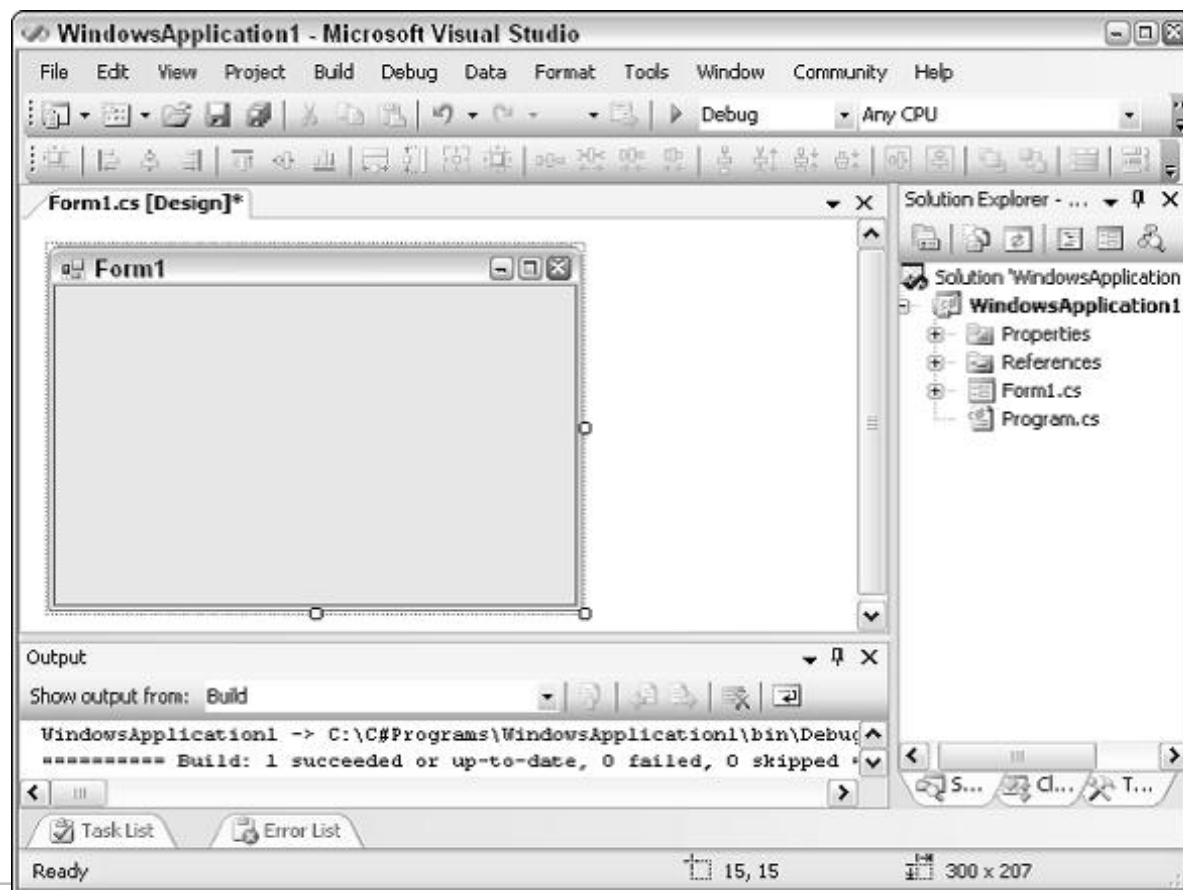
Forma-dizajner je moćan grafički alat u okviru Visual-studio-a, koji omogućuje lako i brzo postavljanje pojedinih grafičkih elemenata na „formu“.

Za *Visual Basic* 6.0 programere, *form-editor* tj. *Form Designer* u okviru Visual Studio-a deluje vrlo poznato. Medjutim, .NET Forms Designer koji koristi Visual C#, je mnogo moćniji od onog Form-dizajnera u Visual Basic 6.0. Naime, .NET i C# I *Visual Basic* .NET, svi koriste .NET biblioteku podprograma, koja je vrlo moćna i konzistentna. I .NET podržava distribuirane programe za mrežu, a isto tako podržava programe koji koriste više jezika. I još jedna velika prednost, .NET vam omogućuje da modifikujete izvorni kod koji vam on proizvodi za vas.

Forma-dizajner je moćan grafički alat u okviru Visual-studio-a, koji omogućuje lako i brzo postavljanje pojedinih grafičkih elemenata na „formu“, npr. element „dugme“, „tekst-boks“, itd.

SCREENSHOT PRVE STRANE IDE (VISUAL STUDIO 2005) SA FORMA-DIZAJNEROM I SAKRIVENIM TOOLBOX

Nasluci je screenshot prve strane IDE (Visual Studio 2005) sa Forma-dizajnerom i sakrivenim Toolbox-om.



Slika 3: IDE

Prva GUI aplikacija

06

PRVA GUI APLIKACIJA

Objasnićemo ovde kako da napravimo jednu jednostavnu GUI (grafičku) aplikaciju.

Da bi započeli sa primenom Visual C#, objasnićemo ovde kako da napravimo jednu jednostavnu GUI (grafičku) aplikaciju u četiri koraka:

Korak 1:

Startovati okruženje Visual C# ili Visual Studio, i onda kliknuti na

- Ikonu **New Project** tj. prvu ikonu s leva u gornjem **Toolbar**,
- ili izabrati **New-Project** u **File menu**.

Korak 2:

U Dijalogu **New Project**,

- kliknuti na **Windows Application**,
- i onda otkucati npr. `FirstWindowsApp` umesto `WindowsApplication1` u Name box-u, ukoliko želimo da damo novo ime našoj prvoj Windows-aplikaciji,
- i onda kliknuti OK, da bi se startovao 'projekt', *project*.

Korak 3:

Jedan prozor se otvara, koji se zove `Form1`. U prozoru **Properties**, osigurati da se prikazuje `Form1` na vrhu. Onda,

- kliknuti dugme **AZ**, da bi se izvršilo alfabetsko sortiranje u prozoru **Properties**,
- onda naći liniju **Text**, i otkucati ime koje želimo da damo formi umesto `Form1`, npr. `MyForm1` ili `MyLabel1` ili `MyLabel`, ili neko drugo ime, u zavisnosti kakvu aplikaciju pravite i kako želite da je nazovete.

IZVRŠAVANJE PROGRAMA

Ova prva aplikacija samo proizvodi jedan prozor na ekranu, koji se može pomerati, minimizirati, menjati mu veličina, i konačno napustiti aplikacija, zatvaranjem prozora.

Korak 4:

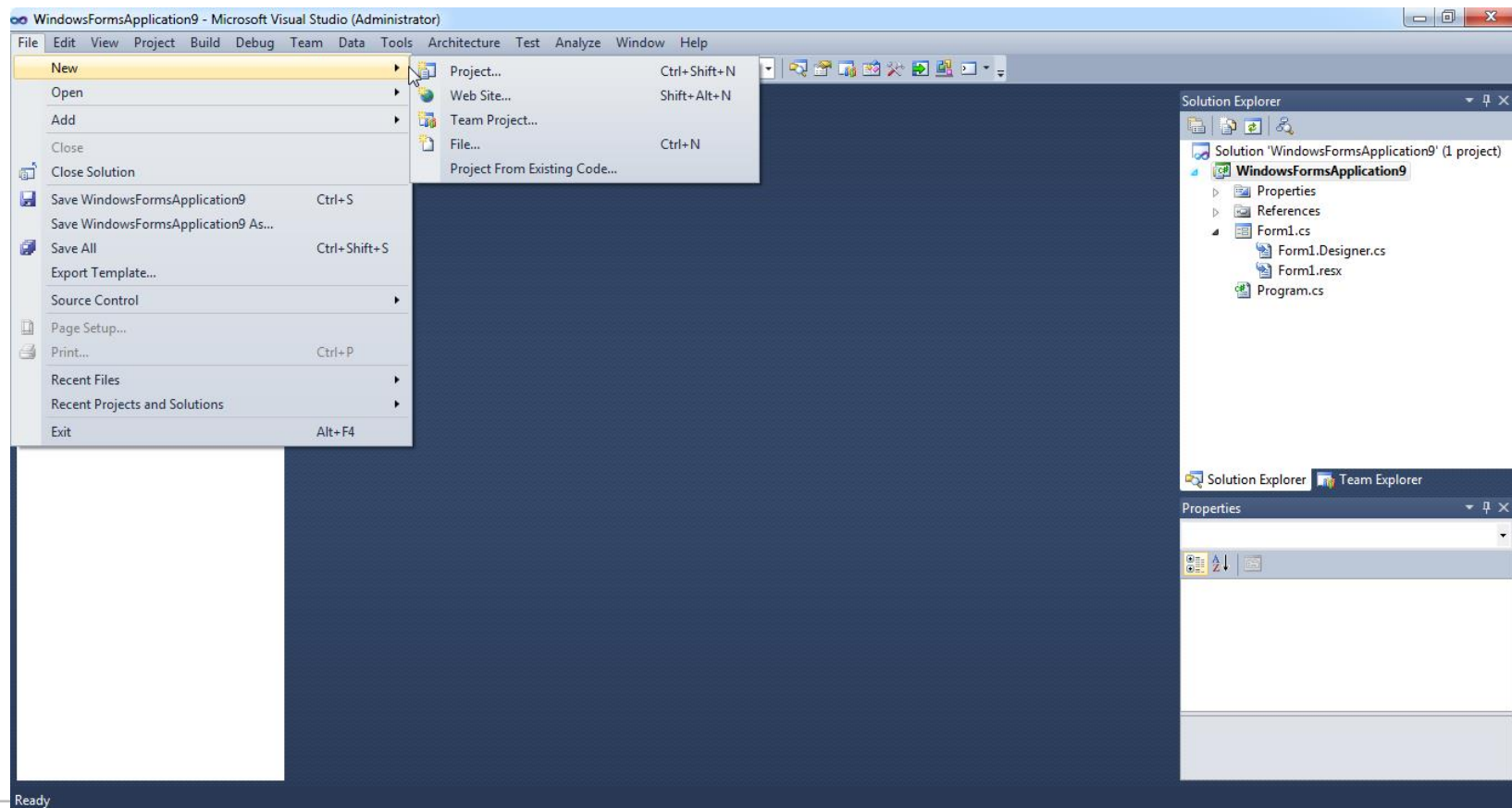
-U **Toolbar**, kliknuti dugme Start, označenim sa malom strelicom na desno (nalazi se neposredno ispred **Debug**), u cilju izvršavanja programa.

-Ova prva aplikacija samo proizvodi jedan prozor na ekranu, koji se može pomerati, minimizirati, menjati mu veličina, i konačno napustiti aplikacija, zatvaranjem prozora.

-Ovo je krajnje jednostavan program, sa njim se ne postiže mnogo, ali to je prva vaša Windows- GUI aplikacija („**grafička aplikacija**“) u Visual C#.

KORAK 1

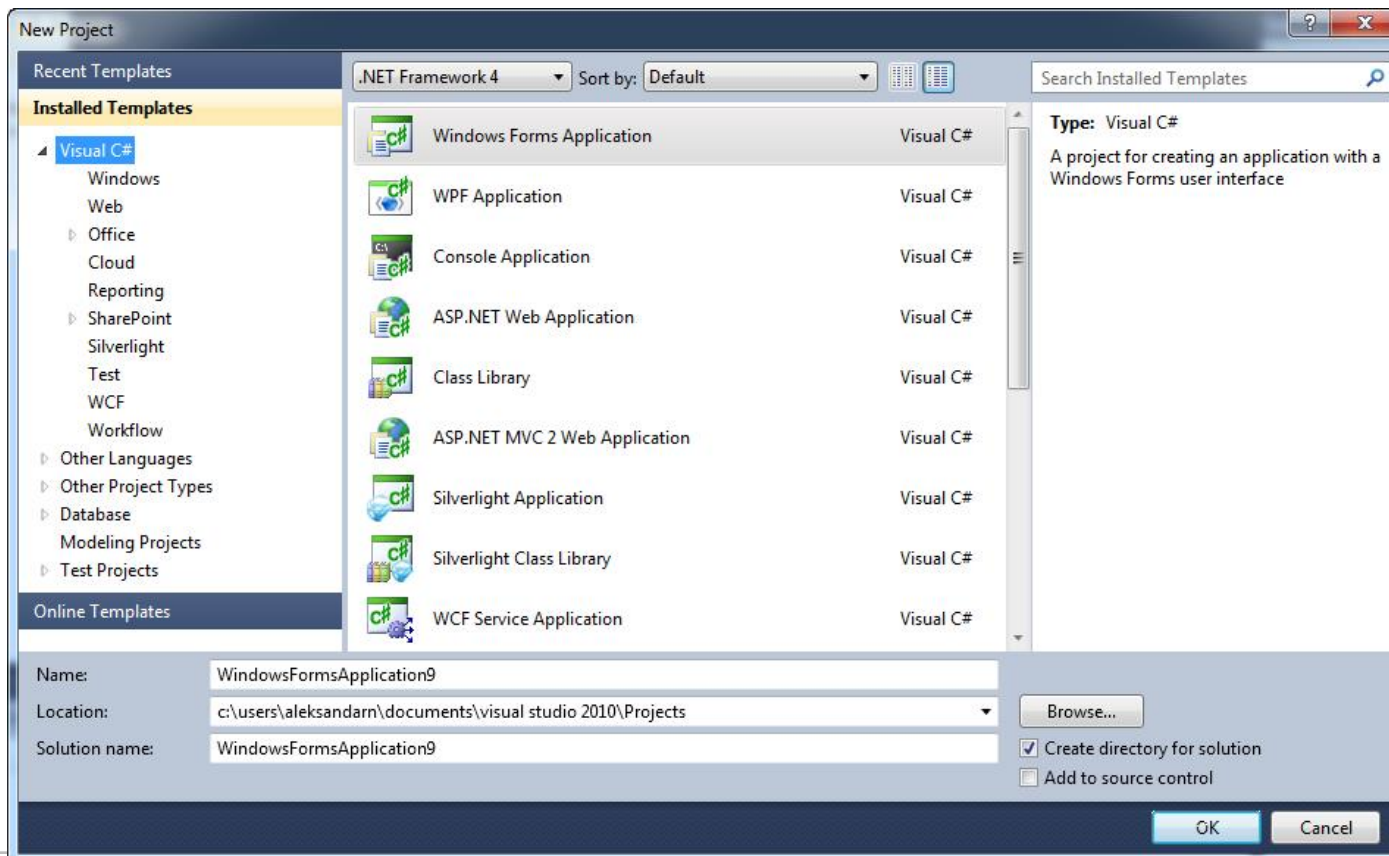
Na slici je ilustrovan Korak 1 izgradnje GUI aplikacije.



Slika 1: Korak 1

NEW PROJECT DIALOG

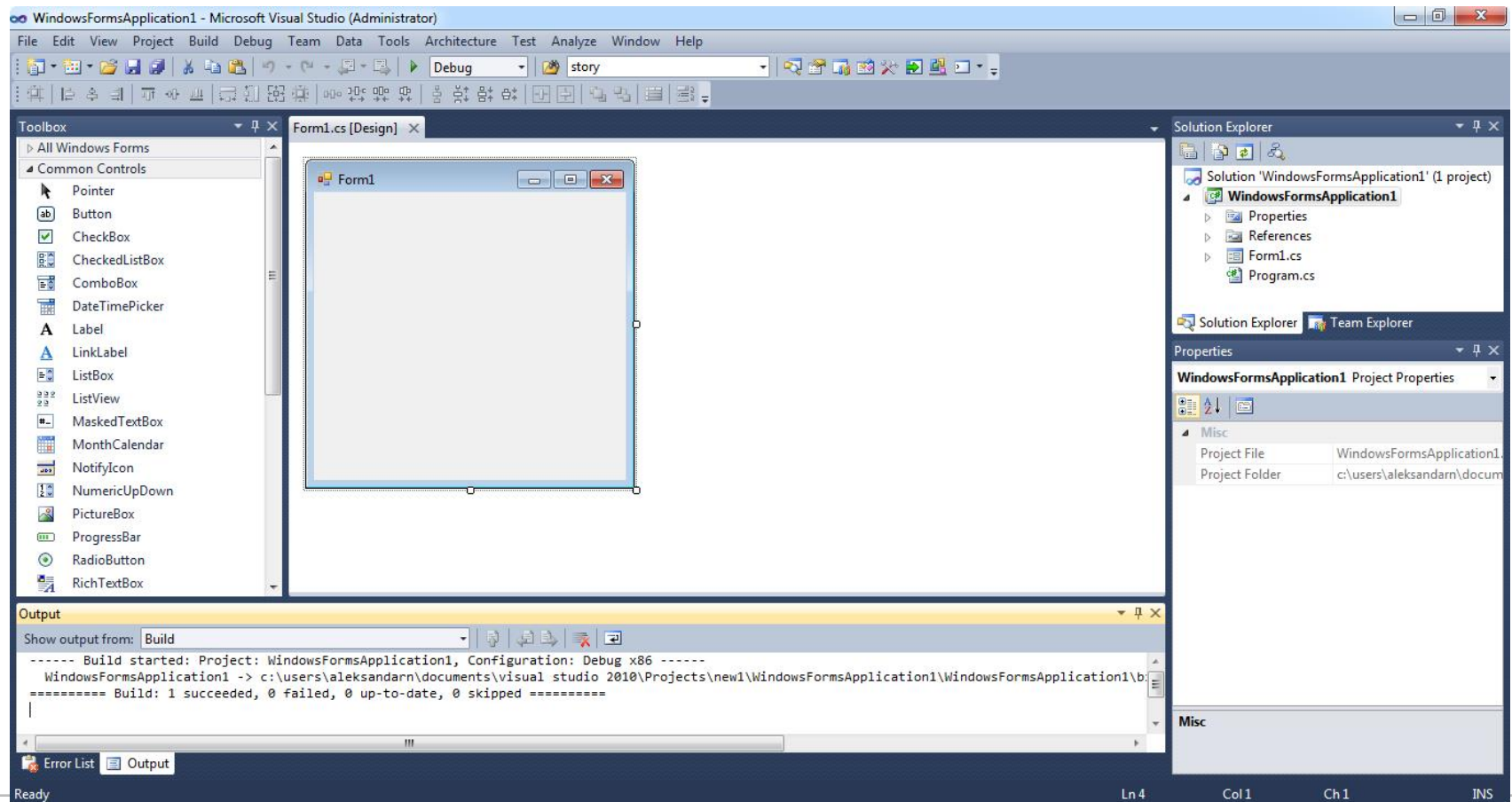
Na slici je New Project dialog.



Slika 2: New Project dialog

STARTOVANJE PROJEKTA WINDOWSAPPLICATION1

Slika prikazuje startovanje projekta WindowsApplication1 (startovanje prve Windows-aplikacije).



„FORMA“ NASLOVLJENA SA FORM1

Na slici vidimo „formu“ naslovljena sa Form1



Slika 4: Forma Form1

Toolbox

07

TOOLBOX

Toolbox je prozor u Visual C# koji sadrži pojedine komponente neke grafičke aplikacije.

Toolbox je prozor u Visual C# koji sadrži pojedine komponente neke grafičke aplikacije, npr. „dugme“, ili „tekstboks“, ili „nalepnica“, itd. Dole na slici je prikazan kako izgleda Toolbox. I njene komponente, tj. objekti u Toolbox-u (tzv. „kontrola“):

Label,

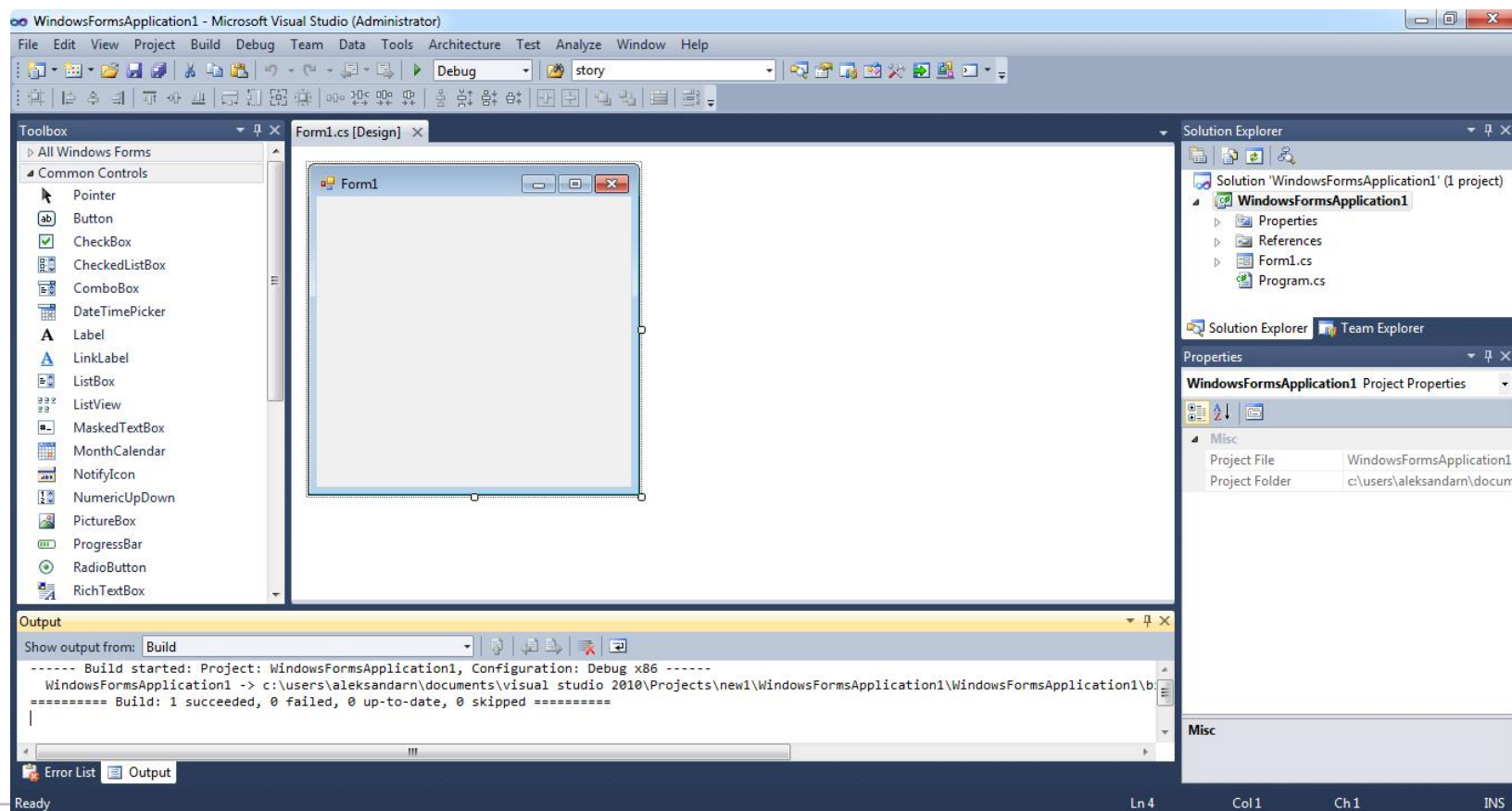
Button,

TextBox,

itd.

VISUAL STUDIO TOOLBOX SADRŽI ČITAV NIZ „KONTROLA“

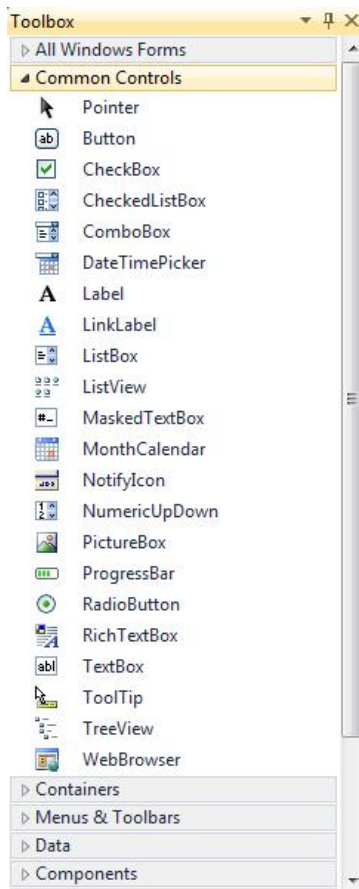
Slika: Visual Studio Toolbox sadrži čitav niz „kontrola“ („kontrole“ su objekti koje se mogu postaviti na „formu“: npr. Button, CheckBox, TextBox, Label, itd.)



Slika 1: Toolbox

TOOLBOX WINDOW

Slika: Toolbox window



TOOLBOX - PRIMER

Ovde ćemo prvo posmatrati primer gde će se koristiti objekat Label uToolbox-u.

Ovde ćemo prvo posmatrati primer gde će se koristiti objekat **Label** uToolbox-u da se doda neki tekst na „formu“:

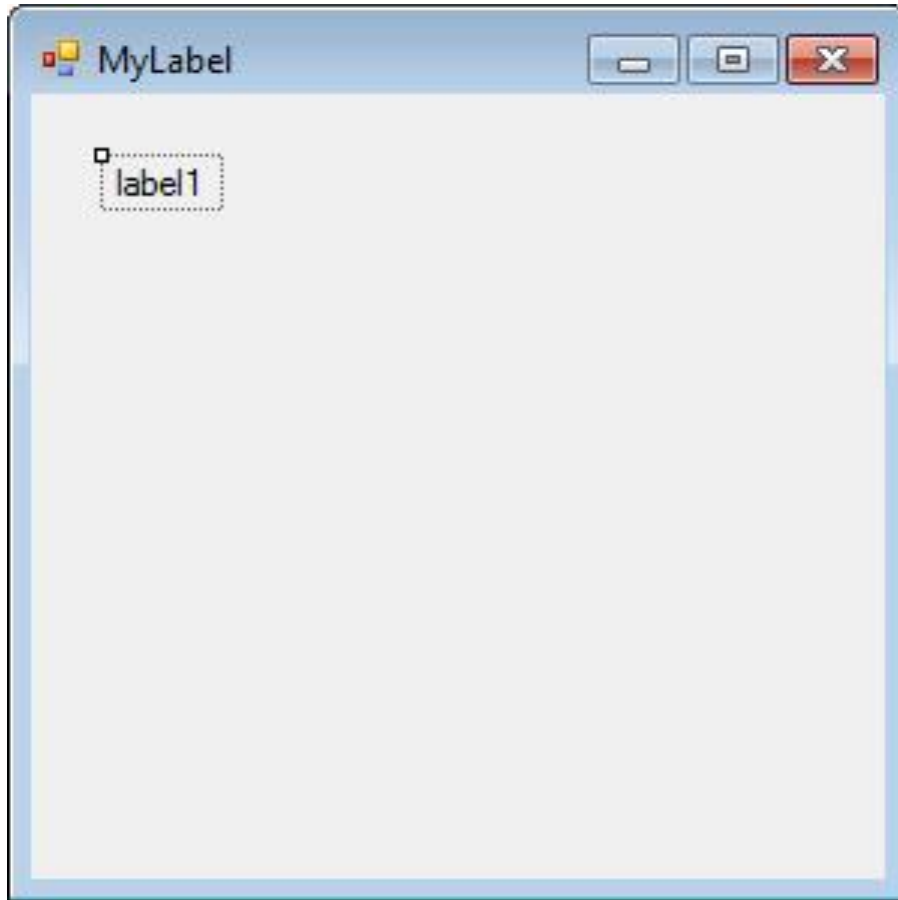
Korak 1 Klikne se na ikonu Label(tj. „Nalepnica“).

Korak 2 Pomoću *Mouse pointer-a*, Miš-pokazivača, i pritiskanja levog dugmeta Miša, može se na Form-u postaviti levi gornji ugao Nalepnice, tj. objekta **Label**. Držeći pritisnuto dugme, poveća se nalepnica po želji, i onda se otpusti dugme. Ako treba, objekat **Label** se može pomeriti, ili promeniti njena veličina. Nalepnica dobija automatski predefinisano ime, *default name*, „label1“.

Na ovaj način smo na „formu“ (Visual C# formu tj. „Windows formu“) postavili na željenom mestu tekst tj. „nalepnicu“ „label1“ željene veličine. Ali ovo je najjednostavniji slučaj korišćenja objekata u Toolbox-u. Medjutim, mogu se formirati mnogo složenije C# - forme od ove. Npr. dole je prikazana „forma“ sa dva Textbox-a (dva „tekstboks“) i jednim Button (jednim „dugmetom“). Kreiranje ove složenije forme je potpuno slično sa formiranjem prethodne jednostavne forme, s tim što se postavlja objekat po objekat na „formu“, npr. prvo dugme pa onda prvi pa onda drugi „tekstboks“. Dole je takodje dat izgled Toolbox-a u okviru Visual 2010 Express.

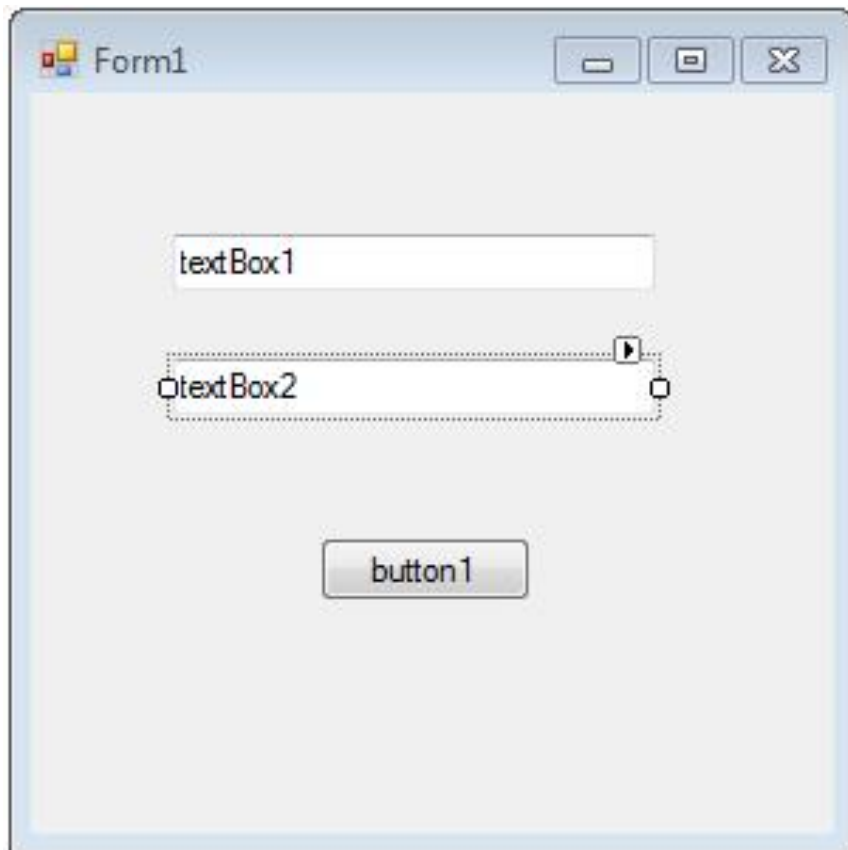
VISUAL C# FORMA SA „NALEPNICOM“

Slika: Visual C# forma naslovljena MyLabel sa „nalepnicom“ label1



FORMA SA DVA „TEKSTBOKSA“ I JEDNIM „DUGMETOM“

Na slici vidimo formu sa imenom Form1 i sa dva „tekstboks“ i jednim „dugmetom“.



Property-editor

08

PROPERTY-EDITOR

Mogu se menjati karakteristike (properties) elemenata iz Toolbox-a, pomoću Properties Window.

Property Editor tj. *Properties Window* (prozor *Properties*) prikazuje liste karakteristika, *properties*, za pojedine objekte u *Toolbox*-u, tj. za pojedine „kontrolne”, gde su objekti *Label*, *Button*, *TextBox*, itd, izlistani u *Toolbox*. Mogu se menjati ove karakteristike, *properties*, pomoću *Properties Window*. Karakteristike su npr. *Font*, *Color*, *Size*, *Text*, itd. Dole je na crtežu prikazan *Properties Window*.

Upotreba *Properties Window* je brza i jednostavna, mada zahteva prvo upoznavanje sa mogućnostima koje pruža. Ali, posle ovog upoznavanja, jednostavno je koristiti *Properties Window*. Npr

Korak 1:

za objekat *Label* nazvan „label1“, čije kreiranje smo opisali u prethodnom poglavlju, može se promeniti ime ume-asto *Label1* u *MyLabel*, tako što će se kliknuti *Text* u *Properties Window*, i onda ukucati:

MyLabel

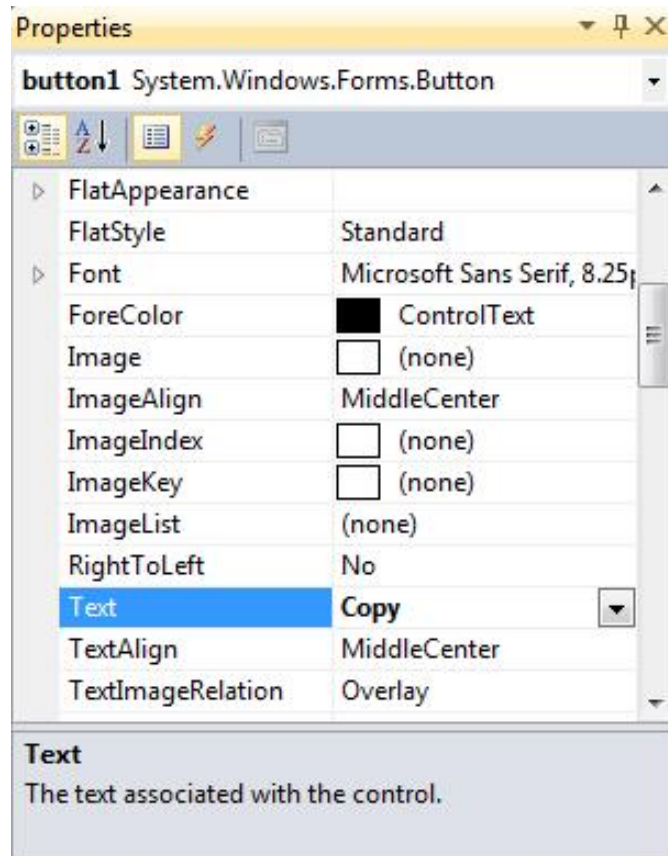
Korak 2:.

Takodje, ako se klikne na *Font*, može se promeniti veličina slova, npr. uvećati veličina slova.

Dalje, za „formu“ *Form1* sa dva *Textbox*-a i jednim *Button*, može se na potpuno sličan način promeniti ime da umesto „button1“ bude npr. „*Copy*“. Ova modifikovana „forma“ prikazana je takodje na odgovarajućem crtežu.

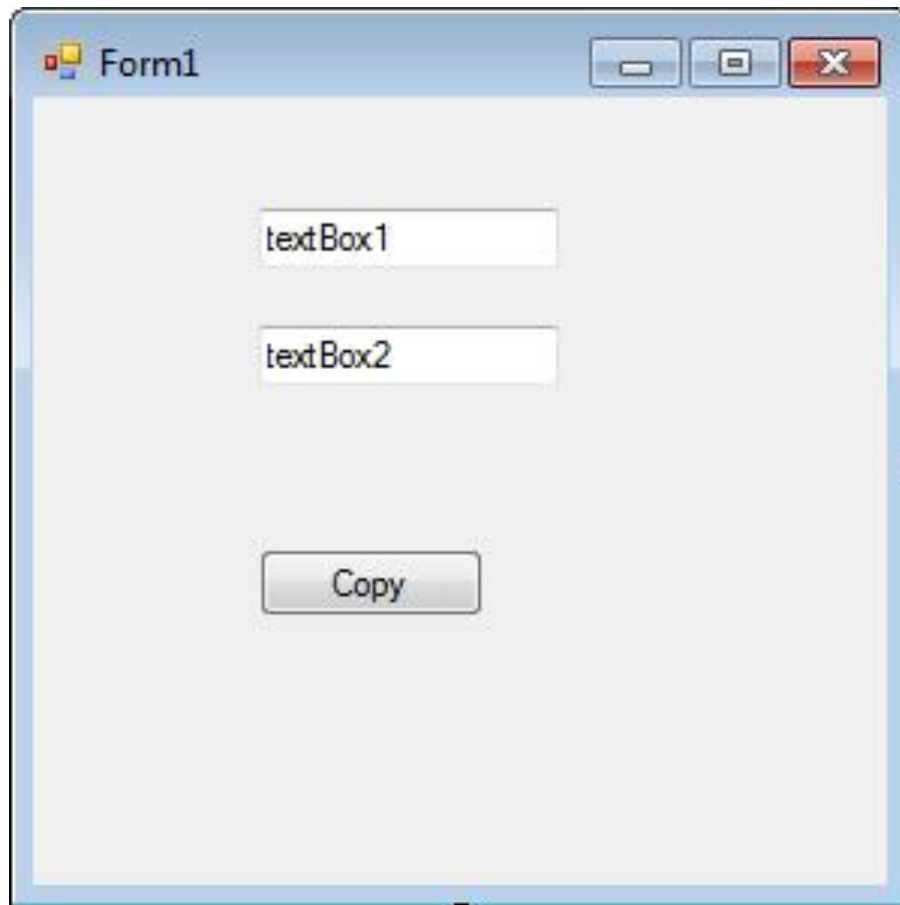
PROPERTIES WINDOW

Slika prikazuje Properties Window.



FORMA FORM1 SA „DUGMETOM“ NAZVANIM „COPY“.

Slika prikazuje formu Form1 sa „dugmetom“ nazvanim „Copy“.



Uvod u C# - Vežba

09

INSTALACIJA VISUAL STUDIO 2010 IDE

Softverski zahtevi:

Visual Studio 2010 može biti instaliran na sledećim operativnim sistemima:

- Windows XP (x86) - Service Pack 3 – sve edicije osim Starter edicije
- Windows Vista (x86 & x64) - Service Pack 2 - sve edicije osim Starter edicije
- Windows 7 (x86 & x64)
- Windows Server 2003 (x86 & x64) - Service Pack 2
- Windows Server 2003 R2 (x86 & x64)
- Windows Server 2008 (x86 & x64) - Service Pack 2
- Windows Server 2008 R2 (x64) Podržane arhitekture:
 - 32-Bit (x86)
 - 64-Bit (x64) Hardverski zahtevi:

- 1.6GHz ili brži procesor
- 1 GB (32 Bit) ili 2 GB (64 Bit) RAM
- Ukoliko se razvojno okruženje instalira na virtuelnu mašinu potrebno je dodati najmanje 512 MB virtuelnoj mašini
- 3GB prostora na hard disku
- 5400 RPM hard disk
- DirectX 9 video card @1024 x 768 ili većoj rezoluciji
- DVD-ROM Drive ukoliko se ne instalira sa download-ovanog fajla.

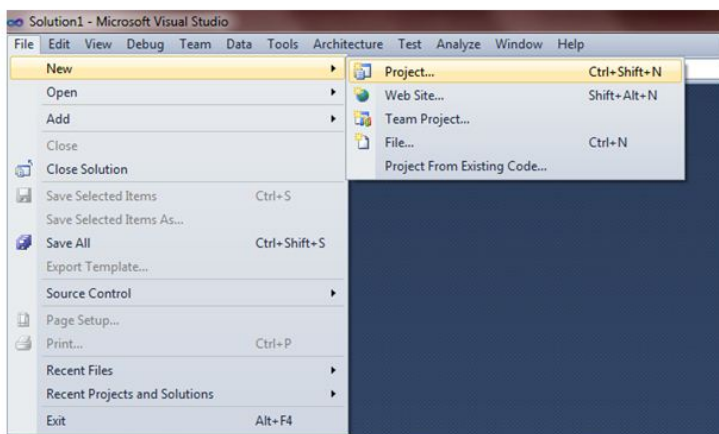
UPOZNAVANJE SA OSNOVNIM ALATIMA U VISUAL STUDIO 2010 IDE

Visual Studio 2010 IDE (Integrated Development Environment) je okruženje za razvoj i testiranje aplikacija.

Visual Studio 2010 IDE (Integrated Development Environment) je okruženje za razvoj i testiranje aplikacija. Postoji veliki broj tipova aplikacija koje VS podržava, te ne postoji jedinstveni šablon za sve.

Kreiranje novog projekta je jednostavno i može se uraditi i po intuiciji:

1. Izaberite File->New->Project...



Slika-7: Kreiranje projekta

2. Na Project dialog prozoru izaberite Visual C#.

3. Na listi Templates izaberite Windows Form Application.

4. U tekst polje Name unesite ime vašeg projekta ili možete prihvatiti već ponudeno ime, na primer WinForms1

5. U polju location definišite lokaciju ili prihvatite već ponuđenu, na primer c:\users\bosko\documents\visual studio 2010\Projects.

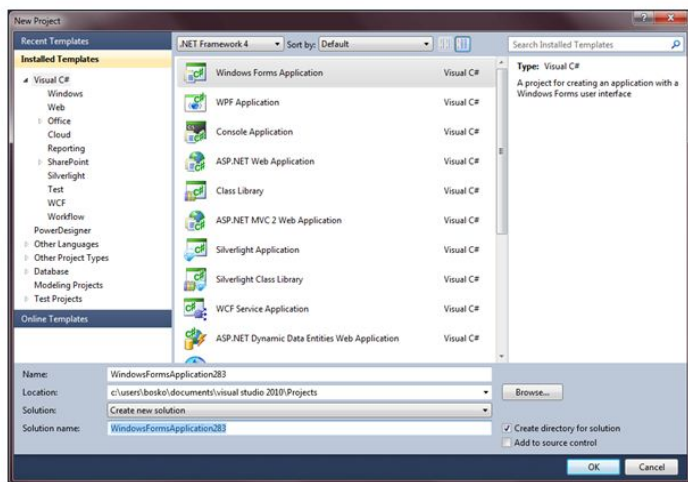
6. U listi solution odaberite opciju create new solution.

7. U teks polju solution name definišite ime za solution ili prihvatite ponuđeno, na primer WindowsFormsApplication283.

UPOZNAVANJE SA OSNOVNIM ALATIMA U VISUAL STUDIO 2010 IDE DRUGI DEO

Visual Studio 2010 IDE (Integrated Development Environment) je okruženje za razvoj i testiranje aplikacija.

Slika:



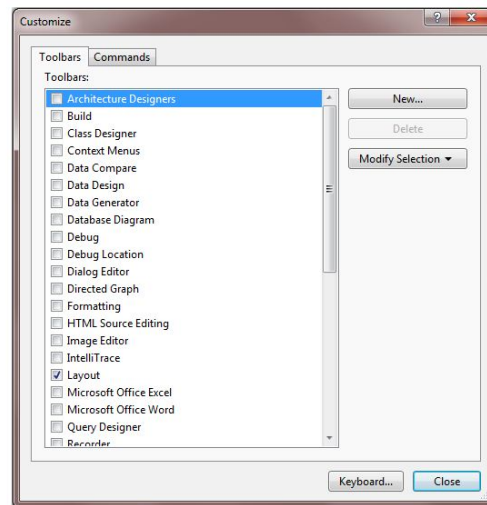
Slika-8: Kreiranje windows forms aplikacije

8. Klikom na dugme OK izvršićete potvrđivanje i kreiranje nove aplikacije sa podrazumevanom formom.

Naravno, toolbar je konfigurabilan i postoji više tipova ove kontrole. Desnim klikom miša možemo videti spisak sa aktivnim objektima toolbara.

Iako je uobičajeno da bar bude uz neku od ivica glavnog prozora, bitno je napomenuti da je moguće da toolbar stoji i samostalno.

U prethodnih nekoliko rečenica, naglasili smo da je toolbar konfigurabilan. Ako sa glavnog menija odaberete Tools->Customize možete podešavati alatke koje ce biti na radnoj površini ili dodavati nove. Ovo podešavanje se izvodi na tabu Toolbars.



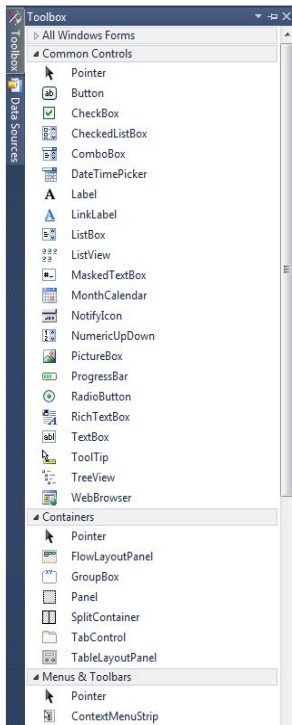
Slika-10: Toolbar

TOOLBOX

Uz levu ivicu ekrana nalazi se bar Toolbox (kutija sa alatima dizajnera) koji je vertikalno orijentisan. U objektu toolbox razvrstane su kontrole po kategorijama.

Uz levu ivicu ekrana nalazi se bar Toolbox (kutija sa alatima dizajnera) koji je vertikalno orijentisan. U objektu toolbox razvrstane su kontrole po kategorijama.

Na naslovnoj liniji postoji dugme pribadača kojim se menja ponašanje ove kontrole. Pritiskom na pribadaču bar može biti fiksiran za ivicu glavnog prozora. Ideja ovog rešenja je povećanje korisne površine ekrana kada se kutija sa alatima ne koristi.



Slika-11: Toolbox

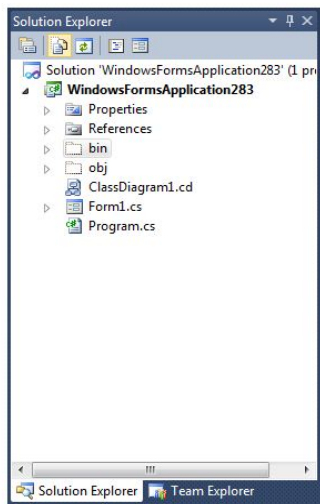
SOLUTION EXPLORER I PROPERTIES

Na desnoj strani ekran je podeljen u dve sekcije.

Na desnoj strani ekran je podeljen u dve sekcije. Gornja sekcija uključuje pet različitih tabovakoji se otvaraju u glavnom ekranu od kojih pojedini predstavljaju

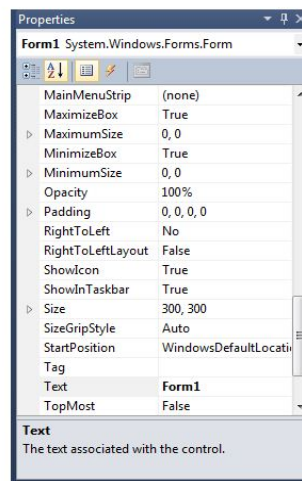
novinu u VS2010. Vaš zadatak je da pregledate sva dešavanja prilikom izbora

tabova I upoznate se sa infomacijama koje dobijate, jer će Vam to u narednom period biti od velike koristi.



Slika-12: Solution explorer

U donjoj sekciji na desnoj strani nalazi se prozor koji prikazuje svojstva stavki koje su selektovane na gornjoj.



Slika-13: Svojstva

Izgled radne površine je kao što smo i ranije rekli potpuno proizvoljan i moguće ga je u potpunosti prilagoditi sopstvenim potrebama. Opis koji je dat važi za inicijalni izgled IDE okruženja.

Upotreba ostalih alata biće pikazana kasnije na konkretnim primerima.

OSNOVNO O WINDOWS KONTROLAMA

Danas je potpuno nezamislivo da imamo neku aplikaciju koja nema nikakvu interakciju sa korisnikom.

Danas je potpuno nezamislivo da imamo neku aplikaciju koja nema nikakvu interakciju sa korisnikom.

Da bismo kreirali svoju prvu aplikaciju koja će imati korisnički interfejs i koja će u startu moći nešto da prikaže na ekran, a da to nije konzolna aplikacija dovoljno je da kreiramo novi projekat prema opisu iz prve vežbe i sa malo rada postignemo željeni cilj. Upravo u tome i jesu čari vizuelnog programiranja – nije potrebno mnogo za inicijalne stvari u UI, ali daleko od toga da pravljanje funkcionalnog UI jednostavno. Time se nemojte zavaravati!

Kada kreirate neku aplikaciju koja sadrži interakciju sa korisnikom moraćete da upotrebite objekte kojima obezbeđujete razmenu informacija između aplikacije i korisnika. Ti objekti će prikazivati podatke korisniku ili će prenositi podatke od korisnika do vašeg programa. Oni se nazivaju kontrole.

Osobine Windows kontrola

Pre nego što se upustimo u dalje objašnjavanje kontrola, jedna važna napomena: pomen kontrole u ovoj vežbi odnosi se isključivo na kategoriju Windows kontrola.

Kontrole su prilično raznovrsne (ima mnogo različitih tipova) i one se postavljaju na formu po svojim svojstvima i po svojoj nameni. Istovremeno, iako su raznovrsne, postoje i mnoge sličnosti među njima.

Zajednicko svojstvo za sve kontrole je da su po svojoj prirodi takode i prozori, odnosno imaju većinu sličnih svojstava. Njihov zadatak je da prihvataju određene događaje na interfejsu i omoguće dalji tok. Događaji su kao takvi, specifični samo za određenu kontrolu, ili su pak zajednički.

Posebno je bitno, kao što ste i u ranijim programerskim predmetima do sada imali, korišćenje specifične nomenklature prilikom imenovanja. Ime treba da bude opisno tj. Da što više opisuje ono što predstavlja. Ukoliko je za to potrebno više od jedne reči C# nomenklaturi takva imena počinju malim slovom a svaka nova reč

velikim. Na primer: firstName, lastName, phoneNumber...

Razlozi za uvođenje nomenklature su jasni: održavanje koda, rad u timu, itd...

RELACIJA RODITELJ - DETE

Postoji nekoliko razlicitih relacija izmedu prozora koji cine jednu ili vise aplikacija. Za nas je za sada interesantan odnos je roditelj-dete tzv. parent-child relationship.

Postoji nekoliko razlicitih relacija izmedu prozora koji cine jednu ili vise aplikacija. Za nas je za sada interesantan odnos je roditelj-dete tzv. parent-child relationship.

Na primer, kontrola koju ste postavili na neku formu predstavlja prozor ciji roditelj je drugi

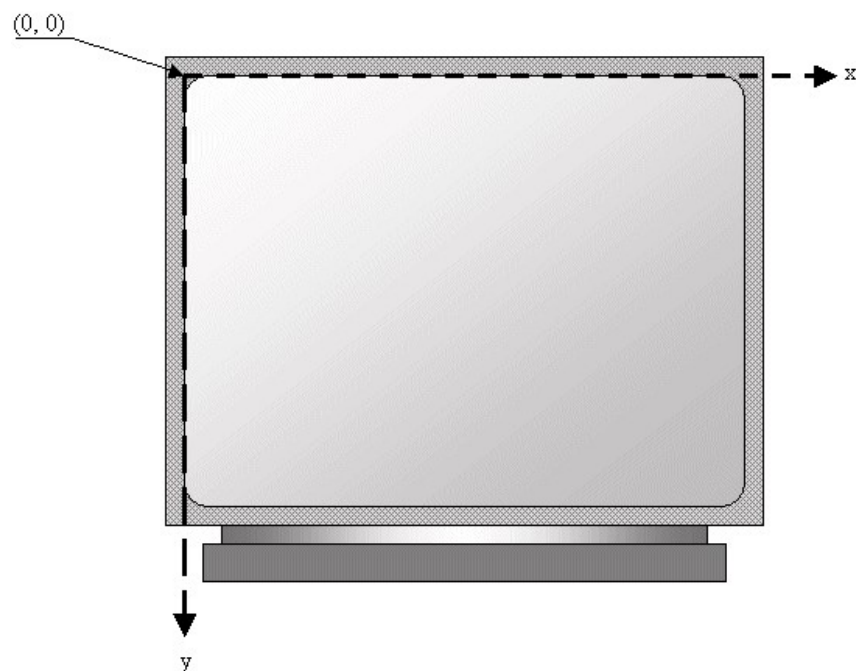
prozor tj. Forma. Ovo je podrazumevana relacija kada se kontrole postavljaju i manifestuje se "lepljenjem" kontrole za tu formu.

Pomeranjem forme pomeraju se sve kontrole na njoj, tj. sva deca prozori. Ista relacija postoji izmedu formi koje pripadaju nakoj MDI aplikaciji.

KOORDINATNI SISTEM

Objašnjenje svojstva Location

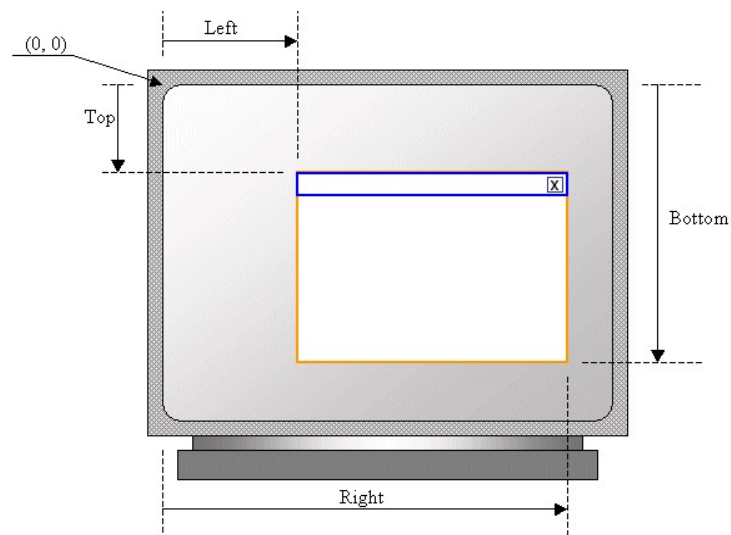
Da bi se bolje razumelo svojstvo Location evo nekih dodatnih objašnjenja u vezi koordinatnog sistema koji se primenjuje kada je reč o postavljanju objekata na ekranu odnosima između prozora u Windows operativnom sistemu.



Slika-14: Prozor

Pozicija nekog prozora čiji je roditelj desktop racuna se od gornjeg levog ugla ekrana, kao što je to prikazano na slici. Ose koordinatnog sistema orijentisane su tako da rastu od leve strane nadesno za x osu i od vrha nadole za y osu.

Rastojanje od koordinatnog pocetka tacke do leve ivice je svojstvo "Left". Adekvatna ostala svojstva zovu se "Right", "Top", "Bottom".

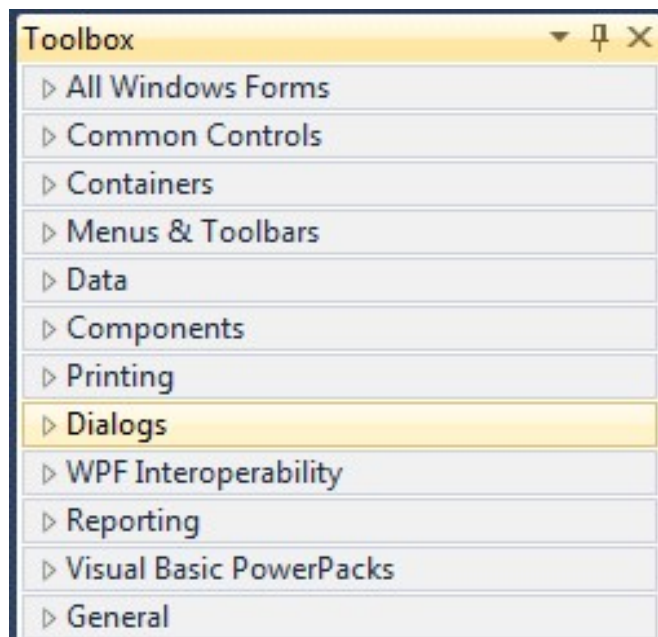


Slika-15: Primer koordinate

POSTAVLJANJE KONTROLE NA FORMU

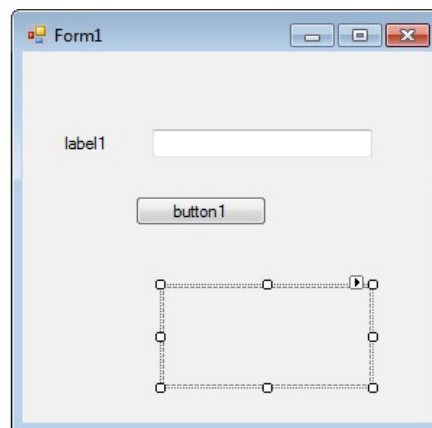
Izbor kontrole se vrši sa toolbox palete na levoj strani ekrana. Kategorije na paleti toolbox obezbeđuju nam brži izbor tražene kontrole

Izbor kontrole se vrši sa toolbox palete na levoj strani ekrana. Kategorije na paleti toolbox obezbeđuju nam brži izbor tražene kontrole



Slika-16: Toolbox

Nakon selektovanja mišem, kontrola se uzima i otpušta (drag and drop) na tekući prozor, odnosno formu. Kontrola će biti pozicionirana na kursoru miša.



Slika-17: Forma

Pored ovog načina, postoji još nekoliko načina dodavanja kontrola na formu.

- Dvostruki klik na kontrolu - kontrola će biti automatski dodata formi u gornjem levom uglu;
- Postavljanje većeg broja istih kontrola na formu – pritisnut taster Ctrl i selekcija kontrole levim kliom miša.

SELEKCIJA KONTROLE, PODEŠAVANJE POZICIJE I VELICINE

U cilju promene nekih svojstava, kontrole i njene pozicije, kontrola se mora selektovati. Selekcija kontrole se realizuje levim klikom miša na odabranu kontrolu.

U cilju promene nekih svojstava, kontrole i njene pozicije, kontrola se mora selektovati. Selekcija kontrole se realizuje levim klikom miša na odabranu kontrolu. Oko kontrole se pojavljuje pravougaonik sa 8 malih kvadratića koji olakšavaju promene dimenzija kontrole u određenim pravcima.

U cilju selekcije više od jedne kontrole moraju biti pritisnuti tasteri Shift ili Ctrl. Tasterom Shift omogućena je selekcija kontrola od – do, a tasterom Ctrl je

omogućeno selektovanje kontrola koje se odabiraju. Zatim se odabira jedna po jedna željena kontrola. Poslednja ima pravougaonik sa crnim kvadraticima oko nje.

Drugi način da se selektuje skup kontrola je levim klikom miša na površinu forme i povlačenjem kursora se formira pravougaonik, a sve kontrole u njemu da postaju selektovane.

Pomeranje određene kontrole vršite tako što kursorom selektujete tu kontrolu a zatim dok je levi klik miša pritisnut povlačite kontrolu preko forme. Pošto je pomerite na željenu poziciju otpustite taster miša i kontrola zauzima novu poziciju. Ako ste selektovali više od jedne kontrole na isti način možete pomerati i grupu kontrola

. Drugi način je da selektujete kontrolu a onda koristeći “strelice” na tastaturi izvodite pomeranje kontrole po formi. Korak pomeraja odgovara mreži na formi (grid). Ukoliko želite preciznije pomeranje koristite pritisnut taster Ctrl.

ZAJEDNIČKA SVOJSTVA WINDOWS KONTROLA

Svaka kontrola ima svoja svojstva

Name

Svaka kontrola ima svojstvo Name. Smisao svojstva Name je da se preko njega omogući pristup. Ime mora postojati i ono se automatski dodaje pri kreiranju, tj. pri postavljanju neke kontrole/forme. Na primer, kada se neka aplikacija formira automatski se postavlja prva forma i njoj se dodeljuje ime Form1. Svaka sledeća forma dobija novo ime sa narednim brojem: Form1, Form2....

Isto pravilo važi i za kontrole koje se postavljaju na forme. One dobijaju imena po automatizmu koje im IDE okruženje dodeljuje a bazira se na tipu kontrole. Tako na primer tekst polja dobijaju imena text1, text2...

Location

Drugi primer uobičajenog svojstva svih kontrola je njihova lokacija (Location). Naravno ovo svojstvo je veoma bitno za vizuelni izgled forme, ali postoje takođe i kontrole koje se postavljaju na formu sa namerom da obezbede dodatnu funkcionalnost bez potrebe da budu vidljive. To je izuzetak kada pozicija kontrole nije od značaja. Ovo svojstvo je opisano koordinatama X i Y. Ovo je pozicija gornjeg

levog ugla kontrole u odnosu na prozor na kome se kontrola nalazi. Obratite pažnju da je rec o vrednosnom tipu, jer su X i Y definisani kao celi brojevi) a celi brojevi se realizuju kao strukture.

Size

Kada je rec o grafičkim svojstvima kontrole, osim pozicije, treba spomenuti i njenu veličinu Size. Svi objekti su po prirodi pravougaonog oblika pa se velicina

jednostavno opisuje. Svojstva koja opisuju velicinu su: širina (Width), odnosno

visina (Height). Ukoliko se kontrola ne postavlja programski već u toku dizajna forme, promena veličine se obavlja jednostavnim korišćenjem miša, uz pomoć pravougaonika koji označava selektovanu kontrolu. Na pravougaoniku se mišem pritisne kvadratić zavisno od dimenzije koju želimo da promenimo, a zatim se povlačenjem miša menja dimenzija kontrole.

ZAJEDNIČKA SVOJSTVA WINDOWS KONTROLA DRUGI DEO

Svaka kontrola ima svoja svojstva

Drugi način da se promene parametri kontrole (ne samo veličina) je koristeci Properties prozor. Među svojstvima kontrole lako se prepoznaje svojstvo Size ispred koga se nalazi znak +. Ovaj znak, tipično za Windows, omogućuje ekspanovanje stavke. Ispod ove stavke se nalaze posebno Width i Height.

Naravno, sva svojstva mogu se postavljati i programski. Bilo na samom startu pri kreiranju kontrola/formi, bilo u kodu.

Bounds

Pozicija zajedno sa veličinom smeštene su u jednom svojstvu kontrole koje se naziva Bounds. Ovo svojstvo daje pravougaonik kontrole/forme. U kodu ovim svojstvom lako se vrši repositioniranje i promena veličine kontrole.

Text

Veoma korisno svojstvo svake kontrole/forme je Text. Naravno da postoji kod kontrola koje mogu prikazati tekst. Na primer, kod kontrole labela Text svojstvo čuva tekst koji kontrola prikazuje. Kod kontrole dugme predstavlja tekst koji se prikazuje na dugmetu. Dalje, kod forme predstavlja tekstualni sadržaj u naslovnoj liniji itd, itd.

Promena ovog svojstva izvodi se u toku dizajna koristeći Properties prozor i biranjem Text polja a zatim ukucavanjem sadržaja. Obratite pažnju da je podrazumevani tekst u Unicode formatu, tj. lako možete koristiti srpska ili neka druga slova. Naravno da se tekst može menjati i programski.

```
static void Main()
{
    Form1 Fm = new Form1(); Fm.Text = "New Application";
    Fm.Size = new System.Drawing.Size(800, 600);
    Application.Run(Fm);
}
```

Visible

Ranije smo pomenuli svojstvo koje određuje da li je kontrola vidljiva ili ne. Ovo svojstvo se naziva Visible a vrednosti su bulovog tipa (Boolean type) True/False.

Ako kontrola nije vidljiva onda se uobicajeno kaže da je sakrivena - hidden. Ovo svojstvo takodje se može programski menjati jednostavnim dodeljivanjem vrednosti. Ukoliko kontrola nije vidljiva onda nije ni dostupna – logično. Drugim recima ona je za korisnika sa stanovišta unosa i prikaza informacija potpuno bez funkcije.

ZAJEDNIČKA SVOJSTVA WINDOWS KONTROLA TREĆI DEO

Svaka kontrola ima svoja svojstva

To ne znači da su takve kontrole nepotrebne – naprotiv. U toku izvršavanja programa one se mogu učiniti vidljivim i dostupnim, ili nevidljivim i nedostupnim, u zavisnosti od toga kakva vrsta podataka se unosi i šta se sa kontrolom radi.

Enabled

Kada je kontrola vidljiva to ne znači obavezno da je i dostupna korisniku. Posebno svojstvo definiše da li kontrola može primiti ulaz od korisnika čak i kada je vidljiva.

Ovo svojstvo se naziva Enabled i takode je bulovog tipa. Ako je kontrola vidljiva, a

Enabled svojstvo postavljeno na vrednost false, to znači da korisnik može samo citirati vrednosti te kontrole bez mogućnosti da ih menja. Podrazumevani izgled kontrole u tom slučaju je blago zasivljenje (dimmed). Ovo svojstvo se često programski, u samom kodu menja. Obično kada unos nekih podataka povremeno (ili stalno) treba zabraniti, a prikaz podataka ostaviti.

Fokus kontrole i aktivan prozor

Pojam fokusa i aktivnog prozora bitni su za razumevanje relacija među prozorima i za objašnjavanje nekih svojstava (a kasnije i događaja). Za kontrolu se kaže da je u

fokusu ako kontrola može da primi ulaz sa tastature. Takvo stanje poseduju kontrole,

ali na primer i stavke u listi. Ono se može posebno graficki oznaciti što se u podrazumevanom vizuelnom podešavanju i radi.

Pojam aktivnog prozora ne vezuje se za kontrolu koja pripada nekoj formi već za samu formu. Kontrola koja ima fokus nalazi se na aktivnom prozoru. Samo jedan prozor je aktivan u jednom trenutku. Na gornjoj slici dat je prikaz jednog primera. Jasno je da je Form1 aktivni prozor a textBox1 u fokusu i da se nalazi na tom prozoru.

ZAJEDNIČKA SVOJSTVA WINDOWS KONTROLA ČETVRTI DEO

Svaka kontrola ima svoja svojstva

TabIndex i TabStop

Ovde ćemo pomenuti još jedno veoma bitno svojstvo u programiranju na grafičkim okruženjima (GUI). Ovo svojstvo nije vidljivo – bar ne na prvi pogled. Obično ga

nazivamo tab redosled ili u originalu tab ordering. Kretanje sa kontrole na kontrolu na

jednoj formi koja je aktivna možete izvesti i preko tastature koristeći Tab taster. Kada se on pritisne fokus kontrole se prebacuje na sledeću. Upravo tab svojstvo određuje redosled kontrola u ovom slučaju.

I ne samo to. Kada govorimo o tab redosledu i tab svojstvima treba znati još nešto. Postoje kontrole koje ne mogu primiti fokus – na primer labela. Takođe, kontrola koja može da primi fokus mora imati svojstvo TabStop postavljeno na true, ako treba da primi fokus. U suprotnom ta kontrola će biti preskocena dok pritisnete taster Tab. Na sreću, ovo svojstvo je za kontrole koje mogu primiti fokus inicijalno postavljeno na true. Kontrole koje ne mogu primiti fokus (primer Labela) nemaju ovo svojstvo.

Koja pozicija u tab redosledu pripada određenoj kontroli definiše svojstvo TabIndex. Ukoliko tu poziciju želite da promenite treba da promenite vrednost ovog svojstva.

Postavljanje kontrole u fokus

Ranije smo objasnili pojam fokusa kontrole u Windows operativnim sistemima. Naglasimo još jednom da fokus označava da je kontrola spremna da primi ulazne

podatke. Na primer, ako se neko dugme nalazi u fokusu to se prikazuje tackastim

pravougaonikom oko teksta na tom dugmetu (obratite pažnju na oznaku za podrazumevano dugme jer se ona često meša sa fokusom). Ako se tekst kontrola nalazi u fokusu onda se u njoj nalazi kursor koji trepće (tada je očigledno da kontrola prima ulaz sa tastature). Kada se neka stavka liste nalazi u fokusu tada je ona oivičena tackastim pravougaonikom. Programski fokus dajemo nekoj kontroli pozivom metode Focus(). Na primer:

```
bool Focus();
```

ZADACI ZA INDIVIDUALNU VEŽBU

Zadatak 1

Napraviti formu sa 6 labela i textboxova kao i jednim dugmetom. Forma treba da ima labele kao i textboxove za unos: imena, prezimena, adrese, telefona, jmbg-a kao i broja lične karte.

Zaključak

ZAKLJUČAK

Zaključak

C# je napravljen da veoma liči na C/C++, ali za razliku od C/C++, C# je lak tj. brz za učenje. Naime, kod C#, memorijski menadžment je automatski, lokatori tj pokazivači, *pointers*, se retko koriste, i ne koriste se “zaglavni fajlovi” ili “makroi”, *header files* ili *#define macros*.

Mogu se nabrojati sledeći razlozi zašto je korisno programirati tj. pisati programe u Visual C# : Visual C# je potpuno objektno-orijentisani jezik, Omogućuje lako i brzo programiranje moćnih Web-aplikacija, Visual C# se puno koristi od strane kompanije Microsoft, što ga čini pogodnim jezikom za operativni sistem Windows koji je od iste kompanije.

Okruženje Visual Studio obuhvata nekoliko jezika, C++ i C# i J#, ali svi oni koriste iste alate.

Razvojno okruženje Visual Studio je integrisano razvojno okruženje, IDE, *Integrated Development Environment*, koje obuhvata editor, kompajler, kao i ostale alate za razvoj projekta tj. aplikacije u Visual C#. Inače, IDE za C++ je obično frustrirajuće kada se poredi sa IDE za Visual C# Express.

ZAKLJUČAK

Zaključak

- Većina grafičkih aplikacija u Visual C# su bazirani na tzv. „formi“, tj. C# - formi tj. Visual C# - formi. Forma u jeziku Visual C# označava jedan specijalni „prozor“, tj. „platno“ za prikazivanje određenih podataka, tj. gde se mogu postavljati i editovati razni objekti, koje smo pomenuli npr. Label, Button, TextBox, itd. Neka Forma ima karakteristike, *properties*, kao i ostali objekti u Visual C#.
- *Property Editor* tj. *Properties Window* (prozor *Properties*) prikazuje liste karakteristika, *properties*, za pojedine objekte u *Toolbox*-u, tj. za pojedine „kontrolne“, gde su objekti Label, Button, TextBox, itd, izlistani u *Toolbox*. Mogu se menjati ove karakteristike, *properties*, pomoću *Properties Window*. Karakteristike su npr. Font, Color, Size, Text, itd.
- *Toolbox* je prozor u Visual C# koji sadrži pojedine komponente neke grafičke aplikacije, npr. „dugme“, ili „tekstboks“, ili „nalepnica“, itd. U predavanju broj 4 (nedelja 4) će biti detaljno analizirani elementi u *Toolbox*-u.
- Pomoću Visual C#, mogu se praviti Dinamičke Web-aplikacije, ali ove aplikacije zahtevaju da se ima neki **Web-server**. Tako da postoji opcija, kod instalisanja Visual C#, da se instalise na sopstvenoj mašini jedan Web-server. Medjutim, i bez Web-servera može se koristiti Visual C#, ukoliko ne želite da kreirate Web-aplikacije. Tako, da se Web-server može instalirati kasnije. Ukoliko ste na nekoj mreži, možda ima *Internet Information Services Web Server*, IISWS, na nekoj drugoj mašini a ne vašoj, pa se umesto da instalirate Web server na vašoj mašini možete odlučiti da koristite Web server IISWS.