

ПОЛЗВАНЕ НА СКРИПТОВЕ В INDESIGN

Автоматизирането на много от функциите на InDesign може да бъде изпълнено с помощта на скриптове. В най-типичното си приложение, скриптовете се използват за автоматизиране на повтарящи се операции

Скриптове на InDesign с използване на Visual Basic.

Visual Basic е мощен обектно ориентиран език за програмиране, който опростява програмирането. Когато се използват обекти на InDesign, той може да автоматизира и опрости много от функциите на тази програма. Прозорецът на приложната програма е обект. Всеки бутон е обект. Менютата, лентите с инструменти също са обекти. Ако отворим кой да е диалогов прозорец на InDesign, можем да приемем, че всеки елемент на този прозорец е сам посебеси обект. Всяко нещо - от падащото меню до бутоните Cancel и Open са обекти. Дори областта на прозореца, съдържаща списъка и иконите на файловете и папките може да бъде отворена като обект.

Свойства

Всеки обект има свои свойства. Те описват как изглежда той и как функционира в своята приложна програма.

Бутоните имат следните свойства:

- височина
- широчина,

- Тези свойства определят размера на бутона. Надписът върху бутона определя неговата функция.

Език за скриптове

Създаване на приложен скрипт за обект - бутон. Когато се кликне върху бутона, той трябва да извежда на екрана кратко съобщение за потребителя.

1. Стартирайте Visual Basic и създайте нов проект (New Project) - стандартен EXE файл.
2. Visual Basic автоматично създава празна бланка, за да започнете програмирането.
3. От кутията с инструменти на Visual Basic изберете и начертайте нов бутон някъде в празната бланка.

4. Ако виждате прозорец, именуван **Properties**, преминете към следващата стъпка. В противен случай, щракнете с десния бутон на мишката върху новия бутон и изберете **Properties** от контекстното меню което се появява.
5. Намерете и щракнете върху свойството **Caption** (Заглавие) в прозореца **Properties**.
6. Напишете **Click Me** (Щракни върху мен) и натиснете клавиша **Enter**.
7. Щракнете двукратно върху новия бутон. Ще се появи прозорец за написване на програмния код [съкр. „код“].

8. Напишете следните редове в прозореца за кода:

```
Private Sub Command1_Click()
```

```
End Sub
```

```
Dim myMessage as String
```

```
MyMessage = "Congratulation, you 've created a  
button object!"
```

```
Msgbox myMessage
```

'// This is example code demonstrating the use of a
button

'// Това е примерен програмен код, показващ
използването на бутон, който при щракване
върху него извежда съобщение: „Поздравление,
вие създадохте обект - бутон”.

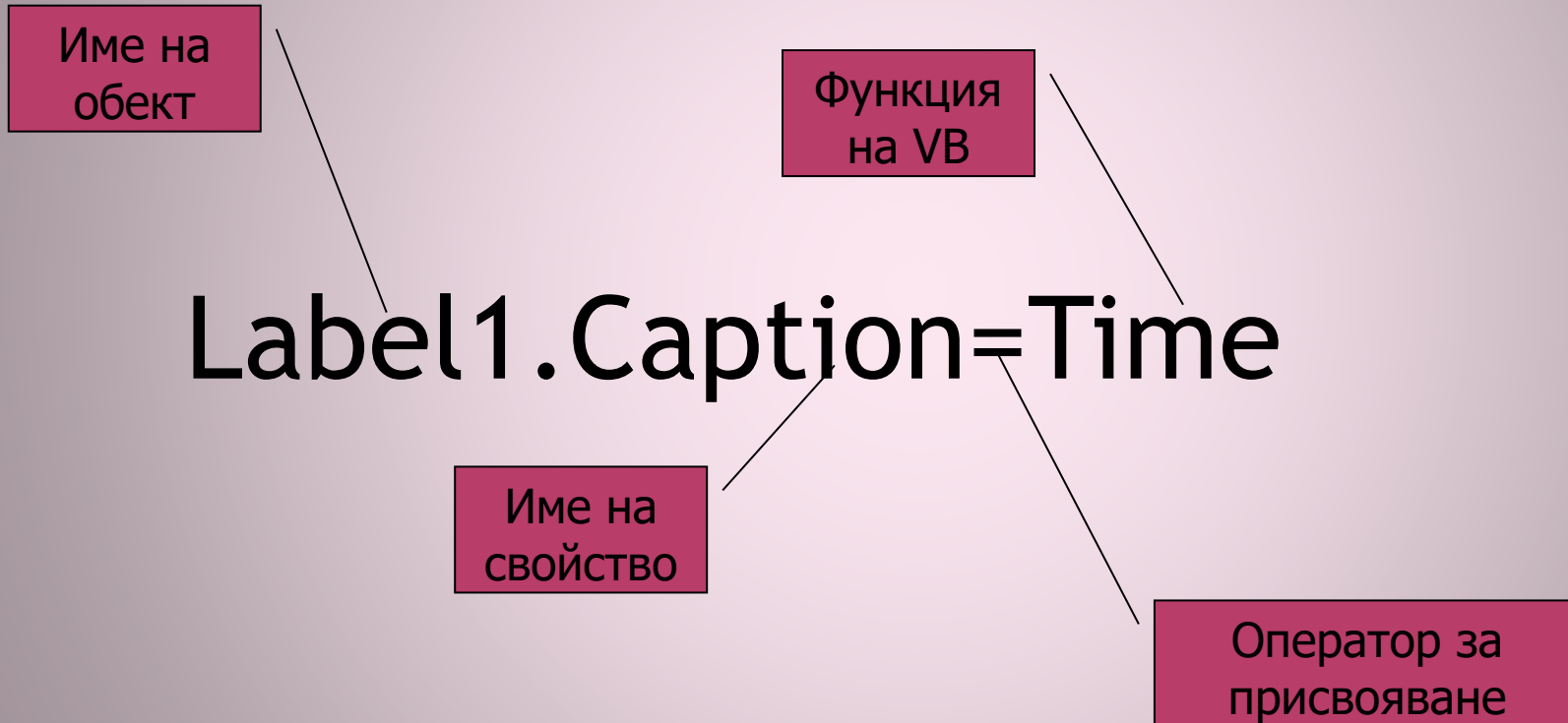
Коментарите не са задължителни към скриптовете, но могат да бъдат много полезни. Всички коментари се отбелязват със знака '//'. При изпълнение на скрипта те се игнорират и не се изпълняват.

ПРОМЕНЛИВИ И ОПЕРАТОРИ ВЪВ VISUAL BASIC

- Програмна конструкция- всяка комбинация от ключови думи, свойства, функции, оператори, методи и символи, които заедно създават валидна инструкция, разпозната от компилатора на Visual Basic.
- Накратко програмната конструкция можем да наричаме и код.
- Програмната конструкция може да бъде и една служебна дума.

Пример: Veer или End.

ПРИМЕР ЗА ПРОГРАМНА КОНСТРУКЦИЯ, ПРИСВОЯВАЩА ТЕКУЩОТО ВРЕМЕ НА СВОЙСТВОТО CAPTION НА ЕДИН ЕТИКЕТ.



ПРАВЕНЕ НА РЕЗЕРВАЦИИ ЗА ПРОМЕНЛИВИ: КОНСТРУКЦИЯТА НА DIM

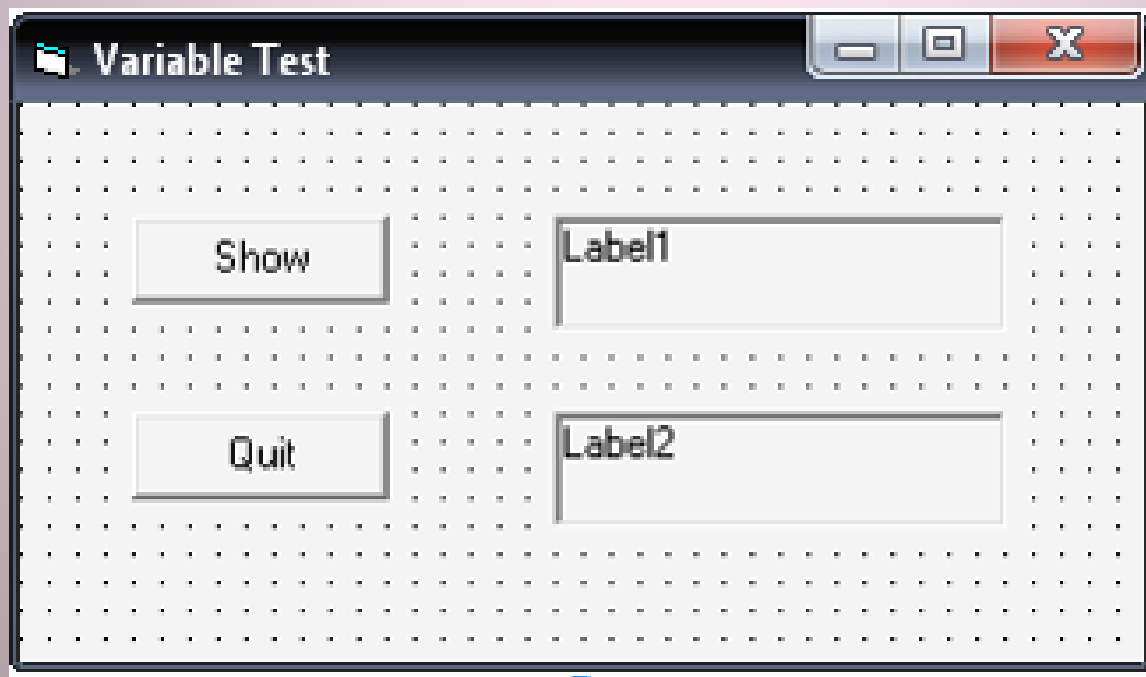
- Местоположение: началото на програмната конструкция.
- Всяко деклариране на променлива започва с конструкцията Dim (Dimension).
- Пример: Dim LastName
- Присвояване: LastName= “Hello”

ДЕКЛАРИРАНЕ НА ПРОМЕНЛИВА

- Неявно деклариране на променлива:
`LastName= "UFO"`
- Недостатък- компилатора не показва съобщение за грешка, ако по-късно се въведе погрешно името на променливата.
- Ако използвате конструкцията `Dim`, ще трябва да поставите отметка на `Require Variable Declaration`, намираща се в `Tools > Options... > Editor`.

ИЗПОЛЗВАНЕ НА ПРОМЕНЛИВИ В ПРОГРАМА

- Създайте следния графичен интерфейс:



ПРОГРАМНАТА КОНСТРУКЦИЯ НА БУТОНА SHOW

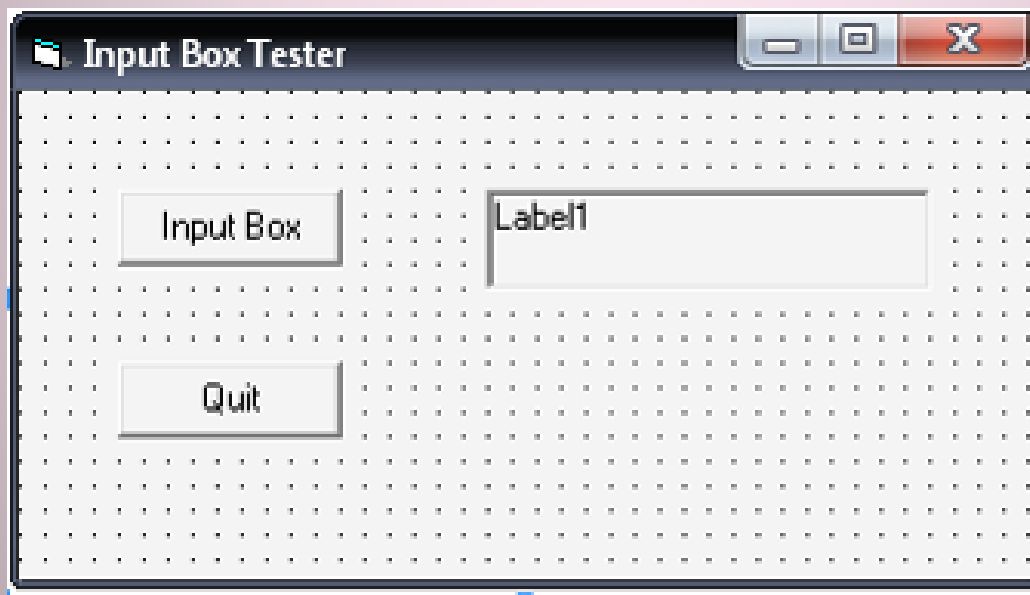
```
Private Sub Command1_Click()  
    Dim LastName  
    LastName = "Ivanov"  
    Label1.Caption = LastName  
    LastName = 66  
    Label2.Caption = LastName  
End Sub
```


ОГРАНИЧЕНИЯ ЗА ИМЕНАТА НА ПРОМЕНЛИВИТЕ В VB

- Трябва да започва с буква;
- Дължината не трябва да надвишава 256 знака;
- Допустими знаци са a..z, A..Z, 0..9 и знаците _ % \$! @ # &;
- Не трябва да се използват ключови думи, обекти или свойства като имена на променливи;
- Имената на променливите трябва да са кратки и едновременно поясняващи за какво се използва променливата.

ПОЛУЧАВАНЕ НА ВХОДНА ИНФОРМАЦИЯ ЧРЕЗ INPUTBOX

- Създайте следния графичен интерфейс:



ПРОГРАМНАТА КОНСТРУКЦИЯ НА БУТОНА INPUT BOX

```
Private Sub Command1_Click()  
    Dim Prompt, FullName  
    Prompt = "Please enter your name."  
    FullName = InputBox$(Prompt)  
    Label1.Caption = FullName  
End Sub
```

КАКВО Е ФУНКЦИЯ?

- Конструкция, която извършва някаква работа (получава аргумент) и след това връща резултат.
- Функцията `InputBox` използва променливата `Prompt`, за покаже в диалоговия прозорец инструкциите за потребителя.

СИНТАКСИС НА ФУНКЦИЯТА INPUTBOX В ПРЕДХОДНАТА ЗАДАЧА

Име на
променлива

Име на
функция

FullName = InputBox\$(Prompt, Title)

Оператор за
присвояване

Аргументи

ИЗПОЛЗВАНЕ НА ПРОМЕНЛИВА ЗА ИЗХОДНА ИНФОРМАЦИЯ

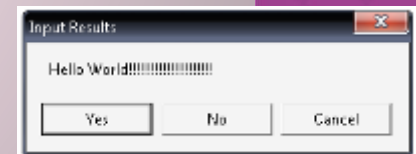
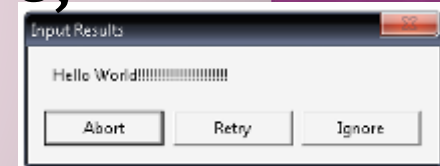
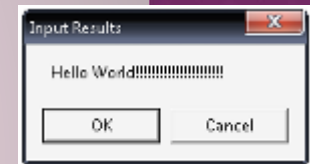
ПОКАЗВАНЕ НА СЪОБЩЕНИЕ

- VB използва диалогов прозорец за извеждане на информация.
- Синтаксис:
`ButtonClicked=MsgBox(Message,
NumberOfButtons, Title)`
- `Message`- текстът, който трябва да бъде показан на екрана.
- `NumberOfButtons`- номер на стила на бутона (от 0 до 5)
- `ButtonClicked`- резултат, показващ кой е натиснатия бутон.
- `Title`- текстът на заглавната лента.

ПРОМЯНА ВИДА НА БУТОНИТЕ В MSGBOX ЧРЕЗ ПРОМЯНА НА NUMBEROFBUTTONS

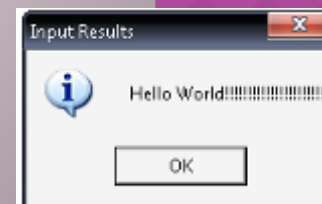
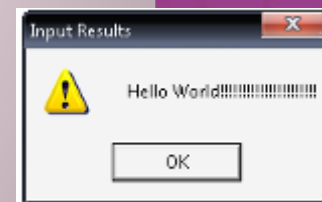
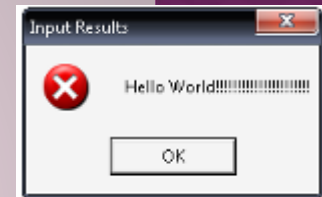
ButtonClicked=MsgBox(Message, **NumberOfButtons**, Title)

- 0- появява се бутон ОК;
- 1- появяват се бутони ОК и Cancel;
- 2- появяват се бутони Abort, Retry и Ignore;
- 3- появяват се бутони Yes, No и Cancel;
- 4- появяват се бутони Yes и No;
- 5- появяват се бутони Retry и Cancel.



ПОСТАВЯНЕ НА ИКОНИ НА MSGBOX

- MsgBox Message, NumberOfIcons, Title**
- 16- Critical Message (извънредно важно съобщение);
 - 32- Warning Query (запитване или въпрос);
 - 48- Warning Message (предупреждение);
 - 64- Information Message (информативно съобщение).



РАБОТА С КОНКРЕТЕН ТИП ДАННИ

ТИП ВАРИАНТ

- Вариантните променливи могат да съхраняват всички базови (предварително дефинирани) типове данни на VB.
- Да превключват между форматите автоматично.
- Лесни за използване, защото не се налага да се декларират.
- Недостатъци: заема големи ресурси, поради това не може да се създаде бърз и “стегнат” код.

ТИПОВЕ ДАННИ

Тип данни	Размер	Обхват	Примерна употреба
Integer	2 байта	От -32 768 до 32 767	Dim Birds As Integer Dim Birds%
Long integer	4 байта	От -2 147 483 648 До 2 147 483 647	Dim Loan As Long Dim Loan&
Single	4 байта	От -3.402823E38 До 3.402823E38	Dim Price As Single Dim Price!
Double	8 байта	От -1.79769313486232D308 До 1.79769313486232D308	Dim Pi As Double Dim Pi#
Currency	8 байта	От -922337203685477.5808 До 922337203685477.5807	Dim Debt As Current Dim Debt@
String	1 байт за знак	От 0 до 65535 знака	Dim Dog As String Dim Dog\$
Boolean	2 байта	True или False	Dim Flag As Boolean
Date	8 байта	От 1 януари 100г. До 31 декември 9999г.	Dim Birthday As Date
Variant	16 байта (за числа); 22 байта + 1 байт за знак (за низове)	Обхватите на всички типове данни	

ПРИМЕРИ ЗА КОНСТАНТИ

- Long Integer: Loan&=350 000
- Single: Price!= 899.99
- Double: Pi#= 3.14159226535
- Currency: Debt@= 7600300.50
- String: Dog\$= “pointer”
- Boolean: Boolean Flag= True
- Date: Date Birthday= #3-1-63#
- Variant: Total= 289.13

ПОТРЕБИТЕЛСКИ ДЕФИНИРАНИ ТИПОВЕ ДАННИ

- Това се налага, когато искате да създадете собствени типове данни.
- Представяват комбинация от няколко основни типове данни.
- Конструкцията `Туре` трябва да се намира в секцията с декларациите на някой стандартен модул.

СЪЗДАВАНЕ НА ПОТРЕБИТЕЛСКИ ДЕФИНИРАНИ ТИПОВЕ ДАННИ

- Пример:

Type Employee

 Name As String

 DateOfBirth As Date

 HireDate As Date

End Type

- Започва с конструкцията Type.

- Изброяват се типовете, които се съдържат в потребителски дефинирания тип.

- Използването им става чрез задаване на променливи в частта Dim от този тип:

Dim ProductManager As Employee

 ProductManager.Name = "Ivan Ivanov"

ИЗПОЛЗВАНЕ НА УСЛОВНИ ОПЕРАТОРИ

БАЗИРАНЕ НА СЪБИТИЯ ПРОГРАМИРАНЕ

- “Интелигентни” обекти - знаят как да се държат, когато потребителя взаимодейства с тях.
- Какви структури управляват тези събития?

ОПЕРАТОРИ ЗА СРАВНЕНИЕ

Оператор за сравнение	Значение
=	Равно на
<>	Различно от
>	По- голямо от
<	По- малко от
>=	По- голямо или равно
<=	По- малко или равно

СИНТАКСИС И СЕМАНТИКА НА УСЛОВНИЯ ОПЕРАТОР IF

- **Синтаксис:**

IF условие Then

 инструкция_1

Else

 инструкция_2

End IF

- **Семантика:** инструкция_1 се изпълнява ако условието е вярно (True), а инструкция_2- ако условието е не вярно (False).

ВЛОЖЕНИ УСЛОВНИ ОПЕРАТОРИ

If a=2 Then

 Text1.Text="Слаб (2)"

Elseif a=3 Then

 Text1.Text="Среден (3)"

Elseif a=4 Then

 Text1.Text="Добър (4)"

Elseif a=5 Then

 Text1.Text="Много добър (5)"

Elseif a=6 Then

 Text1.Text="Отличен (6)"

Else

 Text1.Text="Въведеното число не е оценка"

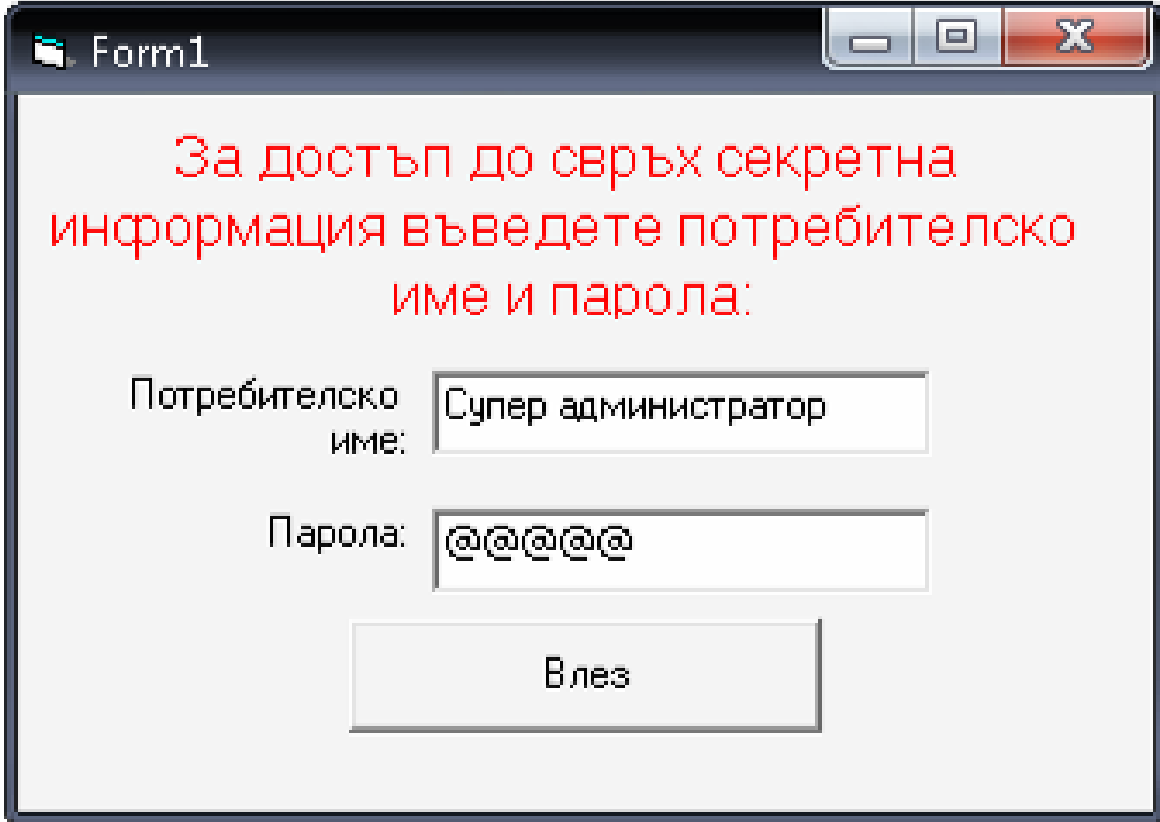
End If

ИЗПОЛЗВАНЕ НА ЛОГИЧЕСКИ ОПЕРАТОРИ В УСЛОВНИ ИЗРАЗИ

ЛОГИЧЕСКИ ОПЕРАТОРИ

Логически оператор	Значение
And	Ако и двата условни израза са True, резултатът е True.
Or	Ако някой от условните изрази е True, резултатът е True.
Not	Ако условният израз е False, резултатът е True. Ако условният израз е True, резултатът е False.
Xor	Ако един и само един от условните изрази е True, резултатът е True. Ако и двата са True или False, резултатът е False.

ПРАКТИЧЕСКА ЗАДАЧА



Form1

За достъп до свръх секретна информация въведете потребителско име и парола:

Потребителско име: Супер администратор

Парола: @@@@

Влез

УПЪТВАНЕ

- За не се виждат символите на паролата изберете от прозореца Properties свойството PasswordChar и въведете желаня от вас символ за скриване (* или @ или ^ или други).
- Използвайте условния оператор за разпознаване.

СЪЗДАВАНЕ НА ГРАФИЧНИ БУТОНИ

НАЧИНИ ЗА СЪЗДАВАНЕ НА БУТОНИ

- Чрез контрола команден бутон;
- Чрез графични бутони- поставя се с контрола Image. Пример за графичен бутон са инструменталните ленти.

ОТКРИВАНЕ НА СЪБИТИЕ MOUSEDOWN

- Използва се за по-реалистичен вид на бутоните при събитие кликване (бутоната променя вида си).
- `MouseDown`- събитие, което се извършва когато потребителят позиционира показалеца на мишката върху обекта и натисне левия бутон. Т.е. сменя графичния обект при събитие.
- `MouseUp`- събитие, което се извършва при отпускане на бутоната.

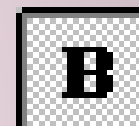
СМЯНА НА БУТОНИ

Всеки бутон има три състояния:

○ Нормално положение на бутона (изпъкнало положение);



○ Натиснат бутон (вдлъбнато положение);



○ Забранен бутон (текущо неактивен).

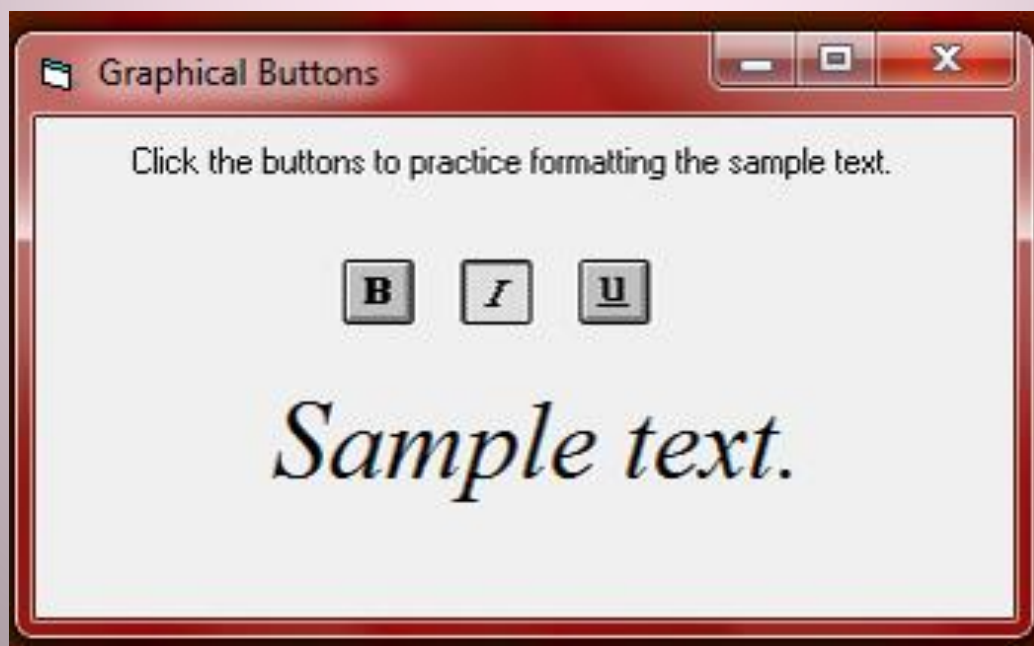


ПОЛЕЗЕН СЪВЕТ ПРИ ПРАВЕНЕ НА БУТОНИ С ИНСТРУМЕНТИ

- Можете да създадете графични бутони и ленти с инструменти на всяка форма на VB 6.0.
- Но MDI формите имат специални вградени свойства, улесняващи работата с колекции от бутони.
- Контролът Toolbar ActiveX, включен в изданията Professional и Enterprise на VB 6.0 ви помагат да създавате и управлявате ленти с инструменти на своите MDI форми.

ПРАКТИЧЕСКА ЗАДАЧА

- Създайте програма с графични бутони, която задава стил на текст.
- Свалете графичните файлове за бутоните от сайта (Buttons.rar).



УКАЗАНИЯ

- Използвайте събитието `MouseDown`.

- Примерен код на единия от бутоните:

```
Private Sub Image1_MouseDown(Button As Integer, Shift  
As Integer, X As Single, Y As Single)
```

```
    If Image1.Tag = "Up" Then
```

```
        Image1.Picture = Image4.Picture
```

```
        Label2.FontBold = True
```

```
        Image1.Tag = "Down"
```

```
    Else
```

```
        Image1.Picture = Image7.Picture
```

```
        Label2.FontBold = False
```

```
        Image1.Tag = "Up"
```

```
    End If
```

```
End Sub
```

- Поправете грешките от тип качествен код.

**ПРИБАВЯНЕ НА ИЗОБРАЖЕНИЕ, ЧРЕЗ
КОНТРОЛИ LINE И SHAPE**

КОНТРОЛЪТ LINE

- Създава права линия.
- Специфични свойства:
 - `BorderWidth`- дебелина на линията;
 - `BorderStyle`- прави линията плътна, на точки или прекъснатата;
 - `BorderColor`- цвят;
 - `Visible`- видима(`True`) или невидима(`False`).

КОНТРОЛ - SHAPE

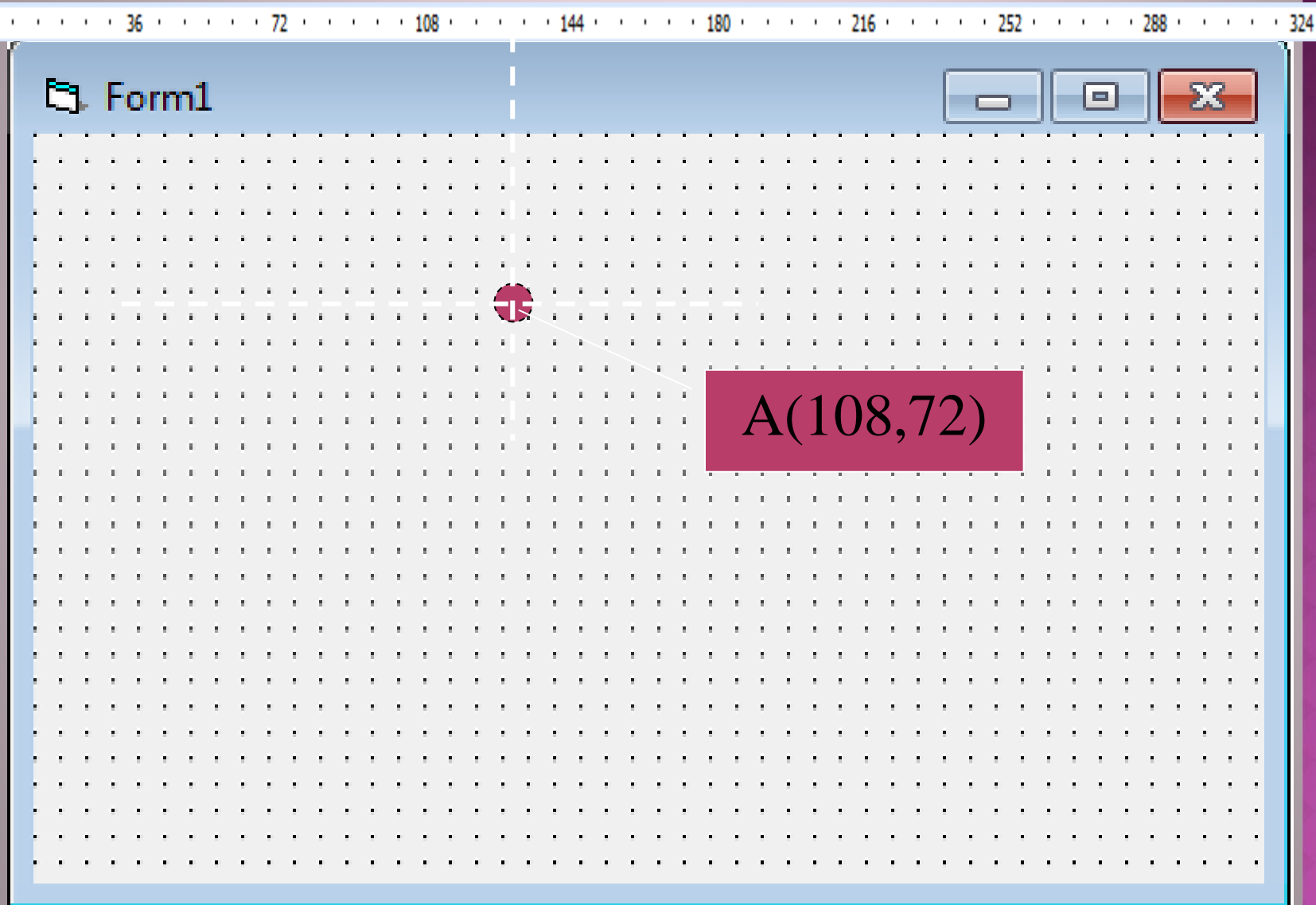
- ◎ Създава правоъгълници, квадрати, овали и окръжности.
- ◎ Специфични свойства:
 - Shape- избира вида на фигурата;
 - FillColor- цвят на обекта;
 - FillStyle- шарка на цвета на обекта;
 - BorderColor- цвят на контура;
 - Visible- видима(True) или невидима(False).

ИЗПОЛЗВАНЕ НА МЕТОДИ ЗА СЪЗДАВАНЕ НА ФИГУРИ

- Линия се определя от две точки- например $A(x_1, y_1)$ и $B(x_2, y_2)$ с начална точка A и крайната точка B .
- Окръжност се определя с център (точка с координати) и радиус.
- Пример за линия:
Line.X1=120
Line.X2=6360
Line.Y1=720
Line.Y2=720
- Пример за окръжност: Circle(1500, 1500), 750. Това е окръжност с радиус 750 и център с координати (1500, 1500)

ДОБАВЯНЕ НА АНИМАЦИЯ КЪМ ПРОГРАМАТА

ИЗПОЛЗВАНЕ НА КООРДИНАТНА СИСТЕМА



ПРИДВИЖВАНЕ НА ОБЕКТИ В КООРДИНАТНАТА СИСТЕМА

- Синтаксис
Object.Move left, top
- Пример:
Picture1.Move 1400, 700
- Семантика- обекта се придвижва на ново местоположение с координати $x=1400$ и $y=700$ (точката е горния ляв ъгъл на обекта).
- Важно: $1400 = 1$ инч.

ОТНОСИТЕЛНО ДВИЖЕНИЕ.

Преместваме обекта от първоначалното му състояние.

○ Примери:

- `Picture1.Move Picture1.Left - 50, Picture1.Top - 10`
- `Picture1.Move Picture1.Left + 50, Picture1.Top + 10`

ПРАКТИЧЕСКА ЗАДАЧА: АНИМАЦИЯ С МЕТОДА MOVE

- Добавяне на пушек към програмата DragDrop от предходния урок.
- Свалете архива DragAndDrop.rar от сайта. В нея се съдържа програмата и помощни графични файлове.
- Разгледайте програмата Smoke.exe.
- Използвайте обект Timer.

РАЗШИРЯВАНЕ И СВИВАНЕ НА ОБЕКТИ

- За движение на обекти се използват свойствата `Top` и `Left`.
- За разширяване и свиване се използват свойствата `Height` и `Width`.

Примери:

`Image1.Height = Image1.Height + 200`

`Image1.Width = Image1.Width + 200`

КАЧЕСТВЕН ПРОГРАМЕН КОД

КАКВО Е КАЧЕСТВЕН ПРОГРАМЕН КОД?

- Когато след време погледнете вашата програма да разберете веднага как сте я направили и каква логика в нея сте използвали.
- Да можете много бързо да поправите или модифицирате програмния код.

ИМЕНУВАНЕ НА ИДЕНТИФИКАТОРИ

- Ако идентификаторът е обект (името се дава от свойството Name), тогава се поставя префикс, съответстващ на типа на обекта. Например: `cmdExit`, `imgBird` и други, които ще видите в следващия слайд.
- Ако идентификаторът е променлива или константа, тогава името трябва да показва за какво се използва.

ТАБЛИЦА ЗА ИМЕНУВАНЕ НА ОБЕКТИ

Обект	Префикс	Обект	Префикс
ComboBox	Cbo	CheckBox	Chk
Command	Cmd	Dialog	Dlg
Data	Dat	DirListBox	Dir
DriveListBox	Drv	FileListBox	Fil
Frame	Fra	Form	frm
HScrollBar	Hsb	VScrollBar	Vsb
Image	Img	Label	Lbl
Line	Lin	listBox	Lst
Menu	Mnu	OLE	Ole
OptionButton	Opt	Picture	Pic
Shape	Shp	TextBox	Txt
Time	Tmr		

СЪЗДАВАНЕ НА ПРОЦЕДУРИ

- Цялата процедура трябва да се вижда на екрана.
- Ако е по-голяма, тогава трябва да се раздели на две или повече логически завършени части с помощта на други процедури.
- Индикатор за добре написана процедура е бързо разбиране и преглед с един поглед без прелистване.

ДЕКЛАРИРАНЕ НА ПРОМЕНЛИВИ

- Не използвайте едни и същи променливи за няколко различни неща. Например: променлива, която брои индексите на масив с променлива, която извежда оценки.
- Ако имате число, което използвате няколко пъти, по-добре е да го дефинирате като константа.

**СТРУКТУРА НА СТРАНИЦА,
СЪДЪРЖАЩА ИЗОБРАЖЕНИЯ И
ТЕКСТОВЕ С РАЗЛИЧНИ
СПЕЦИФИКАЦИИ.**

СКРИПТОВЕ ЗА INDESIGN

Създаване на структура на единична страница, която да съдържа различни елементи.

1. Стартирайте Visual Basic и създайте нов проект (стандартен EXE файл)
2. Изберете Reference от меню Project.
3. Намерете Adobe InDesign Type Library и щракнете върху полето за отметка. Ако не можете да намерите библиотеката, щракнете върху бутона Browse и намерете файла scripting.rpln. Файла е в папка InDesign/Required на вашият твърд диск. (Тази библиотека не е колекция от шрифтове, а файл с информация за съдържащите се обекти, свойства, методи и пр. използвани в Visual Basic.
4. Щракнете върху бутона ОК, за да затворите бутона Reference.
5. Поставете новия бутон някъде във формуляра.
6. Щракнете двукратно върху бутона, за да отворите прозореца за написване на програмния текст (код).
7. Въведете следния код в прозореца Code.

```
Private Sub Command1_Click()  
    '// Тази функция създава примерна страница на InDesign.  
    '// Деклариране на променливи  
    Dim myInDesign As InDesign.Aplication  
    Dim myDocument As InDesign.Document  
    Dim myFrame As InDesign.TextFrame  
    Dim myBounds As Variant  
    Dim myRestangle As InDesign. Restangle  
    Dim myImage As InDesign.Image  
    Dim myParagraph As InDesign. Paragraph  
    Dim myLargelImage As String  
    Dim mySmallImage1 As String  
    Dim mySmallImage2 As String  
    Dim mySmallImage3 As String
```

‘// Задава препрадки на променливи към изображенията, които ще съдържа страницата. Променливите се използват за опростяване на програмата. Ако имената или местоположенията на файловете трябва да бъдат променени, това лесно може да стане тук. Не трябва да претърсвате целия код, за да намерите и промените файловете един по един.

myLargelImage = *C:\Corbis\Are0097d.jpg*

mySmallImage1= *C:\Corbis\Are0073e.jpg*

mySmallImage2= *C:\Corbis\Are0061e.jpg*

mySmallImage3= *C:\Corbis\Are0067e.jpg*

‘//Необходимо е да промените имената и местоположенията на изображенията, с които ще разполагате и ще ползвате. За да разрешите съвместната работа на InDesign и Visual Basic трябва да създадете препратки към обекта InDesign.Application. Следната функция GetObject осигурява достъп на Visual Basic до обекта на InDesign.Application.

```
Set myInDesing = GetObject (“,  
    “InDesign.Aplication”)  
‘// Изображенията ще бъдат поставени в  
    страницата на InDesign.  
MyInDesing.UserInteractionLevel =  
    inDontDisplayAlert  
‘// Сега можете да инструктирате обекта да  
    създаде нов документ.  
Set myDocument = myInDesing.Document.Add
```

‘// След като сте създали нов обект в Visual Basic се нуждаете от команда, която ще се заеме с обекта, ще зададе неговите свойства и ще ползва неговите функции. Командата Set, се отнася до формирането на обекта. Функцията myInDesing.Document.Add създава нов документ и връща препратка (връзка) към него. Когато зададете myDocument = на тази препратка, Visual Basic ще знае, че ще ползвате новия документ навсякъде, където се използва myDocument. Приемете myDocument като псевдоним на новия документ.

ИМПОРТИРАНЕ И ПОМЕСТВАНЕ НА ГОЛЯМО ИЗОБРАЖЕНИЕ.

Поставете (импортирайте) голямо изображение в страницата. За да направите това, създайте правоъгълна рамка с желаните размери на изображението. Оразмеряването на правоъгълника става, чрез задаване на `GeometricBounds`. Координатите на свойството `GeometricBounds` се задават в следния ред: `Y1, X1, Y2, X2`.

```
myBounds = Array (“3p0”, “3p0”, “63p0”, “36p0”)
SetmyRestangle =
myDocument.Sprreads.Item91).Restangles.Add
myRestangle. GeometricBounds = myBounds
```

‘// сега поставете изображението вътре в
правоъгълника.

```
Set myImage = myRestangle.place (mylargeImage)
myImage. GeometricBounds =
myRestangle.VisibleBounds
myRestangle.Fit =ContentToFrame
myImage. GeometricBounds = myRestangle.
GeometricBounds
```


Когато изображението се поставя вътре в правоъгълника, може да не се преоразмери автоматично, за да се вмести в границите на правоъгълника. За да се вмести, трябва да зададете функцията Fit.

КОМПОЗИРАНЕ НА ТЕКСТ

Когато изображенията са поставени, необходимо е да се добави и текст в страницата:

Създаване на Горна текстова рамка (Малко заглавие):

```
myBounds = Array ("3p0", "12p0", "5p0",  
"36p0")
```

```
Set myFrame =  
myDocument.Spreads.Item(1).TextFrames.Add
```

```
myFrame.FillColor =  
myDocument.Swatches("paper")
```

```
myFrame.GeometricBounds = myBounds
```

```
myFrame.Text.TextContents = "InDesing  
Scripting"
```

With myFrame.Paragraphs.Item(1)

- . Font = “Arial Black”**
- . PointSize = 24**
- . Justification = inCenter**

End With

Променете шрифта, ако искате да използвате друг знаков стил. Ако няма зададен шрифт , InDesign използва по подразбиране Times Roman. Когато се изгражда страницата, InDesign прилага спецификация Auto, ако няма зададена стойност за тази настройка.

Цветът по подразбиране на текста е черен. За да стане текстът по-ясно видим на определен фон, се създават настройки с опциите - Gray Out Images на предпоченциите за Images Display. Когато пуснете за изпълнение този примерен скрипт, изображенията ще се появяват нормално. За да промените цвета на текста, използвайте инструментите, командите и другите средства за управление на цвета в InDesign.