

Visual Basic 6.0

Osnove VB

Uvod u Visual Basic



- **Visual** – metoda koja se koristi za kreiranje programske aplikacije
- **BASIC** (**B**eginners **A**ll-purpose **S**ymbolic **I**nstruction **C**ode
- programski jezik s nekoliko inačica:
 - Visual Basic Standard, Professional, Enterprise Edition
 - Visual Basic Application Edition
 - Visual Basic Scripting Edition (VBScript)

Programiranje pod Windowsima

```
Misc: Notepad
[File Edit Format Help]
Something about Windows XP activation

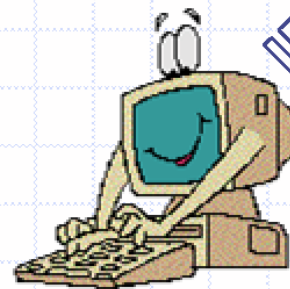
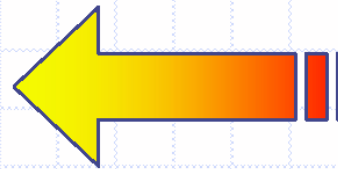
Some one sent me this piece of information that is quite remarkable. I have not tested it but it's
from a reliable source. I quote:

'Did you know that by replacing 11 files on a retail Windows XP CD for an existing corporate edition
CD, you can turn it into an unactivated corporate version? This has been tested and it does
successfully work! It is important to note that this does NOT hack Windows XP in any way, it merely
bypasses WPA. You do, however, need a valid 25-character PLK. It will be interesting to see if
Microsoft fixes this loophole in the first service pack release. I have not monitored my traffic logs
through our checkpoint firewall to see what was communicated yet... I really am surprised Microsoft
would leave such an obvious and easily exploitable method to get around WPA. The files you need to
replace are:

1386\dpc.dll.dl_ |
1386\evnt.txt      *** cosmetic only
1386\ntlm.inf.ca
1386\osmbios.kl
1386\osmbios.ca
1386\osmbios.da
1386\osmbios.nl
1386\pidgen.dll
1386\setupp.ini
1386\setupreg.hiv
1386\win2000\win2000.inf
End quote.
```

Windows aplikacija

Operativni sustav



Događaj

Korisnik

Sadržaj

- Programska okruženja
- GUI
- Šta je i zašto VB?
- Pregled IDE
- Forme (Forms), Kontrole (controls) i Moduli (modules)
- Razvojni put programa
- VB projekat
- Vežbe

Programska okruženja - aplikacije

- Osnovna razlika između programiranja u programskim okruženjima koja *nisu grafička* (MS DOS, UNIX ...) i onim koja su *grafička* (Windows, Mac) je način pisanja i izvršavanja aplikacija.
- Sa tog stanovišta razlikujemo:
 1. **Proceduralne** (tradicionalne) aplikacije
 2. **Event- driven** aplikacije

1. Proceduralne aplikacije

- U tradicionalnim ili proceduralnim aplikacijama, sama aplikacija kontroliše koji se deo koda izvršava.
- Izvršavanje aplikacije počinje u prvoj liniji koda i prati prethodno definisanu putanju kroz aplikaciju, pozivajući procedure (podprograme) ukoliko je to potrebno
- Program se, generalno gledano, uvek izvršava na isti način

2. Event – driven aplikacije

- To su aplikacije zasnovane na **DOGAĐAJU** (*event*). Izvršavanje aplikacije ne prati prethodno definisanu putanju. Umesto toga, različiti delovi koda se izvršavaju kao odgovor na odgovarajuće događaje .
- Događaji mogu biti izazvani:
 - nekom akcijom korisnika,
 - porukama (message) sistema,
 - nekom drugom aplikacijom
 - samom aplikacijom.

Event – driven aplikacije *nastavak...*

- Osnovno pravilo kod event-driven programiranja je da se napiše kod koji može da definiše izvršenje aplikacije na sve moguće događaje koji se mogu dogoditi za vreme izvršenja aplikacije.
- Obzirom da se događaji mogu izvršavati bilo kojim redom, aplikacija se svaki put može izvršiti na drugi način

Graphical User Interface (GUI)

- GUI je grafičko okruženje namenjeno za interakciju korisnika sa aplikacijom.
- Korisnik kontroliše rad aplikacije (programa) preko grafičkih komponenti
- Tako nastaju događaji (event) koji generišu odgovarajuće poruke (message) koji potom obrađuje operativni sistem.

Komponente GUI

- GUI aplikacija prikazuje korisniku grafički interfejs koji se sastoji od grafičkih komponenti:
 - Forme (*forms*)
 - Komandni tasteri (*command buttons*)
 - Labele (*labels*)
 - Tekstualna polja (*text boxes*)
 - Meni (*menu*)
 - ...

Upravljanje u grafičkom okruženju

- Korisnik upravlja sa izvršenjem aplikacije izazivanjem događaja:
 - Klikom na komandni taster (Ok, Cancel ...)
 - Izborom iz menija (File, Edit ...)
 - Unosom teksta
 - Pomeranjem miša
 - ...

Šta je Visual Basic?

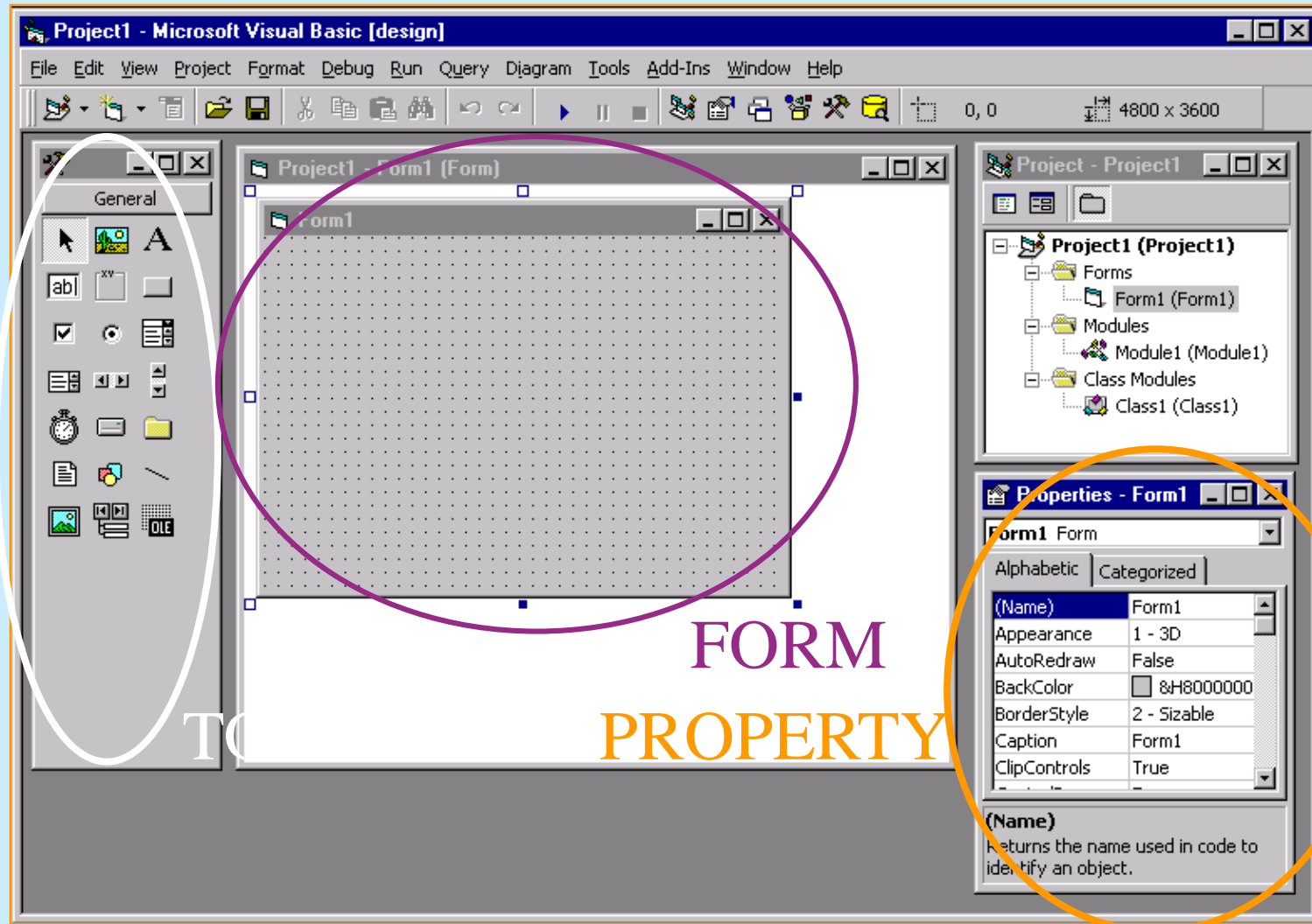
- Beginner's All-purpose Symbolic Instruction Code
- Visual Basic razvijen 1991 (Microsoft Corporation)
- To je razvojni sistem specijalno namenjen za kreiranje grafičkih aplikacija
- On sadrži
 - Alate za grafički dizajn
 - Svoj (programski) jezik

Zašto Visual Basic?

- Relativno brz razvoj aplikacija za Windows OS
- Prilagodljiv za sve tipove razvoja:
 - graphical user interfaces (GUI)
 - Profesionalnog izgleda
 - Rad sa bazama
 - Kreiranje web strana
 - ActiveX / COM components

Visual Basic IDE

Menu, tool box, forms, project, properties...



Tipičan razvoj VB aplikacije

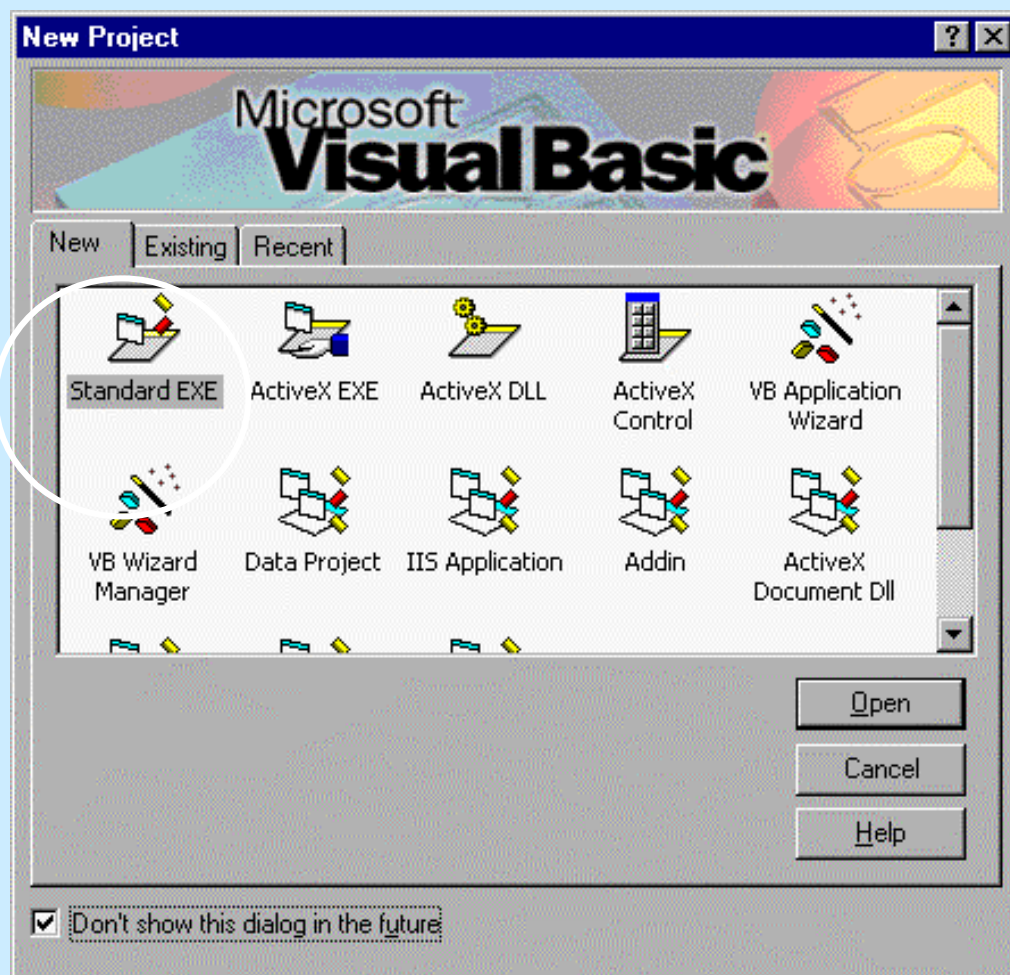
- Tri glavna koraka:
 1. Dizajniranje interfejsa
 2. Pisanje koda
 3. Izvršavanje programa i testiranje (run & test)
 4. Otkanjanje grešaka (debugging)

1. Dizajniranje interfejsa

- Za razliku od ne grafičkih programskih okruženja u VB (Windows) se prvo definiše grafički korisnički interfejs!
- Tek nakon što se on dobro osmisli i kreira piše se programski kod koji treba:
 - Vezati za korisnički interfejs
 - Da pokrije sve moguće događaje bilo da su izazvani od korisnika ili sistema

Dizajniranje interfejsa *nastavak...*

- Pokrenite VB, odaberite “Standard EXE” opciju

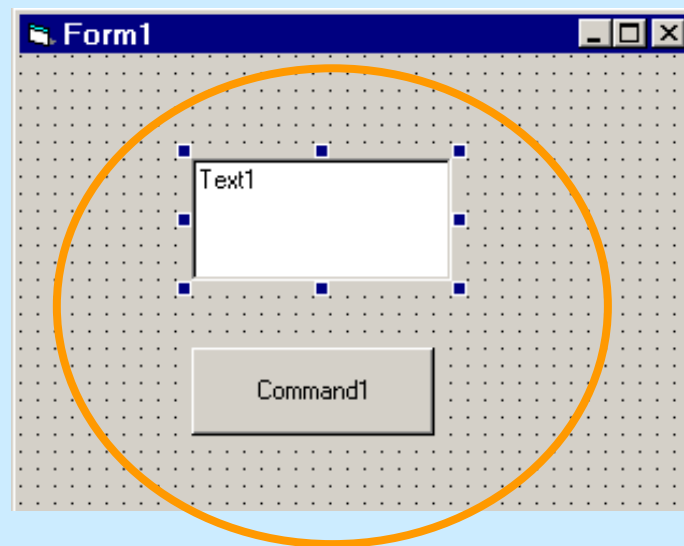
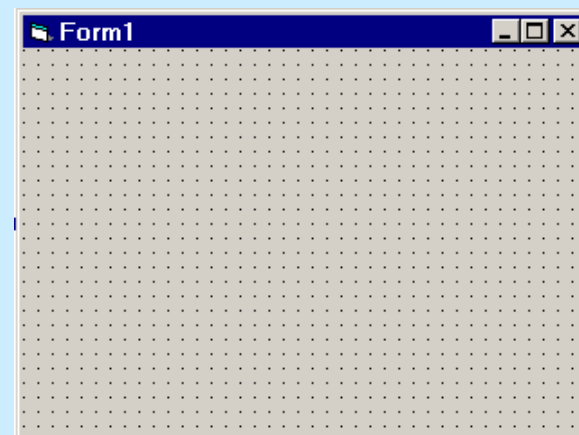


Dizajniranje interfejsa *nastavak...*

- Dobićete prazan prozor (window) nazvan **Form1**
 - u Visual Basic-u, window = ***form***
- “Prenesite” na Formu elemente interfejsa sa palete Toolbox
 - u VB, elementi korisničkog interfejsa = ***control***
 - to su labele, tasteri (buttons), polja za unos teksta (text boxes),.....

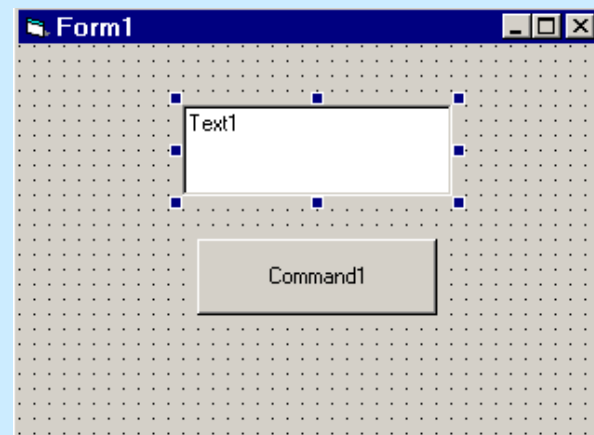
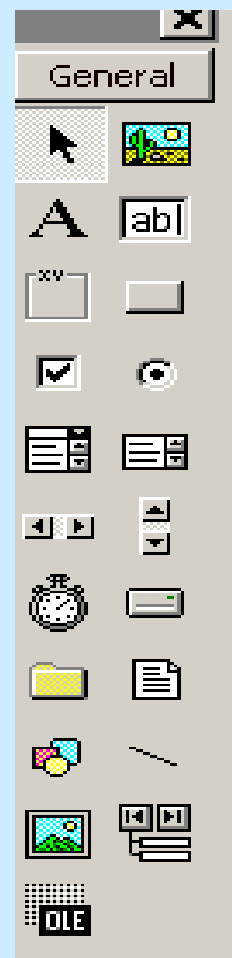
Forme i kontrole

- Forma (FORM) je prozor koji kreirate
- Kontrola (CONTROL) je grafički objekat koji se koriste za unos podataka, komandi, ispis....



Kreiranje kontrola

- Nije potrebno da programirate da bi formirali grafički izgled vaše aplikacije.
- Komande za crtanje su slične onima u Paintbarshu.
- Kliknete na kontrolu u toolboxu zatim pomerite kursor na formu i definišete veličinu kontrole.

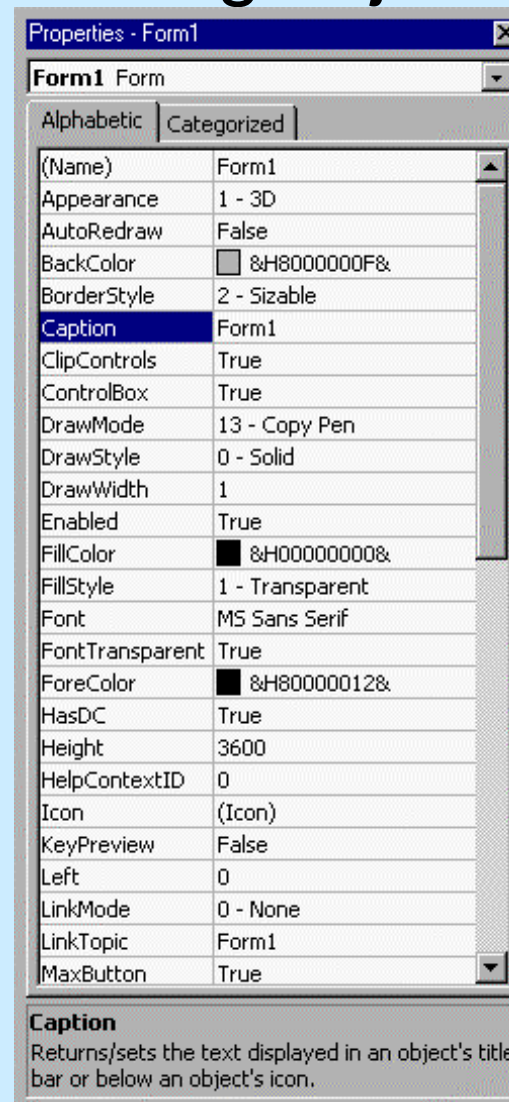


Svojstvo (Property)

- Svojstvo je karakteristika jednog objekta. Za svaki tip objekta VB definiše skup svojstava koji se mogu primeniti na objektu.
- Svako svojstvo ima svoje ime i opcije za podešavanje.
- Svojstvo se može menjati:
 - Dok gradite aplikaciju
 - U toku izvršavanja aplikacije

Properties window

- U toku razvoja aplikacije svojstva nekog objekta možete menjati
 - koristeći properties window.
 - Ili pisanjem koda
`Form1.Caption = "neko ime"`



ImeObjekta.Svojstvo = NovaVrednost

Primer: Svojstva komandnog tastera

- BackColor - Boja pozadine
- Caption— Šta piše na tasteru
- DisabledPicture— Izgled tastera kad je zabranjen
- Enabled— Određuje da li korisnik može koristiti taster.
- Height— Visina tastera
- Left— Pozicija leve ivice tastera
- Style— Da li je taster standardni ili grafički
- Top— Pozicija gornje ivice tastera
- Width— Širina tastera
- ...

2. Pisanje koda

- Kod je potreban da bi aktivirali interfejs a ne da bi ga kreirali.
- Ne morate da pišete kod da bi kreirali objekte vašeg interfejsa ali je to potrebno da bi se nešto dogodilo kada korisnik stupi u interakciju sa interfejsom
- Svaki objekat ima skup svojstava, isto tako mu može biti pridružen određeni skup događaja (events). Događaj predstavlja određenu akciju npr. pritisak tastera na mišu koji se prepoznaje od strane forme ili kontrole

Pisanje koda *nastavak...*

- Mada objekti u VB automatski prepoznaju jedan skup događaja Vi ste ti koji definišete kako će se i da li će se reagovati na određeni događaj
- Kada objekat u VB prepozna da se odigrao događaj on automatski poziva event proceduru sa imenom koja odgovara tom događaju
- Naziv jedne event procedure se sastoji iz imena objekta, donje crtice i naziva događaja
ImeObjekta_NazivDogadaja()

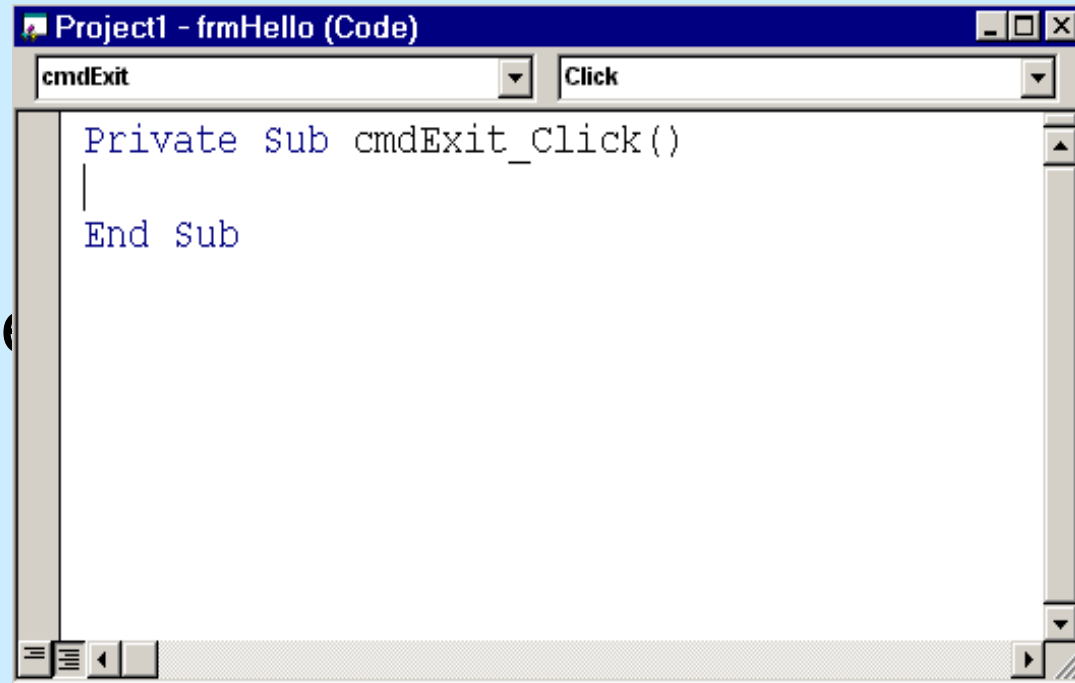
code window

- Do ovog prozora možete doći na više načina ali obzirom da je VB event-driven orijetisan najčešće mu pristupate tako što dvaput kliknete na neki objekat.
- Nakon ovoga otvara se prozor za pisanje programskog koda koji kontroliše upravo taj objekat

code window nastavak...

- comboBox u levom uglu prikazuje ime objekta (cmdExit)
- comboBox u desnom uglu prikazuje ime događaja (Click)

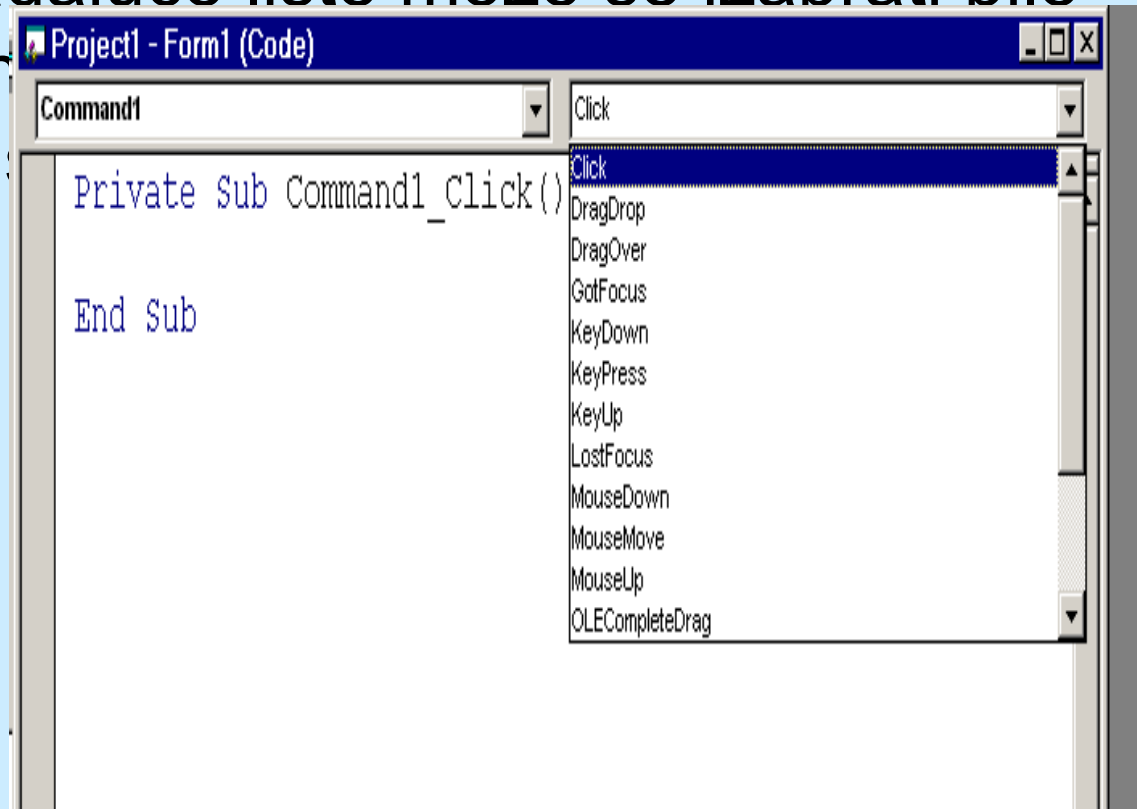
- Dve linije već postoje



```
Project1 - frmHello (Code)
cmdExit Click
Private Sub cmdExit_Click()
|
End Sub
```

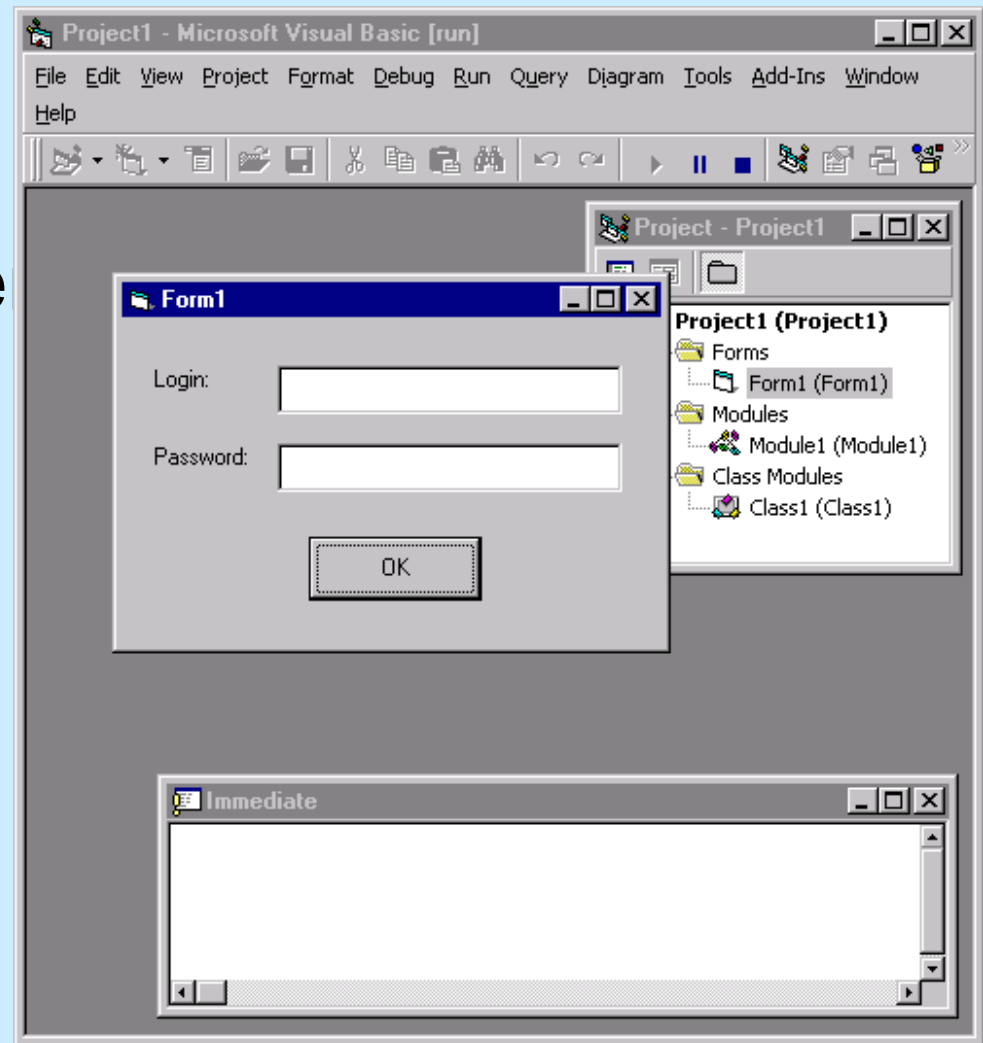
code window nastavak...

- VB automatski nudi da kod vežete za događaj CLICK.
- Otvaranjem padajuće liste može se izabrati bilo koji od ponuđenih željeni program.



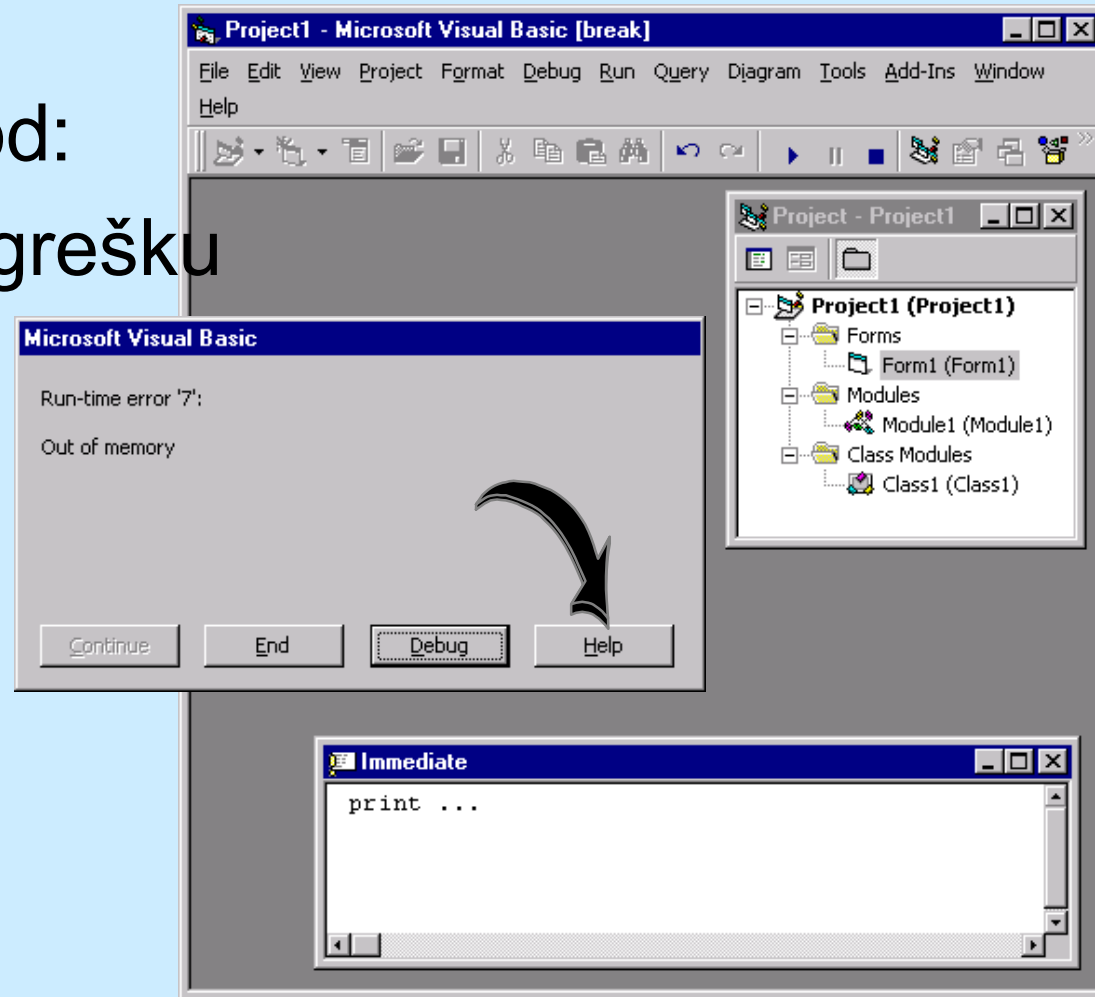
3. run mode

- Aplikaciju možete pokrenutu u bilo kom trenutku!
 - Pritiskom na F5
 - Pomoću Run taste
 - Iz Run menija



4. Otklanjanje grešaka

- Debugging mode
- Ulazite u ovaj mod:
 - Nailaskom na grešku
 - ctrl-break
 - breakpoint
 - Pause taster



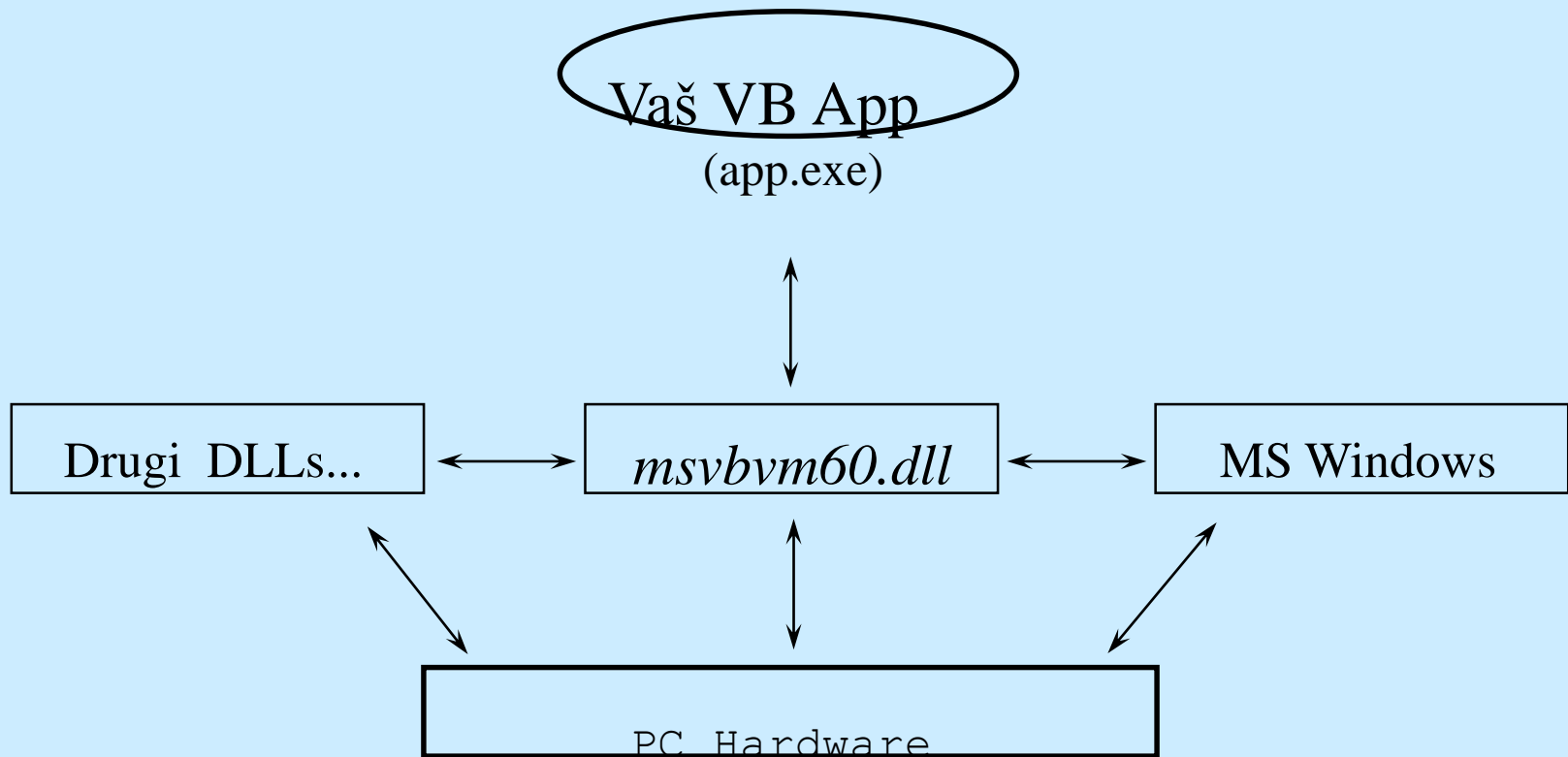
- “Immediate” window postaje aktivan

Gde je prevođenje (compile)?

- Nema posebnog koraka za prevođenje u IDE
 - Direktno prelazite iz dizajniranja u izvršenje programa!
- Sintaksa se proverava automatski u toku unosa
- Semantika se proverava u toku izvršenja programa
- Međutim, kada pravite EXE:
 - VB generiše prevedeni kod
 - Potpuno proverava sintaksu i semantiku

VB exe fajl nije potpun...

- Da bi pokrenuli VB app
msvbvm60.dll mora biti instaliran



Delovi VB projekta?

- Source code fajlovi, postoji tri tipa:
 - *form* (prozori)
 - *class* (object-oriented programming)
 - *module* (standardni potprogrami sa VB kodom)
- Svi fajlovi koje obuhvata projekat su navedeni u project window
 - aktiviranje View >> Project Explorer
- Kompajlirane biblioteke (libraries) / objekti (objects):
 - DLLs (dynamically-linked libraries)
 - ActiveX / COM objects

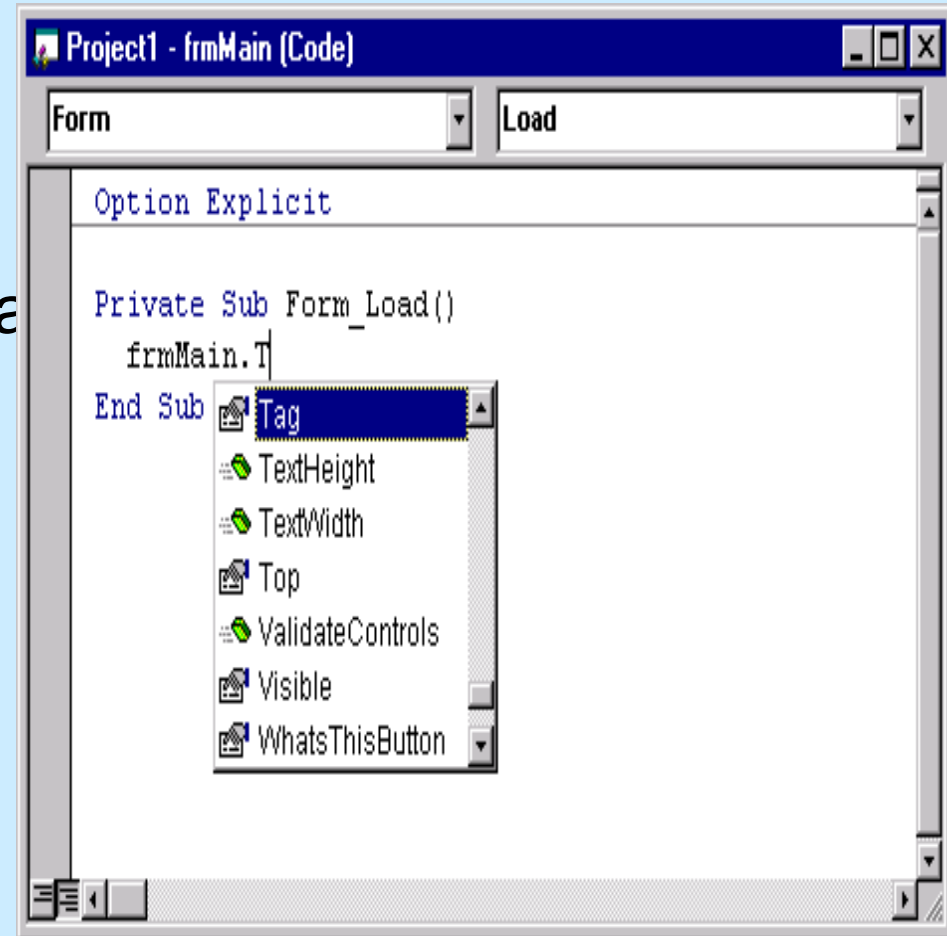
Tipovi fajlova (ekstenzije)

- Source code:
 - .frm / .frx (forme i binarne slike)
 - .cls (class)
 - .bas (moduli)
 - .vbp (predstavljaju jedan *VB projekat*)
 - .vbg (možete spojiti više *VB projekata* u jednu *grupu*)
- Prevedene biblioteke / objekti:
 - .dll
 - .ocx
 - .exe

Pomoć

- *IntelliSense*
 - Kad god je moguće, IDE će ponuditi sugestiju u skladu sa onim što kuca
- SPACE da prihvatite
- ESC da odbacite

- Kompletiranje reči: ctrl-space



online help

- The online help system je
 - Kompletan
 - context-sensitive
- Selektujte nešto u programu, pritisnite F1
 - Npr kontrolu u toolbox-u
- Otkucajte reč, pritisnite F1
 - Npr otkucajte “call” u kode window-u

Message box

Često je potrebno prikazati različite poruke korisniku tokom faze korištenja programa.

Da se ne bi kreirala posebna forma sa porukom Visual Basic ima ugrađenu zgodnu i veoma korisnu osobinu:

prozor za poruke, message box, (msgbox).

Pozivanje prozora sa porukama je veoma jednostavno.

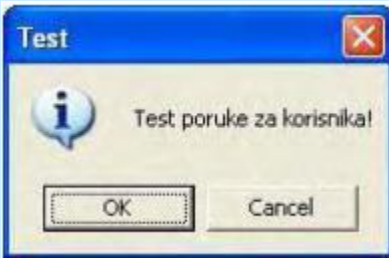
```
MsgBox("Pozdrav")
```

Message i Input Box

Pored prosljeđivanja teksta poruke moguće je takođe upravljati i sa prikazom komandnih dugmadi (OK, Cancel, Yes, No...) te sa prikazom sličica koje odgovaraju vrsti poruke: informacija, upozorenje ili pak kritična greška.

Message box

- Prozor koji se prikazuje na osnovu gore napisane komande je prikazan na slici 3.8.



- Opcionalni parametar koji se može proslijediti prilikom pozivanja message box-a je naslov koji se ispisuje na naslovnoj traci, title bar-u, prozoru za poruke.

MsgBox "Tekst poruke za korisnika!", vbOKCancel + vbInformation, "Test"

Message box

Tabela 3.1 - Vrijednosti funkcije msgbox

Konstanta	Vrijednost	Dugme
vbOK	1	OK
vbCancel	2	Cancel
vbAbort	3	Abort
vbRetry	4	Retry
vbIgnore	5	Ignore
vbYes	6	Yes
vbNo	7	No

Input box

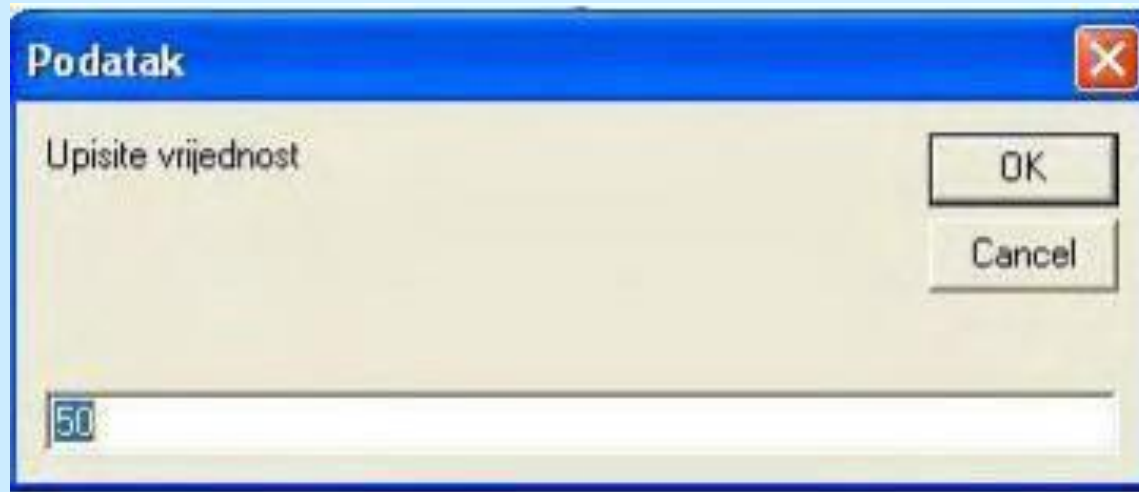
- Input Box se često može iskoristiti za upis određenog podatka od strane korisnika te pridruživanje te vrijednosti varijabli. Sintaksa komande za InputBox je slična sintaksi koja se koristi kod message box-a:

```
inputData = InputBox("Tekst poruke")
```

Input box

- Primjer prozora tipa InputBox koji se pojavljuje usljed naredbe:

InputBox "Upisite vrijednost", "Podatak", 50



Input box

- Vrijednost koja se upiše u polje pridružuje se varijabli u naredbi za pozivanje InputBox-a, na primjer u slučaju naredbe:

vrijednostX = InputBox ("Upisite vrijednost")

msgbox vrijednostx

vrijednost koju korisnik upiše pridružuje se varijabli vrijednostX.

Input box

Tokom faze pisanja programskog koda Visual basic editor automatski provjerava sintaksu u svakoj liniji koda.

Ako Visual Basic pronade da nešto nije u redu u liniji programskog koda, pojavit će se dva neovisna događaja koja će sugerirati grešku:

1. Linija programskog koda koja je pogrešno napisana bit će prikazana crvenom bojom.
2. Prozor sa porukom o grešci će da se pojavi na ekranu, ukratko objašnjavajući u čemu je problem.

Zadaci

- **Zadatak 1:** Sastavite program za izračunavanje površine kružnice koristeći ugrađene `InputBox ()` i `MsgBox ()` Visual Basic funkcije.

Rješenje:

```
Private Sub Form_Load()
```

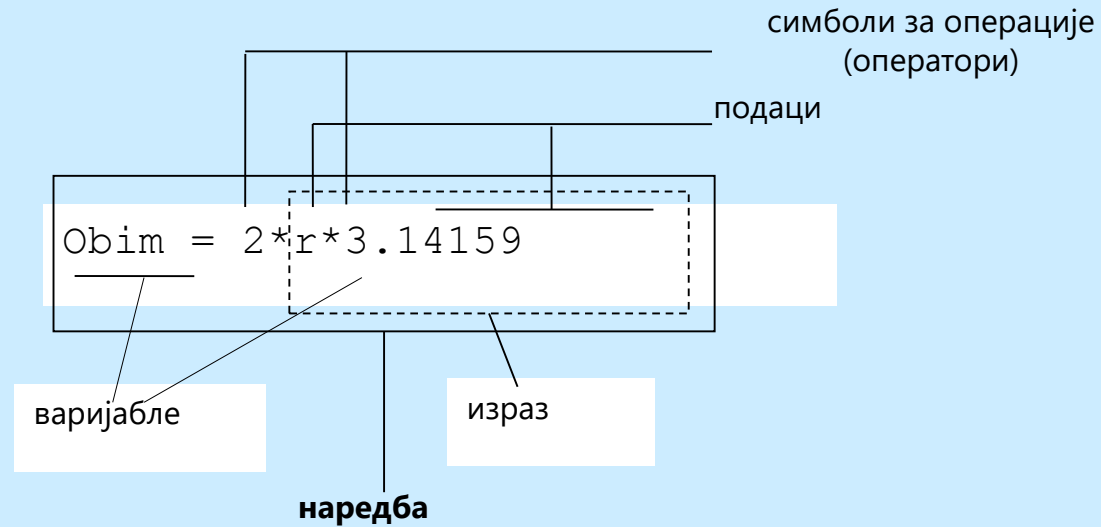
```
    R = InputBox("Unesi R")
```

```
    P = R * R * 3.14159
```

```
    MsgBox (P)
```

```
End Sub
```

Zadaci



Zadaci

Zadatak 2: Sastavite program za izračunavanje nominalnog napona epruvete kružnog poprečnog presjeka opterećene na zatezanje koristeći ugrađene `InputBox ()` i `MsgBox ()` Visual Basic funkcije.

Zadaci

```
Private Sub Form_Load()  
    Sila = InputBox("Unesite silu [N]")  
    R = InputBox("Unesite radijus  
epruvete [mm]")  
    P = R * R * 3.14159  
    Napon = Sila / P  
    MsgBox Napon  
End Sub
```

Agenda

- Osnovna obilježja radnog okruženja (eng. IDE)
Visual Basica
- Elementi programskog projekta
- Osnovni objekti
- Metode, svojstva i događaji objekata
- VB pomoć
- Sažetak

