بنام خداوند یکتا جزوه درس برنامه سازی پیشرفته 1 دانشگاه الزهرا(س) رشته نرم افزار محیط ویژوال بیسیک استاد : حسن کاسبی خیره مسجد زمستان 1393



# فهرست

8	فصل اول
9	روالها و توابع
10Fu	تعريف nction
11	تعريف Sub
12	آرگومانھا
دار 13	آرگومان،ای نام
14	خروج از روالها
15	توابع بازگشتی
ىتى 17	انواع توابع بازگش
ئشتى	مزایای توابع بازگ
19(Mo	ماژولها (odule
21(Class Mo	کلاسھا (dule
تصهها به کلاس	اضافه کردن مشخ

-

– اضافه کردن متدها به کلاس ها
طريقه فراخواني كلاس ها 24
فصل دوم
ورود به نرمافزار 27
آشنایی با محیط
ايجاد يک پايگاه داده يا DataBase
ايجاد جدول
کاربرد Null در طراحی جدول
ايجاد فيلد Identity
ايجاد كليد اصلي براي جدول
ذخيره كردن ساختار جدول طراحي شده
حذف ساختار جدول طراحي شده 40
ويرايش ساختار پايگاه داده
ايجاد دياگرام
ايجاد ارتباط بين جداول 44

45.	حذف ارتباط بين جداول
46.	ذخیره کردن دیاگرام
47.	حذف دياگرام
48.	کار با رکوردها در جدول
49.	ار تباط پایگاه داده ایجاد شده با ویژوال بیسیک
51.	فصل سوم
52.	کار با پایگاه داده
52.	ايجاد پروژه
64.	فصل چهارم
64 . 65 .	<b>فصل چهارم</b> متد pset
64 . 65 . 66 .	<b>فصل چهارم</b> متد pset متد Line
64 . 65 . 66 . 68 .	فصل چهارم. متد pset. متد Line. متد Circle.
<ul> <li>64</li> <li>65</li> <li>66</li> <li>68</li> <li>70</li> </ul>	فصل چهارم متد pset متد Line متد Circle متد Color
<ul> <li>64</li> <li>65</li> <li>66</li> <li>68</li> <li>70</li> <li>70</li> </ul>	فصل چهارم. متد pset. متد Line. متد Circle. متد Color. متد Font.
<ul> <li>64</li> <li>65</li> <li>66</li> <li>68</li> <li>70</li> <li>70</li> <li>71</li> </ul>	فصل چهارم. متد pset. متد Line. متد Circle. متد Color. متد Font. متد Microsoft Agent

فصل پنجم
منو ها 75
ايجاد منو
ايجاد منو با ويزارد
کاربرد (Menu Editor)
روش های باز کردن Menu Editor 77
تنظيم مشخصه هاي منو 77
مشخصه های متداول شی منو 77
اضافه کردن کلید های میانبر به گزینه ها
ايجاد منوهاي باز شو 79
79Popup Menu
آرايەھا
اعلان ارايه
اعلان ارايه مانند يک متغير
اعلان ارایه با کلید واژه To
تغيير تعداد عناصر آرايه

آرايه کنترلی
اضافه کردن عنصر ها به ارایه کنترلی در زمان اجرا
آرایه های چند بعدی در ویژوال بیسیک
فصل ششم
برنامه نویسی شیء گرائی object oreinted programming
توارث عمومي
توارث خصوصي
فصل هفتم
مفهوم پايداري
پروندهها برای ذخیره و بازیابی دادهها
انواع پروندهها
خواندن د و ندههای تر تسی



## روالها و توابع:

بعضی مواقع شما نیاز دارید که تیکه کدی را در اکثر فرمها استفاده کنید. بنابراین برای راحتی کار و منظمی و درهم نشدن کدها از روالها و یا از توابع استفاده می کنیم. Sub و Function هر دو برای این منظور هستند با این تفاوت که برای تعریف یک زیرروال یا (Subroutine) از Sub استفاده می کنیم و برای تعریف تابع از Function استفاده می کنیم. با این حال، تفاوت اصلی این دو در این است که تابع مقدار بازگشتی دارد ولی Sub اینطور نیست. توی تابع یک متغیر همنام با تابع هست که هر مقداری داخل آن بریزید، بعنوان خروجی بازگشت داده میشه. درنتیجه از خروجی Function می توانید توی عبارت های محاسباتی و... استفاده کنید ولی Sub را نمیشه در اینطور موارد استفاده کرد. وقتی روال را با Private تعریف می کنیم فقط در محدودهی فرم یا ماژولی که روال در آن تعریف شده و اشیاء داخل آن قابل فراخوانی است، اما وقتی با Public تعریف می کنیم در همهی فرمها قابل استفاده است.

#### تعريف Function:

Public | Private Functionنام تابع) تعريف پارامترها As (نوع داده خروجی

Commands

**Random = Commands** 

**End Function** 

و در آخر حتماً باید دستورات در داخل نام تابع ریخته بشوند. برای مثال:

Public Function Random (Max As Long) As Long

Random= Int (Rnd\*Max)+1

**End Function** 

برای فراخوانی این تابع در قسمت مورد نیاز باید نام تابع را به یک متغیر نسبت دهیم و سپس با آن همانند متغیر رفتار می کنیم. مثل:

Private Sub CmdRandom\_Click()

Max= InputBox ("Input Your Number")

A= Random (Max)

End Sub

### تعريف Sub:

(پارامترها) نام روال Public | Private Sub

Commands

**End Sub** 

برای نمونه:

Public Sub Random (A As Long)

A= Int (Rnd\*A)+1

Print A

End Sub

برای فراخوانی می توانیم از کلمه کلیدی و اختیاری Call استفاده کنیم.

Private Sub CmdRandom\_Click()

A= InputBox ("Input Your Number")

Call Random (A)

**End Sub** 

# آرگومانها

آرگومان ها متغیرهایی هستند که به عنوان جانگهدار برای مقادیری که به Function یا Sub ارسال می شوند، عمل می کند. آرگومان ها را داخل پرانتزهای دستور اعلان روال ها، می توانیم ایجاد کنیم.

نوع داده خروجی As (تعريف پارامترها) نام تابع Public | Private Function

Commands

**End Function** 

مثال برای تعریف آر گومان:

Public Function Random (Max As Long) As Long

Random= Int (Rnd\*Max) + 1

**End Function** 

مثال برای فراخوانی تابع و آرگومان مربوط به آن:

#### Private Sub CmdRandom\_Click ()

A= InputBox ("Input Your Number")

B= Random (A)

End Sub

**نکته:** استفاده از آر گومانها کارایی کد را افزایش میدهد.

برای استفاده از آرگومانها یکسان بودن نوع و ترتیب و تعداد آنها خیلی اهمیت دارد: اگر نوع مقدار با نوع آرگومان تعریفی یکسان نباشد برنامه خطا میدهد.

اگر آرگومان های تعریفی برای مثال 3 عدد بودند، باید 3 مقدار دریافت کنند در غیر اینصورت پیغام خطا صادر می شود.

# آرگومانهای نامدار

برای ساده تر شدن کار ما بعضی از مواقع از =: برای مقداردهی به آرگومان استفاده می کنیم. برای مثال:

تعریف تابع Random و آر گومان:

Public Function Random (Max As Integer)

Random= Int (Rnd\*Max)+1

**End Function** 

و فراخوانی تابع:

Private Sub CmdRandom\_Click ()

A= Random (Max:=10)

**End Sub** 

# خروج از روالها

بعضی مواقع میخواهیم که قبل از پایان روال، از روال خارج شویم. پس برای

انجام اینکار از Exit Function برای خروج از تابع و از Exit Sub برای خروج از

رويه استفاده مي كنيم.

توابع بازگشتی تا به الآن مسائلي ساده را پي گرفتيم. ولي در مسائلي همانند فاکتوريل بايد توجه داشته باشید که تا وقتی که عدد به صفر نرسیده تابع رو فراخوانی کنید. به عبارت ساده، تابعی که خود را فرابخواند تا به نتیجه نهایی برسد، تابع بازگشتی نامیده میشود. برای درك بهتر اين مسئله توجه كنيد: فرض کنید، میخواهیم فاکتوریل عدد 5 را محاسبه کنیم که میشود:  $5 \times 4 \times 3 \times 2 \times 1 = 120$ در اینصورت می توانیم این فرمول را هم برای این عدد بنویسیم: 120 = (فاكتوريل عدد 4) × 5 چون فاکتوریل عدد 4 نیز همان 4 × 3 × 2 × 1 میباشد که برابر با 24 می شود. برای محاسبه فاکتوریل 4 نیز می توان فرمول زیر را به کار برد:  $4 \times (3$ فاكتوريل عدد = 24و همین طور فرمول زیر را برای فاکتوریل عدد 3:

3 imes (2 فاكتوريل عدد ) = 6

و باز هم فرمول زیر را برای فاکتوریل عدد 2:

$$2 imes \left(1 
ight)$$
فاكتوريل عدد 1  $= 2$ 

و نهايتاً فاكتوريل عدد 1 كه برابر همان عدد 1 مي شود.

بنابراین می توان گفت که برای محاسبهی فاکتوریل یک عدد از ضرب همان عدد در فاکتوریل یک واحد کمتر از آن، نتیجه را می توانیم به دست آوریم. حال اگر این فرمول را به صورت یک تابع پیادهسازی کنیم، به آن تابع بازگشتی گفته می شود.

Function Fact (Num As Integer) As Integer

If Num > 1 Then

Fact = Num \* Fact (Num-1)

Else

Fact = 1

End If

**End Function** 

در این مثال، مقدار پارامتر (متغیر Num) بررسی شده و در صورتی که از عدد یک بزرگتر باشد، مقدار برگشتی تابع حاصل ضرب همان عدد در فاکتوریل یک واحد کمتر از آن خواهد شد؛ در غیر اینصورت (اگر برابر با یک باشد)، مقدار بر گشتی تابع نیز یک میشود.

> **انواع توابع بازگشتی** 1. توابع بازگشتی مستقیم

2. توابع بازگشتی غیرمستقیم

1) اگر یک تابع خودش را فراخوانی کند از نوع تابع باز گشتی مستقیم هست. همانند مثال فوق.

2) اگر تابع دیگری مثلاً به نام Number تابع Fact را فراخوانی کند و تابع Fact هم تابع Number را فراخوانی کند، در اینصورت تابع باز گشتی از نوع غیرمستقیم خواهد بود، چون تابع Fact خودش را به طور غیرمستقیم فراخوانی کرده است.

برای مثال:

Public Function Fact (Num As Integer)

Commands

Fact = Number (Num)

**End Function** 

Public Function Number (IntNum As Integer)

Commands

Number = Fact (IntNum)

**End Function** 

## مزایای توابع بازگشتی:

1. كارايي برنامه بالا رفته و از كدنويسي كمترى استفاده مي شود.

2. خوانایی برنامه سخت میشود (برای همین اکثر برنامهنویسهای حرفهای که

مي خواهند كدهايشان لو نرود از توابع باز گشتي استفاده مي كنند.

## ماژولها (Module)

برای اینکه یک تیکه برنامه در همه جای برنامه قابل استفاده باشد از ماژولها استفاده می کنیم. اگر در فرم تابع یا روالی را تعریف کنیم، آن تابع یا روال فقط در همان فرم قابل استفاده است و اگر بخواهیم آن تابع یا روال را در فرم دیگری استفاده کنیم باید فرم مربوط به روال خوانده شود.

فرض کنید این تابع در یک فرم نوشته شده است:

Private Sub Form1\_Load ()

#### Public Function Random (Max As Long) As Long

Random= Int (Rnd\*Max)+1

**End Function** 

**End Sub** 

حال برای فراخوانی این تابع در یک فرم دیگر از این دستور استفاده می کنیم:

Private Sub CmdRandom\_Click()

Form1.Random

**End Sub** 

اما این کار کمی دشوار به نظر می آید و برای خوانده شدن روال باید اول Form1 خوانده شود. پس برای راحتی و منظمی و نیاز نداشتن به خوانده شدن فرم مربوط به تابع یا روال، ما از ماژول ها استفاده می کنیم. به این صورت که روی پنجره Project کلیک راست کرده و به این آدرس می رویم:

Add => Module => Open

حالا می توانیم کدی را که در Form1 نوشتیم را در قسمت General در پنجره Module بنویسیم. در اینصورت براحتی می توانیم در رویداد Click در CmdRandom از متد تولید شده استفاده کنیم به این صورت که:

Private Sub CmdRandom\_Click()

Max= InputBox ("Input Your Number")

A= Random (Max)

Print A

End Sub

کلاس بیشتر شبیه به ماژول است ولی با امکانات اضافی، و می توان گفت که مجموعه ای از مشخصهها و متدها است. داخل ماژول ما فقط می توانیم توابع و APIها رو قرار بدهیم و هنگام نیاز صداشون بزنیم. ولی داخل کلاس علاوه بر این مسایل میتوانیم Event تعریف کنیم.

كلاسها (Class Module)

برای Class ساختن باید در قسمت Project کلیک راست کنیم سپس به این آدرس می رویم:

Add => Class Module اجزای کلاس می توانند به صورت Private یا Public تعریف شوند. متغیرهای Private فقط توسط متدهای کلاس قابل استفادهاند، اما متغیرهای Public علاوه بر متدهای کلاس توسط سایر زیربرنامههای موجود در برنامه نیز قابل استفادهاند.

## اضافه کردن مشخصهها به کلاس:

اگر از منوی Tools گزینه Add Procedure را انتخاب کنیم، گزینه Random) را انتخاب کنیم، گزینه را Random) را انتخاب کرده و در کادر متن Name به عنوان مثال نام (Random) را بنویسیم و سپس OK را بزنیم روال های Get Random و Let Random به برای کدنویسی ما حاضر می شوند.

#### Public Property Get Random() As Variant

Commands

**End Property** 

#### Public Property Let Random(ByVal vNewValue As Variant)

#### Commands

#### **End Property**

روال Get Random برای خواندن مقدار مشخصه Random است و روال

Random برای مقداردادن به مشخصه Random می باشد. در این قسمت باید توجه

داشت که در پایان متدهای Get و Let تعریف شده، As Variant داریم. پس برای

اینکه بتوانیم نوع داده های مورد نظر را عوض کنیم، مثلاً در اینجا Variant را به

Integer تبديل مي كنيم.

Public Property Get Random() As Integer

Random=Num

End Property

Public Property Let Random(ByVal vNewValue As Integer)

Num= VNewValue

End Property

#### - اضافه کردن متدها به کلاسها:

برای اضافه کردن متد به کلاس نیز از همان گزینه Add Procedure در

منوی Tools کمک می گیریم به اینصورت که ایندفعه به جای اینکه از گزینه

Property استفاده کنیم از گزینه یSub یا Function استفاده می کنیم و Name

را با توجه به کاری که میخواهیم انجام بدهیم انتخاب میکنیم. مثلاً من اینجا از نام

Random و گزینه Function استفاده کردم و کد زیر در کلاس ماژول برای من

نوشته می شود.

Public Function Random()

**End Function** 

حال برای تکمیل این متد می توانیم کد زیر را بنویسیم:

Public Function Random (Max As Long) As Long

Random= Int (Rnd\*Max)+1

**End Function** 

طريقه فراخواني كلاسها:

دو روش برای استفاده از این کلاس ها وجود دارد:

روش اول:

نام کلاس As New نام شیء Public

روش دوم:

ابتدا باید نام شیء را یک کلاس تعریف کنیم همانند اعلان متغیر. مانند:

نام کلاس As نام شیء Dim

سپس کلاس را داخل شیء میریزیم همانند مقداردهی به یک متغیر. مانند:

نام کلاس New = نام شیء Set

بهتر است وقتی که کار ما با شیء تمام شد، حافظهی استفاده شده توسط شیء را آزاد

سازيم به اين صورت كه:

Nothing = نام شیء Set



## ورود به نرمافزار:

برای ورود به محیط SQL Server با هر ویندوزی که هستید باید از قسمت

نرمافزارها گزینهی SQL Management Studio را انتخاب کنید و منتظر بمانید تا

محيط SQL Server باز شود.



در قسمت بعدی به پنجرهی تشخیص هویت برخورد میکنیم که در این قسمت باید

Server Type یعنی نوع سرور و Server Name یعنی نام سرور و

Authentication که تصدیق را مشخص می کند را تعیین می کنیم.

SOL	Server 2008 R2	
Server type:	Database Engine	~
Server name:	(local)	~
Authentication:	Windows Authentication	~
User name: Password	user\symbol	~
	Remember password	

من برای نمونه این گزینهها را پیشفرض در نظر می گیرم و Connect را میزنم.

آشنایی با محیط

وقتی که وارد SQL Server Management Studio می شویم، به قسمت های مختلف این محیط دسترسی داریم. برای نمایش پنجره های مختلف این نرمافزار می توانید از منوی View پنجره ی مورد نظر خود را انتخاب کنید اما مهمترین پنجره هایی که مورد نیاز ما هست پنجره های Object Explorer و منوی New پنجره هایی که مورد نیاز ما هست پنجره ی Object Explorer و منوی و منوی از جمله دیتابیس ها، جداول و... می باشد و منوی New Query برای تایپ و اجرای دستورات SQL می باشد.





## ایجاد یک پایگاه داده یا DataBase!

برای ایجاد یک پایگاه داده باید در پنجرهی Object Explorer روی پوشهی

DataBase راست کلیک کرده و گزینه ی New Database را انتخاب می کنیم.

R	
File Edit View De	ebug Tools Window Community
😫 New Query 📄 🛉	🔥 😘 🚯 🕼 💕 🖬 🕘 🚳 💂
Object Explorer	<b>→</b> ‡ X
Connect 🕶 📑 📑	Y 🛃 🍒
<ul> <li>Iocal) (SQL Serve</li> <li>Databases</li> </ul>	r 10.50.1600 - user\symbol)
🕀 🧰 Security	New Database
<ul> <li></li></ul>	Attach Restore Database Restore Files and Filegroups
	Start PowerShell
	Reports +
	Refresh

با انجام این کار یک کادر محاورهای به صورت زیر باز میشود که در کادر

Database Name باید نام پایگاه داده را تایپ کنیم که من در این مثال نام Test

را تايپ كردەام.

8	New Database — 🗖 💌				
Select a page	🔄 Script 👻 📑	Help			
Poptions Filegroups	Database <u>n</u> ame: <u>O</u> wner:	idevina	Test <default></default>		
	Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth
	Test	Rows	PRIMARY	3	By 1 MB, unrestricted growth
	Test_log	Log	Not Applicable	1	By 10 percent, unrestricted growt
Connection					
Server: (local)					
Connection: user\symbol					
View connection properties					
Progress					
Ready	٢			<u>A</u> dd	> <u>R</u> emove
					OK Cancel

هر پایگاه داده شامل حداقل دو فایل میباشد که قابل حذف شدن نیستند؛ این فایلها

عبارتند از:

فایل اصلی (MDF) حاوی اطلاعات راهاندازی بانک اطلاعاتی است. این فایل برای ذخیره دادهها نیز به کار میرود.

فایل سابقه تراکنش (LDF)، اطلاعاتی را برای ترمیم بانک اطلاعاتی در اثر خرابی، ذخیره می کند. **نکته:** برخی از بانکهای اطلاعاتی ممکن است آنقدر بزرگ باشند که به چند فایل کمکی نیاز داشته باشند تا در دیسکهای حافظه جداگانه نگهداری و استفاده شوند. این فایلها، فایلهای کمکی نام دارند.

با کلیک بر روی OK در زیر گزینه های Database پایگاه داده ای به نام

TEST ايجاد مي شود.

### ایجاد جدول:

برای ایجاد جدول باید با کلیک بر روی + که در کنار نام پایگاه داده قرار دارد، زیرپوشههای پایگاه داده را باز کنیم و سپس با راست کلیک بر روی پوشهی Tables و انتخاب گزینهی ...New Table می توانیم یک جدول ایجاد کنیم.



با انتخاب این گزینه جدولی به شکل زیر ایجاد می شود. و 2 پنجره Properties و

Column Properties نیز به همراه جدول مشاهده می شود.

Microsoft SQL Server Management Studio			_ 🗆 🗡
Tools Window Community Help			
JSER.TEST - dbo.Table_1	<del>~</del> ×	Properties	<b>▼</b> ₽
Column Name Data Type Allow Nulls		[Tbl] dbo.Tab	le 1
		8141 6	-
		(Identity)	
		(Name)	Table 1
		Database N	lai TEST
		Description	1
		Schema	dbo
		Server Nam	ne user
		Table Designation	gner
		Identity Co	lu
		Indexable	Yes
		Lock Escala	iti Table
		E Regular Dat	
		Replicated	INO Ca
		Text/Image	
olumn Properties	7		
		(Identity)	

این قسمت برای طراحی جدول میباشد.

**کاربرد Null در طراحی جدول:** 

وقتی که مشخصهی Null از یک فیلد تیک دار شود، هنگام اضافه کردن رکورد جدید، می توانید این فیلد را خالی رها کنید ولی اگر تیک یک فیلد را بردارید هنگام مقداردهی کردن به هیچ عنوان نمی توانید فیلد را خالی رها کنید.

**نکته:** این گزینه برای کلیدهای اصلی و Identity همیشه غیرفعال میباشد. چون کلید اصلی هیچگاه نمی تواند خالی رها شود و همچنین فیلدی که Identity میباشد به طور خودکار مقداردهی میشود.

#### ایجاد فیلد Identity:

فیلد Identity همانند فیلد Auto Number در Access میباشد. یعنی بطور اتوماتیک مقداردهی عددی میکند. برای این کار بعد از انتخاب فیلد مورد نظر (که از نوع int تعریف شده است) در کادر تنظیمات گزینه Is Identity را برابر با Yes قرار میدهید.
Microsoft SQL Server Management Studio	_ 8 _>
Tools Window Community Help	
JSER.TEST - dbo.Table_1*	✓ × Properties ✓ ↓
Column Name Data Type Allow Nulls	[Tbi] dbo.Table 1
	Image:
olumn Properties	
Has Non-SQL Server Subscriber No	<b>^</b>
Identity Specification Yes	
Usidehoty) Yes	×
Identity Incenter 1 Identity Sould I	
Austruty Seed 1 Tridexable Vec	
In Columpat In	~

گزینهی Identity Increment مشخص می کند که با هر بار اضافه شدن رکورد جدید، چه مقدار به عدد قبلی اضافه شود و گزینهی Identity Seed نیز برای تعیین اولین مقدار این فیلد است. درواقع مشخص می کند که فیلد مورد نظر از چه عددی شروع شود.





### ذخيره كردن ساختار جدول طراحي شده

بعد از طراحی برای ذخیره کردن ساختار جدول طراحی شده از کلیدهای

ترکیبی Ctrl+S می توانیم استفاده کنیم. که با استفاده کردن از این کلیدها یک کادر

محاورهای به شکل زیر برای ما باز می شود.

	Choose Name	? ×
Enter a name for the tal	ole:	
Tb_1		
	ОК	Cancel

حال نام مورد نظر را تایپ کرده و OK را میزنیم.

#### حذف ساختار جدول طراحي شده

برای حذف جدول باید روی جدول مورد نظر راست کلیک کرده و گزینهی

Delete را انتخاب کنیم و سپس در کادر محاورهای ایجاد شده OK میزنیم.



#### ویرایش ساختار پایگاه داده

برای ویرایش ساختار پایگاه داده پس از ایجاد، باید روی جدول مورد نظر

راست کلیک کنید و بعد گزینهی Design را انتخاب کنید.



#### ایجاد دیاگرام

برای ارتباط بین جداول از دیاگرامها استفاده میکنیم. برای این کار باید در

پایگاه دادهی مورد نظر روی پوشهی Database Diagrams راست کلیک کرده و

گزینه ی New Database Diagram را انتخاب می کنیم.



Item(s) Saved

با انتخاب این گزینه کادر محاورهای باز می شود که در آن می توانیم جدول های مورد نیاز برای ارتباط را انتخاب کنیم.

	Add Table	?	×
Tables			
TB_2 TB_3			
	Refresh Add	Clos	e

با انتخاب جدولها و کلیک کردن روی گزینهی Add می توان جداول را به دیاگرام

اضافه کرد. حال در این دیاگرام جدولهای انتخاب شده را بدون ارتباط داریم:

USER.TEST - Diagram_0*		• ×
D code	TB_2 10762 192	
	TB_3 (db3 TB3	
•		

#### ايجاد ارتباط بين جداول

برای ایجاد ارتباط باید بر روی کلید اصلی کلیک کرده و آن را بر روی کلید

خارجی جدول دیگری در گ کرده تا کادر محاورهای زیر باز شود:

•	Tables and Columns ? ×		
Selected R	Relationship name: FK_Tb_3_Tb_11		lbe
	Primary key table:	Foreign key table:	
	Tb_1	▼ Tb_3	
	ID	ID	
		ОК	Cancel

حال کلید اصلی و کلید خارجی را چک کرده و در صورت صحیح بودن روی OK کلیک می کنیم تا اینکه ارتباط ایجاد شود و بعد از ایجاد ارتباطات صفحهای به شکل زیر داریم:



## حذف ارتباط بين جداول

برای حذف ارتباط بین جداول بر روی خطهای ارتباط کلیک راست کرده و

گزینهی Delete Relationship From Database را انتخاب می کنیم.



# **ذخیره گردن دیاگرام** حال برای ذخیره کردن دیاگرام ایجاد شده از کلیدهای ترکیبی Ctrl+S استفاده میکنیم و در کادر محاورهای بازشده نام دیاگرام را انتخاب کرده و OK را انتخاب میکنیم.

Choo	ose Name	×
Enter a name for the diagram	:	
	ОК	Cancel:

#### حذف دیاگرام

برای حذف دیا گرام در پوشهی Database Diagrams، بر روی دیا گرام

مورد نظر راست کلیک می کنیم و گزینه ی Delete را انتخاب می کنیم.



سپس از کادر محاورهای ایجاد شده OK را میزنیم.

×	D	elete Object			-	- 🗆 🗙
Select a page	🔄 Script 👻 🚺 Help					
	Object to be deleted					
	Object Name	Object Type	Owner	Status	Message	
	Diagram_0	DatabaseDiagr				
Connection						
Server: (local)						
Connection: user\symbol						
View connection properties						
Progress						
Ready						
					OK	Cancel

#### کار با رکوردها در جدول

برای اینکه رکوردی را در جدول اضافه کنیم باید روی جدول مورد نظر راست کلیک

کنیم و سپس گزینهی Edit Top 200 Rows را انتخاب می کنیم.



حال جدول برای اضافه کردن، ویرایش و یا حذف رکورد آماده می شود.

#### ارتباط پایگاه داده ایجاد شده با ویژوال بیسیک





حال مابقی کارها اعم از حذف و جستجو و یا ذخیرهی داده توسط ویژوال بیسیک (مدیریت پایگاه داده توسط محیط ویژوال بیسیک) همانند مدیریت Access میباشد.

برای مثال برای اضافه کردن رکورد به پایگاه داده خط دستور زیر استفاده می کنیم:

Adodc1.RecordSet.AddNew



#### کار با پایگاه داده

تقریبا تمام برنامه های کاربردی نیاز به دسترسی به داده ها دارند. برای برنامه های کاربردی که در یک کامپیوتر اجرا می شوند دسترسی به داده ها و پیاده سازی آن ساده است و نیاز به برنامه نویسی کمی دارد.

#### ايجاد پروژه

یک پروژه جدید باز کنید و از منوی Project گزینه ...Components رو

انتخاب کنید و گزینه های (OLEDB) Microsoft ADO Data Control 6.0 و گزینه های (OLEDB) و Microsoft ADO Data Control 6.0 (OLEDB) رو علامت بزیند و کلید OK

رو بزنید تا به ToolBox اضافه بشن.



Component	s ×	
Controls Designers Insertable Objects		General
<ul> <li>☐ IconListToolPriv</li> <li>☐ iptActiveXLib</li> <li>☐ MeetingJoinAx 1.0 Type Library</li> <li>☑ Microsoft ADO Data Control 6.0 (OLEDB)</li> <li>☐ Microsoft Calendar Control 12.0</li> <li>☐ Microsoft Chart Control 6.0 (OLEDB)</li> <li>☐ Microsoft Comm Control 6.0</li> <li>☐ Microsoft Common Dialog Control 6.0</li> <li>☐ Microsoft Data Bound List Controls 6.0</li> <li>☑ Microsoft DataGrid Control 6.0 (OLEDB)</li> <li>☐ Microsoft DataGrid Control 6.0 (OLEDB)</li> <li>☐ Microsoft DataList Controls 6.0 (OLEDB)</li> <li>☐ Microsoft DataRepeater Control 6.0 (OLEDB)</li> <li>☐ Microsoft External Item Picker</li> </ul>	▲ Browse	
Microsoft DataGrid Control 6.0 (OLEDB) Location: C:\Windows\SysWow64\MSDATGRI OK	D.OCX Cancel Apply	

در این مرحله شما باید یک پایگاه داده هاتون رو به شئ ADO ربط بدین برای

اینکار ابتدا روی شئ ADO راست کلیک کنید و گزینه ADODC Properties رو

انتخاب كنيد.

Property Pages
General Authentication RecordSource Color Font
Source of Connection
C Use Data Link File
Browse
C Use ODBC Data Source Name
New
Use Connection String
Build
Other Attributes:
OK Cancel Apply Help

در کادری که باز میشه (Property Pages) گزینه (Data) گزینه Data) (Data میشه String Microsoft.ACE.OLEDB.12.0 کلیک کنید. پنجره دیگهای باز میشه Microsoft.ACE.OLEDB.12.0 رو انتخاب و کلیک Next در اون پنجره باید گزینه Next یشید. در این مرحله باید آدرس یک پایگاه دادهها رو تعیین کنید. حالا در پایین کادر یک کلید به نام Test آدرس یک پایگاه دادهها رو تعیین کنید. حالا در پایین کادر یک کلید به نام م موفقیّت آمیز بوده یا نه. اگه پیغام Link Connection Succeeded به نمایش دراومد یعنی همه چیز درسته؛ روی کلیک کنید و دوباره NK رو بزنید تا زمانیکه دراومد یعنی همه چیز درسته؛ روی کلیک کنید و دوباره NK رو بزنید تا زمانیکه

انتخاب این گزینه با توجه به نسخه Office که استفاده میکنید. این گزینه برای Office که Office میکنید. این گزینه برای Office 2007

	Da	nta Link P	roper	ties		×
Provider	Connection	Advanced	AI			
Select t	ne data you wa	nt to conne	ct to:			
OLE	DB Provider(s)	)				^
Micr	osoft Jet 3.51 ( osoft Jet 4.0 O	DLE DB Prov	vider ider			
Micr	osoft Office 12 osoft Office 15	0 Access D 0 Access D	atabase atabase	Engine OLE Engine OLE	DB Pro DB Pro	
Micr	osoft OLE DB I	Provider for	Analysis	Services 11	.0	
Micro	osoft OLE DB I	Provider for	Analysis	Services 9.0	)	
Micr	osoft OLE DB I osoft OLE DB I	Provider For Provider for (	Data Mi ODBC D	ning Service	s	
Micro	osoft OLE DB I	Provider for	OLAP Se	ervices 8.0		
Micr	osoft OLE DB I	Provider for	Oracle			
Micr	osoft OLE DB I	Provider for	Search			
Micr	osoft OLE DB I	Provider for	SQL Ser	ver		
Micro	osoft OLE DB S	Simple Provi	der		>	×
				N	lext >>	
		ОК	0	Cancel	Hel	p

در این قسمت ادرس پایگاه داده را وارد میکنیم.

<b>9</b> 01	Data I	ink Pror	ortios	×
<u>21</u>	Data		Jeilles	
Provider	Connection Adv	anced All		
Specify th	e following to con	nect to this	data:	
1. Enter	the data source a	and/or locat	ion of the data:	
Da	ta Source:	D:\db\da	ata.accdb	
Lo	cation:			
2 Enter	information to log	on to the s	erver:	
0	Use Windows N1	Integrated	security	
۲	Use a specific us	er name an	d password:	
	User name:	Admin		
	Micr	osoft Da	ata Link	×
3. Er	<b>1</b> Test	connectio	n succeeded.	~
			ОК	
	(	Ж	Cancel	Help

حالا دوباره روی ADO راست کلیک کنید و گزینه آخر رو انتخاب کنید و در کادری که ظاهر شده، برگه سوّم یعنی RecordSource و انتخاب کنید و در قسمت Command Type روی ComboBox کلیک کنید و گزینه 2 – adCmdTable رو انتخاب کنید که با اینکار ComboBox پایینش فعّال میشه و

باید در اون قسمت جدول Student رو انتخاب کنید و کلید OK رو بزنید.

Property Pages ×
General Authentication RecordSource Color Font
RecordSource
Command Type
2 - adCmdTable
Table of Charad Dependent Name
Table or Stored Procedure Name
student
Command Text (SQL)
$\sim$
OK Cancel Apply Help

روی شئ DataGrid کلیک کنید و در جعبه Properties (سمت راست) گزینه DataSource رو پیدا کنید و روش کلیک کنید، باید بهطور خودکار گزینه Adodc1 آشکار بشه، روش کلیک کنید تا انتخاب بشه حالا دوباره روی شئ DataGridراست کلیک کنید و گزینه Retrieve Fields رو بزنید تا شئ DataGrid به طور خودکار خودش رو با جدول Student تطبیق بده. حالا برنامه رو اجرا کنید، برای حذف فیلدهای اضافی روی DataGrid راست کلیک کنید و گزینه Edit رو انتخاب کنید و فیلدهای لازم ندارین رو حذف کنید (راست کلیک روی فیلد و انتخاب گزینه Delete) و بعد یک جای خالیه فرم کلیک کنید تا تغییرات اعمال بشه. حالا دوباه برنامه رو اجرا کنید.

Properties - D	ataGrid1	×
DataGrid1 DataGrid		-
Alphabetic Categorized		
BorderStyle	1 - dbgFixedSingle	~
Caption		
CausesValidation	True	
ColumnHeaders	True	
DataMember		
DataSource	Adodc 1	
DefColWidth	0	
DragIcon	(None)	
DragMode	0 - vbManual	
Enabled	True	
Font	MS Sans Serif	
ForeColor	&H8000008&	
HeadFont	MS Sans Serif	
HeadLines	1	
Height	1455	
HelpContextID	0	v



ب برای روی DataGrid کلیک کنید و در قسمت Properties DataGrid کلیک کنید و در قسمت DataGrid رو DataGrid رو AllowUpdate رو AllowDelete رو True گزینه های AllowUpdate ، AllowAddNew و AllowUpdate Update و Update قرار بدید. با اینکار امکان افزودن رکورد جدید، پاک کردن یک رکورد و Update

Properties - D	ataGrid1	×
DataGrid1 DataGrid		-
Alphabetic Cate	gorized	
(Custom)		~
(Name)	DataGrid1	
Align	0 - vbAlignNone	
AllowAddNew	False	
AllowArrows	True	
AllowDelete	False	
AllowUpdate	True	
Appearance	1 - dbg3D	
BackColor	8H8000005&	
BorderStyle	1 - dbgFixedSingle	
Caption		
CausesValidation	True	
ColumnHeaders	True	
DataMember		
DataSource	Adodc1	
DefColWidth	0	¥
<u> </u>		

#### Caption

Sets/returns grid caption text

اگه میخواین فیلدهاتون رو به Textboxمتصل کنید.

Properties - Text1		×
Text1 TextBox		-
Alphabetic Cate	gorized	
(Name)	Text1	~
Alignment	0 - Left Justify	
Appearance	1 - 3D	
BackColor	&H8000005&	
BorderStyle	1 - Fixed Single	
CausesValidation	True	
DataField	stu_code 🔹 👻	
DataFormat		
DataMember		
DataSource	Adodc1	
DragIcon	(None)	
DragMode	0 - Manual	
Enabled	True	
Font	MS Sans Serif	
ForeColor	&H8000008&	
Height	975	
HelpContextID	0	
HideSelection	True	
Index		
Left	5400	
LinkItem		
LinkMode	0 - None	
LinkTimeout	50	
LinkTopic		
Locked	False	~

حالا اگه بخواین به پایگاه داده، داده اضافه کنید کد زیر را اضافه کنید.

اضافه كردن ركورد

Adodc1.Recordset.AddNew

ذخیره داده های وارد شده

Adodc1.Recordset.Update

حالا اگه بخواین پایگاه داده تون فیلتر کنین کد زیر را اضافه کنید.

" \* & Text1.Text & " ' '' & Text1.Text & " ' ''

براي جابهجايي بين ركوردها:

If Not Adodc1.Recordset.BOF Then

Adodc1.Recordset.MovePrevious

End If

If Not Adodc1.Recordset.EOF Then

Adodc1.Recordset.MoveNext

End If

Adodc1.Recordset.MoveFirst

آخرين ركورد

اولين ركورد

Adodc1.Recordset.MoveLast

حذف ركورد

Adodc1.Recordset.delete

تازەسازى ركورد

Adodc1. refresh

حالا اگه بخواین پایگاه داده تون جستوجو کنین کد زیر را اضافه کنید.

b = Adodc1.Recordset.Bookmark

Adodc1.Recordset.Find Trim("stu\_code= ' ") &

Trim(txt\_find.Text) & " ' ", , , 1

If Adodc1.Recordset.EOF Then

Adodc1.Recordset.Bookmark = b

"این شماره دانش آموزی وجود ندارد", , "جستجو"» MsgBox

End If

برای علامت گذاری رکورد معینی میباشد:

b = Adodc1.Recordset.Bookmark

به سمت اولين ركورد:

Adodc1.Recordset.Find Trim("stu\_code= ' ") & Trim(txt\_find.Text) & " ' ", , , 1



متد pset

این متد، برای نمایش دادن یک نقطه روی فرم یا برای تعیین نقطه شروع یک شکل بر روی فرم کاربرد دارد. نحوه استفاده از این متد بصورت زیر هست:

اسم فرم یا عنصر [color] (x,y), اسم فرم یا عنصر (x,y)

با استفاده از دستور بالا، می تونید یک نقطه روی فرم یا هر عنصری که خواستین رسم کنید. X و Y در دستور بالا مختصات مرکز نقطه ،و Color رنگ نقطه رو مشخص می کنه. درضمن در دستور بالایی، یک مقدار Step هم بعد از نام متد وجود دارد. اگر از این پارامتر استفاده شود، مکان رسم نقطه نسبت به مکان فعلی خواهد بود.در آرگومان Color می تونید، یک کد رنگ در مبنای 16 را وارد کنید یا یکی از دستورهای زیر را بنویسید:

VbRed – VbGreen – VbBlack – VbYellow – VbCyan , ... مختصات نقطه به دو روش تعيين مي شود:

1- به صورت مطلق بدون [step]

2 – به صورت نسبی با استفاده از [step] که مختصات نقطه را با توجه به فاصله آن از موقعیت فعلی مکان نما تعیین می کند.

مثال :

Form1.Pset (100,200)

Form1.Pset (10,50) VbRed

Form1.Pset Step (20,30) VbBlue

#### متدLine

با استفاده از این متد، می توانید در ویژوالبیسیک خط راست، بین دو نقطه رسم کنید. برای این کار باید از دستورات زیر استفاده کنید:

اسم فرم يا [color], [B | BF], (X2,Y2) – (X2,Y2).

عنصر

در دستور بالا مقدارهای پارامتر اول یا همان X1 , Y1 مختصات نقطه شروع خط و مقدارهای داخل پرانتز دوم یعنی (X2 ,Y2) مختصات نقطه دوم هستن که خط باید بین این دو نقطه رسم شود. پارامتر Color مثل دستور قبلی رنگ خط را مشخص می کند. اما در این دستور، دو پارامتر دیگه به عنوان B و BF و جود دارد .در صورتی که بعد از انتخاب رنگ پارامتر B در دستور نوشته شود، یک مستطیل از مختصات داده شده رسم می شود. به عبارتی، مختصات خط وارد شده قطر بزرگ مستطیل خواهد بود و در صورتی که پارامتر BF در دستور نوشته شود، یک مستطیل توپر رسم میشود و با رنگ وارد شده در دستور پر می شود. برای یادگیری بهتر به دستورات زیر دقت کنید:

form1. Line (50,50)-(750,500)

Form1.Line (20, 30)-(500, 700), vbBlue, BF Form1.Line (100, 100)-(600, 600), vbRed, B دستور اول ،یک خط با مختصات داده شده در فرم به رنگ سیاه رسم می کند. دستور دوم، یک مستطیل توپر، به رنگ آبی با مختصات داده شده رسم می کند و درنهایت دستور سوم، یک مستطیل توخالی به رنگ قرمز، در مختصات داده شده رسم می کند.

#### متد Circle

برای رسم کردن دایره، در ویژوالبیسیک از این متد استفاده می شود. این متد علاوه بر رسم دایره، می تواند قسمتی از یک دایره و یا بیضی رسم کند. نحوه استفاده از این متد بصورت زیر هست:

#### اسم فرم یا عنصر Circle [step] (X,Y) , R, [color],,,A اسم فرم یا

آرگومان [step] باعث تعیین مکان نسبی مرکز دایره خواهد شد. پارامتر بعدی یا همون R شعاع دایره را مشخص میکند.

مختصات (X, y) مرکز دایره را مشخص میکند. پارامتر R شعاع دایره برحسب پیکسل است.

و پارامتر Color رنگ دایره را مشخص می کند. پارامتر آخر یا همان A (که نسبت قطرعمودی به قطر افقی است)

اگر A1 باشد (حالت پیش فرض) یک دایره رسم خواهد شد.

در صورتی که مقدار A عددی کوچکتر یا بزرگتر از 1 باشد ،شکل بیضی رسم میشود.

Form1.circle(1000,1250),400,,,,0.2 Form1.circle(1750,1250),400,,,,1 مثال:

#### متد Color

برای فعال کردن کادر color از کنترل CommonDialog1 و سپس

متد showcolor استفاده می شود.

# CommonDialog1.ShowColor Myfrom.back=CommonDialog1. Color

**متد** Font

برای مدیریت قلم ها، مقداردهی مشخصه flags است، که مشخصه flags

میتواند یکی از سه ثابت زیر را داشته باشد:

cdlCFScreenFonts= قلم صفحه نمایش 1

cdlCFPrinterFonts=قلم چاپگر2

cdCFBoth=هردو مجموعه3

CommonDialog1.ShowColor

Microsoft Agent چیست؟

Microsoft Agent یک ActiveX است که می شود با آن یک شخصیت رو کنترل کرد تا یک سری فرمانها و صحبتهای ما رو به گوش کاربر برساند. اگر از محصولات Office استفاده کرده باشید حتماً آن کلیپی رو که برایتان ادا درمیاورد را دیدین. بله به آن می گویند Agent. حالا می خواهیم بدانیم که چطور می شود از آن در برنامه های خودمان استفاده کنیم.

یکی از بهترین Agentهای طراحی شده Merlin نام داره که اعمال زیادی را برایتان انجام میدهد.

> **برنامه نویسی Microsoft Agent** ابتدا شما باید Agent را در Event فرمتون صدا بزنید.

# Agent1. Characters .Load "merlin" Agent1. Characters . Characters ("merlin").show

بعد از این دستور Merlin نمایش داده میشه حالا چیکارش کنیم؟؟!! اگه شما Microsoft Text2 Speech Engine را در ویندوزتان نصب کرده باشید می توانید به Agent بگویید که برایتان صحبت کند. پس به طریق زیر عمل کنید: Agent1.Characters.characters("merlin").speak("salam")

برای اینکه توسط Agent خودتان فقط یک سری اطلاعات رو بصورت متن به کاربر نشون بدهید می توانید از دستور زیر برای اینکار استفاده کنید که با استفاده از دستور زیر یک کادر شبیه به یک بالون بالای سر Agent باز می شود که متن مورد نظرتان را در آن کادر می نویسد:

# Agent1. Characters. Character ("merlin"). Think "I'm thinking....hmm...E=MC2..Eureka!"

حال برای مخفی کردن آن می توانید از دستور زیر استفاده کنید:

#### Agent1. Characters. Character ("merlin"). Hide

Merlin برای اعمال زیادی حرکات خاصی داره که بسیار جالب هستند. مثلا برای

Merlin مى توانيد از Search, Idle, Think, Greet, Process...

بخواهيد كه برايتان ادا دربياورد.

برای اینکار باید از دستور زیر استفاده کنید:

Agent1.Characters.Character("merlin")."animationn ame"
در دستور بالا بجای animationname باید نام حرکتی که میخواهید برایتان انجام بدهد رو بیارید که برای هر Agent متفاوت هست. البته یک سری اعمال مثل اون چیزهایی که در بالا گفتیم تقریبا تو همه Agentها در نظر گرفته شده. شما خودتان هم میتوانید برای کار خودتان یک Agent طراحی کنید که برای اینکار میتوانید از نرمافزارهایی که برای اینکار وجود دارند استفاده کنید.



متداول ترین ویژگی قابل مشاهده برای برنامه کاربردی است که استفاده از برنامه را ساده تر خواهد کرد. با وجود منو محیط کار برای کاربران طبیعی و آشنا خواهد بود.

تعريف	ویژگی
استفاده از یک یا دو کلمه خاص	Caption
گزینههای منو که بر اساس وظیفه گروهبندی میشوند تا حداقل	0
تعداد سطوح را برای دسترسی به ویژگی مشخص کن.	Organization
کلید دسترسی برای هر گزینه که باید منحصر به فرد باشد و حرف	
اول آن است.	Access Key
کلید میانبر برای هر گزینه	Shortcut Key
کادر علامت برای گزینه هایی که نیاز به فعال یا غیرفعال شدن	
دارند.	Check Box

#### ايجاد منو

در VB دو روش برای ایجاد منو وجود دارد:

منوها

Application Wizard .1

Menu Editor .2

گزینه اول ابزار مناسبی برای ایجاد برنامه کاربردی جدید است. مزیت عمده Application Wizard ایجاد منوهایی است که دارای ویژگیهای استاندارد ویندوزند.

در صورتی که میخواهید ویژگیهای دیگری را به آنها اضافه کنید باید برنامهنویسی کنید، یا از Menu Editor استفاده کنید.

ايجاد منو با ويزارد

File/new project/vb appilication wizard

Appilication wizard امکان سفارشی کردن نوار ابزار، پرونده منبع، مرورگر،

اتصال پایگاه داده را فراهم میکند.

(Menu Editor) كاربرد

امکان ایجاد نوار منو یا اصلاح منوهای ایجاد شده را فراهم میکند و به طور

معمول در بالای فرم قرار میگیرد.

# روشهای بازکردن Menu Editor

1. فشار دادن همزمان کلیدهای Ctrl+E

2. د کمه Menu Editor در نوار ابزار استاندارد

Tools/menu editor

از دستور Load me برای بار گذاری و از دستور Unload me برای پاک کردن آن از حافظه استفاده کنید.

یا:

**تنظیم مشخصه های منو** مانند هر کنترلی میتوان هنگام ایجاد منو مشخصه نیم را مقداردهی کرد. عدم مقداردهی Caption یک خط خالی خواهد شد و عدم مقداردهی name سبب بروز خطا می شود.

مشخصه های متداول شی منو

شرح	مقدار / نوع	مشخصه
متنی که روی نوار منو ظاهر میشود.	String	Caption

77

یک علامت تیک قبل از رشته کشپن گزینه قرار میدهد.	Boolean	Checked
در صورتی که تودرتو باشد رشته کپشن به صورت غیرفعال نخواهد بود.	Boolean	Enabled
نام شی فقط در زمان طراحی قابل دسترس است	String	Name
به عنوان کلید میانبر استفاده می شود.	n/a	Shortcut
یک منوی سطح بالا در فرم MDI ایجاد می کند تا لیستی از پنجرههای باز نمایش دهد.	Boolean	Window List

برای تعریف یک کلید دسترسی در کادر متن کپشن قبل از نویسه مورد نظر از نویسه

&

استفاده کنید.

### اضافه کردن کلید های میانبر به گزینه ها

از طریق روش Application Wizard نمی توان برای گزینه ها کلیدهای میانبر اضافه کرد.ولی انجام این کار در Menu Editor ممکن است. کلید میانبر هر منو را می توان در کادر محاوره ای Menu Editor از لیست Shortcut انتخاب کرد.

**نکته:** هر برنامه فقط می تواند یک نمونه از کلید میانبر را داشته باشد. و هنگام تعیین کلید میانبر تکراری خطای زیر را نشون میدهد.

Shortcut key already assigned

#### ایجاد منوهای باز شو:

دو نوع منو وجود دارد: نوار منويي و نوار بازشو.

نوار منویی به طور ثابت در بالای فرم قرار می گیرد اما منوی بازشو منویی است که در

هر جایی از فرم ظاهر میشود.

Popup Menu توی VB با استفاده از این دستور میتوانین قسمتی از منوها را به همراه زیرمنوهای بعد

از آن در ناحیهای از فرم ظاهر کرده و از آن استفاده نمود.

شکل کلی آن به شکل زیر میباشد:

#### popupmenu menuname[xpos],[ypos]

**آرایه متغیری است با چندین خانه که می توان در هر خانه آن یک مقدار نگهداری کرد.** به هر خانه آرایه یک **عنصر** گفته می شود. عنصر دارای شماره خاصی است که محل عنصر در داخل آرایه را برمی گرداند. اولین عنصر آرایه در موقعیت صفر قرار دارد.

> **اعلان ارایه** الف) مانند یک متغیر

ب) با استفاده از کلید واژه To

## اعلان ارایه مانند یک متغیر

Dim |public|private arrayname (subscript) as datatype

subscript مکان اخرین عنصر ارایه را تعیین میکند.

### **اعلان ارایه با کلید واژه To**

Dim imy array (1 to 5) as integer

دستور بالا آرایهای از 5 متغیر عدد صحیح ایجاد می کند که اولین عنصر در مکان یک

و

آخرین عنصر در مکان پنج میباشد.

**تغییر تعداد عناصر آرایه** هنگامی که تعداد عناصر آرایه را تغییر میدهیم بعد آن آرایه تغییر می کند. برای انجام این کار از کلید واژه ReDim به شکل زیر استفاده می کنیم:

Redim [preserve] arrayname (subscript)as data type یک کلیدواژه اختیاری است که امکان نگهداری مقادیر تمام عناصر از قبل شده در آرایه را فراهم می کند.

Redim کلیدواژهای در VB است که مشخص می کند آرایه تغییر بعد میابد.

## آرایه کنترلی

آرایههای کنترلی یک ویژگی ذاتی در VB هستند که توانایی و کارایی زبان برنامهنویسی را افزایش میدهند.

# اضافه کردن عنصر ها به ارایه کنترلی در زمان اجرا

Load: از این دستور میتوان برای افزودن کنترلها به آرایه کنترلی در زمان اجرا استفاده کرد.

#### آرایه های چند بعدی در ویژوال بیسیک

دوستان در بخش اول این فصل، درمورد آرایههای یک بعدی و دوبعدی بحث شد و روشهای مرتبسازی و جستجو در آرایهها نیز بحث شد. تا به اینجا، برای دسترسی به خانه آرایههای یک بعدی، از یک اندیس استفاده می کردیم. اما آرایههایی هم وجود دارند، که برای دسترسی به عناصر اونها به بیش از یک اندیس نیاز هست، به این آرایهها، آرایههای چندبعدی گفته میشه.آرایههای چند بعدی بصورت زیر تعریف میشن:

نوع آرایه As (و... و بعد 2 و بعد 1) نام آرایه Dim

هر یک از بعدهایی که در قست بالا می بینید، یک حد پایین و یک حد بالا دارن که بصورت زیر تعریف میشن:

حد پايين To حد بالا

دستورات زیر رو در نظر بگیرید. این دستورات، یک آرایه سهبعدی تعریف می کنه، که حد پایین بعد دوم 2 و می کنه، که حد پایین بعد اول اون 1 و حد بالای اون 4 هست. حد پایین بعد دوم 2 و حد بالای اون 5 هست، و حد پایین بعد سوم اون 1 و حد بالای اون 5 هست.

Dim a(1 To 4, 1 To 30, 1 To 5)

**نکته:** مثل تعریف آرایه های یک بعدی و دو بعدی، در تعریف این آرایه ها می تونید از کلمات کلیدی Dim یا Public استفاده کنید.

### آرایههای دوبعدی

آرایههای دوبعدی، در بخش اول توضیح داده شد. این نوع آرایهها رو می تونید جدول یا **ماتریس** بنامید. در هر آرایه دوبعدی بعد اول، سطر و بعد دوم، ستون هست. برای مثال دستور زیر، یک آرایه دوبعدی رو با 3 سطر و 4 ستون تعریف می کنه:

Dim a(1 To 3, 1 To 4) As Integer

## پردازش آرایههای دوبعدی

آرایههای دوبعدی متداول ترین نوع آرایهها هستند و به سه روش می تونید اونها رو پردازش کنید، روی دادههای اونها عملیاتی رو انجام بدین:

**تصادفی:** در این روش کاربر می تونه، به هر عنصر آرایه دسترسی داشته باشه. یعنی می تونه تو هر عنصر مقداری رو قرار بده یا از مقدارهای عناصر مختلف استفاده کنه.

**سطری:**در این روش، از اول سطر اول شروع می کنه و تمامی عناصر رو پردازش می کنه و سپس میره به سطر بعدی و سپس به سطر دوم، و به همین روش ادامه میده.

**ستونی :**در این روش که برعکس روش قبلی، یعنی سطری هست، ابتدا عناصر موجود در ستون اول و سپس ستون دوم به ترتیب پردازش میشن.

For برای همین، برای پردازش سطری و ستونی یک آرایه دوبعدی به 2تا حلقه For نیاز دارید. در پردازش سطری، شمارنده حلقه بیرونی، اندیس سطر و شمارنده حلقه درونی اندیس ستون هست، و در پردازش ستونی شماره حلقه بیرونی اندیس ستون و شمارنده حلقه درونی اندیس سطر هستش. در نظر بگیرید یک آرایه دارای M سطر و N ستون هست، برای پردازش سطری این آرایه، از حلقههای زیر استفاده میشه. For I = 1 To M

For j = 1 To N

A(I,j)

Next j

Next i

برای پردازش ستونی این آرایه از حلقههای زیر استفاده میشه:

For I = 1 To N

For j =1 To M

A(I,j)

Next j

Next i

دوستان امیدورام مطالب این بخش رو تا به اینجای کار بخوبی یاد گرفته باشید. بحث

بعدی که شروع می کنیم درمورد آرایهای از عناصر ویژوالبیسیک هست.

آرایهای از عناصرهای ویژوال بیسیک

برای ایجاد آرایهای از عنصرها در ویژوال بیسیک کافیه چند عنصر همنام به فرم اضافه کنید. به عنوان مثال، پنج Command Button به فرم اضافه کنید و اسمهای هر کدوم رو یک اسم مثلا Command1 در نظر بگیرید.به این ترتیب عنصرها یک آرایه رو تشکیل میدن. یا یک عنصر رو روی فرم منتقل کنید و از اون کپی بگیرید و روی فرم Paste کنید. هنگام Paste کردن پیامی ظاهر خواهد شد، در این کادر کلید Yes کلیک کنید تا عناصر با هم تشکیل یک آرایه رو بدن. هر عنصری که به این روش ایجاد کنید دارای یک Index خواهد بود. برای دستیابی به عناصر این آرایه از این می استفاده کنید. در واقع Index مشخص می کند که کدام کلید فشرده شده است.



**برنامه نویسی شیءگرائیobject oreinted programming** وسیلهای برای مدل کردن صحیح دنیای واقعی با استفاده از اشیا در برنامه و استفاده مجدد از کد است. یک شی در برنامه دقیقا همان طور تعریف می شود که در دنیای واقعی است؛ خواص معینی دارد که آن را توصیف می کند و متدهایی که می توانید برای انجام کار معینی روی شیء استفاده کنید.

شرحی از داده ها و توابعی که میتوانند روی داده کار کنند را کلاس class مینامند. کلاس را به عنوان الگوئی برای تولید شیء میتوان تصور کرد. کلاس درواقع یک نوع داده user-defined است . اشیاء نمونه هائی از کلاس ها هستند که در زمان اجرا ایجاد میشوند.

چهار مفهوم اصلی وجود دارند که اساس برنامهنویسی شیء گرائی را میسازند و توسط کلاس ها ارائه می شوند. این مفاهیم " انتزاع (abstraction)، کپسوله کردن (encapsulation)، وارث (inheritance) و چندریختی (polymorphism)

چند کلاس ممکن است خصوصیات و رفتارهای مشترکی داشته باشند اما هر یک شامل خواص و توابع دیگری هم باشد. وراثت اجازه میدهد یک کلاس عمومی تعریف شود که اشیا در خصوصیات آن مشترک هستند و این کلاس می تواند توسط سایر کلاس ها ارث برده شود و خواص جدیدی به آن اضافه شود بدون اینکه تأثیری روی کلاس عمومی داشته باشد.

توارث شباهت بین دو کلاس را با استفاده از مفاهیم کلاس پایه (base) و کلاس مشتق شده (derived) بیان می کند. کلاسی که از آن ارثبری می شود کلاس پایه یا مبنا و کلاس وارث که خصوصیات کلاس پایه را به ارث می برد را کلاس مشتق شده می نامند. کلاس پایه شامل کلیه خواص و رفتارهائی است که بین کلاس های مشتق شده مشتر ک است.

مثال: کلاس پایه shape را درنظر بگیرید که دارای خاصیتهای اندازه، رنگ و موقعیت است. هر شکل می تواند رسم شود، پاک شود، حرکت کند و رنگ شود. هر کدام از اشکال دارای خواص و رفتارهای اضافه تری هستند. برای یک شکل معین بعضی رفتارها ممکن است متفاوت باشد مثلا محاسبه مساحت 1. یک کلاس مشتق شده به نوبه خود می تواند کلاس پایه برای سایر کلاس ها

باشد.

 2. اگر کلاس پایه تغییر کند کلاس مشتق شده نیز تحت تأثیر این تغییرات قرار می گیرد.

فرم کلی تعریف یک کلاس مشتق شده به صورت زیر است:

class derived : access base

{
//members of new class;

}

Derived نام کلاس جدید است که از کلاس پایه base مشتق شده است.

قسمت access اختیاری است ولی می تواند private ، public یا protected

باشد.

کلاس مشتق شده کلیه اعضای کلاس پایه را به ارث میبرند اما اجازه دسترسی مستقیم به اعضای خصوصی کلاس پایه را ندارند و تنها از طریق توابع عمومی و سازنده به آنها دسترسی دارند. نحوه دسترسی به اعضای عمومی کلاس پایه در کلاس مشتق شده توسط یکی مجوزهای دسترسی زیر که قبل از نام کلاس پایه ذکر می شود مشخص می شود: •public

• private

protected

مثال زیر یک کلاس پایه، حقوق و دستمزد، و یک کلاس مشتق شده است، BonusPayroll، که لغو یک متد، PayEmployee تعریف می کند. روش، RunPayroll، ایجاد و سپس عبور یک جسم حقوق و دستمزد و یک شی BonusPayroll به یک تابع، پرداخت، که روش PayEmployee از هر دو اشیاء اجرا می کند.

Const BonusRate As Decimal = 1.45D

Const PayRate As Decimal = 14.75D

**Class Payroll** 

Overridable Function PayEmployee(

ByVal HoursWorked As Decimal,

ByVal PayRate As Decimal)

As Decimal

PayEmployee = HoursWorked \* PayRate

**End Function** 

End Class

Class BonusPayroll Inherits Payroll

**Overrides Function PayEmployee(** 

ByVal HoursWorked As Decimal,

ByVal PayRate As Decimal) As Decimal

PayEmployee = MyBase.PayEmployee(HoursWorked,

PayRate) \* BonusRate End Function

**End Class** 

Sub RunPayroll()

Dim PayrollItem As Payroll = New Payroll

Dim BonusPayrollItem As New BonusPayroll

Dim HoursWorked As Decimal = 40

MsgBox("Normal pay is: " &

PayrollItem.PayEmployee(HoursWorked, PayRate))

MsgBox("Pay with bonus is: " &

BonusPayrollItem.PayEmployee(HoursWorked, PayRate)) End Sub

### توارث عمومي

با ذکر کلمه public قبل از نام کلاس پایه اعضای عمومی کلاس پایه به عنوان اعضای عمومی کلاس مشتق شده تلقی میشوند و در اختیار کاربر کلاس مشتق شده قرار می گیرد.

## توارث خصوصي

با حذف کلمه public یا صریحا با ذکر کلمه private یک کلاس پایه می تواند به صورت خصوصی ارث گرفته شود. در توارث خصوصی کلاس مشتق شده کلیه اعضای کلاس پایه را دارا خواهد بود اما به صورت مخفی و اعضای عمومی کلاس پایه اعضای خصوصی کلاس مشتق شده خواهند شد. بنابراین یک شی به عنوان یک نمونه از کلاس نمی تواند به اعضای کلاس پایه دسترسی پیدا کند. 1.توارث خصوصي براي پنهان كردن لايه زيرين

پیادهسازی کلاس پایه مفید است.

2.در توارث خصوصی کلیه اعضای عمومی کلاس پایه خصوصی میشوند. اگر میخواهید عضوی قابل رؤیت شود کافی است نام آن را (بدون آرگومان و مقدار برگشتی) در بخش public کلاس مشتق شده ذکر کنید.

اعضای خصوصی همیشه خصوصی هستند اما گاهی میخواهید اعضائی را از خارج مخفی کنید ولی در کلاس مشتق شده قابل رؤیت باشند. کلمه protected می گوید که اعضای محافظت شده برای هر کسی که از این کلاس ارث میبرد قابل دسترس است و برای بقیه خصوصی است.

مثال: توابع set و read از کلاس Base در مثال قبل در کلاس مشتق شده Derived قابل رؤیت هستند ولی در برنامه مخفی هستند.

با قرار دادن کلمه protected قبل از نام کلاس مشتق شده اعضای محافظت شده و عمومی کلاس پایه به اعضای محافظت شده کلاس مشتق شده اضافه

خواهند شد. بنابراین برای وارثین کلاس مشتق شده در دسترس است و برای بقیه ينهان باقي ميماند. 1. در کلیه حالات اعضای خصوصی کلاس پایه در وراثت شرکت نمیکنند و خصوصي باقي مي مانند. 2. معمولا توارث عمومي است تا رابط كلاس پايه همچنان رابط كلاس مشتق شده باشد. **نکته:** توارث محافظت شده خیلی استفاده نمی شود و فقط برای تکمیل زبان برنامەنويسى است. 3. مناسب ترین روش این است که اعضای دادهای کلاس را صورت خصوصی تعریف کنید تا امکان تغییر پیادهسازی زیرین حفظ شود. و به وارثین کلاس مجوز دسترسی کنترل شدهای به توابع عضو محافظت شده بدهید. کلاس مشتق شده می تواند از بیش از یک کلاس پایه ارثبری داشته باشد. در

توارث چندگانه (multiple inheritance) چند کلاس به عنوان کلاس پایه

اضافه می شوند. نام کلاس های پایه توسط کاما از هم جدا می شود. توارث چند گانه

مسائلی را پیش می کشد که در مبحث چندریختی توضیح داده خواهد شد.

class derived : access base1, access base2 ,.. { //...



### مفهوم پایداری

برای اینکه برنامهای بتواند اطلاعات را در خود نگه دارد باید توانایی ذخیرهسازی روی دیسک سخت را نیز داشته باشد.

پروندهها برای ذخیره و بازیابی دادهها: دادهها در حافظه اصلی و پروندهها روی حافظه جانبی ذخیره میشود، و هرگز برنامهها به طور مستقیم با یک پرونده روی دیسک کار نمی کنند .

**انواع پروندهها** ترتیبی(Sequential): در فایلهای ترتیبی اطلاعات به همان ترتیبی که در فایل ذخیره می شود، قابل بازیابی هستند.

تصادفی (Random): در فایلهای تصادفی نوشتن و بازیابی اطلاعات از هر نقطه فایل امکانپذیر . . . . . . . .

**دودویی(Binary):** پروندههای دودویی نوع خاصی از پروندههای تصادفی میباشد.

تخصیص شماره پرونده: می توان به طور همزمان چندین پروندهی باز داشته باشیم به شرط اینکه هر پرونده شمارهی خاص خود را داشته باشد. تابع Free File ، اولین شماره پرونده آزاد را بر می گرداند. استفاده از آن به صورت زیر است:

Dim fh As Integer fh = Free File دستور Open: برای باز کردن یک پرونده استفاده می شود. استفاده از آن به صورت زیر است:

Dim fh As Integer fh = Free File open "مىير پرونده"for حالت دسترسى As #fh Len = طول ركورد حالتهاى دسترسى به يرونده عبار تند از:

Append: دادهها را به انتهای پرونده اضافه میکند. (درصورتی که پرونده نباشد آنرا ایجاد میکند)

Input: پرونده ترتيبي را براي خواندن باز مي كند.

Output: پرونده ترتيبي را براي نوشتن باز مي كند.

Random: پرونده را برای دسترسی تصادفی باز می کند.

Binary: پرونده را به صورت دودویی باز میکند.

**دستور Close:** برای بستن پرونده از این دستور استفاده کنید.

Close #fh

**نکته:** اگر از دستور **close** بدون پارامتر استفاده شود؛ همه پروندههای باز را می بندد.

Close

**پرونده ترتیبی:** برای استفاده از پرونده ترتیبی باید تمام پرونده را خواند. این پروندهها برای ذخیره کردن پروندههای متنی کوچک که در آنها سرعت چندان مهم نیست، مناسب هستند.

نوشتن پرونده های ترتیبی: برای نوشتن در پرونده های ترتیبی آن را به صورت Output یا Append باز کنید و از دستورهای Print# یا Write# برای نوشتن استفاده کنید.

دستور #Print: با این دستور فقط در پرونده های ترتیبی می توان نوشت.

مقادیری که می خواهی در فایل ذخیر ه شود Print #fh, مقادیری

دستور #Write: با این دستور فقط در پرونده های ترتیبی می توان نوشت. داده ها با #Write به وسیله کاما از هم جدا خواهند شد.

مقادیری که می خواهی در فایل ذخیره شود , Write #fh خواندن پروندههای ترتیبی: باید پرونده را در حالت دسترسی Input باز کنید و از دستورات #Input یا #Line Input برای خواندن پروندههای ترتیبی استفاده کنید.

بازیابی داده ها با دستور Input: با استفاده از این دستور می توانید داده های یک پرونده ترتیبی را بخوانید. خواندن یک پرونده ترتیبی باید دقیقا به همان ترتیب نوشتن آن صورت بگیرد. Input #fh, 1 متغیر 2, متغیر 2, متغیر 1, ۲۰۰۰ نوع داده متغیرهای دستور Input، باید با دستور Print که این مقادیر را در پرونده نوشته است، مطابقت داشته باشد. از آنجایی که داده ها با #Write به وسیله کاما از هم جدا خواهند شد، انطباق #Write و #Input ضروری نیست.

**بازیابی داده ها با دستورLine Input:** با استفاده از این دستور می توانید دادههای این پرونده را خط به خط بخوانید.

شکل کلی دستور Line Input به صورت زیر است:

Dim st As String line Input #fh, st پرونده ی تصادفی: پرونده ای است که در هر نقطه از آن (بدون رعایت ترتیب) می توان در آن نوشت یا از آن خواند. دلیل اولیه اهمیت پرونده های تصادفی معرفی نوع جدیدی از انواع داده است.

UDT: یکی از ویژگیهای ویژوال بیسیک فراهم نمودن امکان ایجاد نوع دادهی سفارشی است که به نام نوع دادهی تعریف شده به وسیله کاربر (user-defined data type) یا UDT معروف است.

**ر کورد:** با ترکیب دادههای ذاتی visual basic می توانید انواع دادههای جدیدی ایجاد کنید. این بسته را نوع دادهی تعریف شده به وسیلهی کاربر می نامند.که به آن گاهی ساختار (structure) یا رکورد نیز گفته می شود. برای تعریف نوع داده ی کاربراز دستور type در ماژول استفاده می شود. تعریف udt در داخل روال مجاز نمی باشد.شکل کلی دستور به صورت زیر است:

نام دلخواه Public Type نوع داده <sub>as</sub> نام عنصر

....

End type

شکل کلی رجوع به عناصر نوع داده تعریف شده توسط کاربر به صورت زیر است:

نام عنصر خاصي از نوع داده . نام متغير

باز کردن پروندههای تصادفی:

Open "random.txt" For random as # 1 Len =Len (نام رکورد)

**دستور های Get و Put :** برای خواندن و نوشتن پرونده های تصادفی از دو دستور #Get و #Put استفاده می کنیم.

**نکته:** در صورتی که شمارهی رکورد در این دستورها ذکر نشود عملیات روی رکورد بعد از رکورد جاری انجام می شود.

Variable: متغیری است که مقدار آن در پرونده نوشته می شود و یا مقداری که از پرونده خوانده شده در آن ذخیره می شود.