

فرادرس

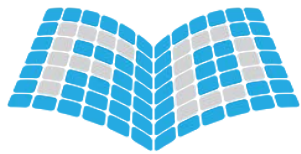
فرا ترا از یک کلاس درس
www.faradars.org

فرادرس آموزش کاربردی C#.Net

مدرس: مهندس رشید شجاعی

مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

faradars.org/fvcs9311



فرادرس

فرا تراژیک کلاس درس
www.faradars.org

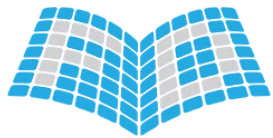
Net چیست؟

Net محیط طراحی شده توسط شرکت مایکروسافت جهت ساخت برنامه‌های توزیع شده است که شامل ابزارهای گوناگونی نظیر

- "کتابخانه کلاس‌های پایه" (BCL)،
- "زبان عمومی زمان اجرا" (CLR)،
- تکنولوژی ایجاد برنامه‌های تحت وب ASP.NET،
- استفاده از کلاس‌های ADO.NET،
- زبان‌های برنامه‌نویسی و... است.

این ابزارها در کنار یکدیگر امکان ساخت انواع مختلفی از نرم‌افزارها از قبیل برنامه‌های تحت ویندوز (Windows Applications)، برنامه‌های تحت وب (Web Applications) و سرویس‌های

وب (Web Services) را فراهم می‌آورند. مجموعه آموزش‌های کاربردی برنامه نویسی C# (سی شارپ)



فرادرس

فرا ترا از یک کلاس درس
www.faradars.org

Net Framework چیست؟

یا همان چارچوب کاری NET، مجموعه‌ای از کتابخانه‌های بسیار غنی جهت کمک به سهولت توسعه نرم‌افزار هستند که در قالب بخش‌های عمده‌ای همچون تکنولوژی ASP.NET، ADO.NET و بسیاری از فناوری‌های دیگر ارائه می‌شوند که تعداد آنها در نسخه‌های اخیر همگام با محدود تر شدن اهداف مایکروسافت بیشتر شده است.

آشنایی با زبان C#

✓ C# یکی از زبان‌های سطح بالا و شی‌گراست که با ارائه رهیافت Component-Based راهی برای طراحی و توسعه نرم‌افزار می‌باشد.

✓ زبان برنامه‌سازی C# به همراه تکنولوژی جدید شرکت مایکروسافت یعنی Net، ارائه گردید و از این رو از تکنولوژی Net، این شرکت بهره می‌برد.

✓ هدف آن ترکیب قدرت محاسباتی زبان C++ و سادگی برنامه‌نویسی با زبان Visual Basic است.

✓ C# برپایه زبان برنامه‌نویسی C++ ایجاد شده است و از برخی از ویژگی‌های C# (سی شارپ) نیز استفاده می‌کند.

آشنایی با نرم افزار Visual Studio

✓ Microsoft Visual Studio محصول شرکت مایکروسافت و محیطی یکپارچه جهت تولید و توسعه نرم افزار (IDE) است که به منظور ایجاد برنامه های تحت ویندوز و تحت وب، طراحی سایت ها و ساخت سرویس های وب استفاده می شود.

✓ این نرم افزار امکان استفاده از ویژگی های مختلف .NET. که قبلاً گفته شد را به کاربر میدهد، از قبیل:

- استفاده از زبان های برنامه نویسی مختلف مانند C#، J#، ++C و

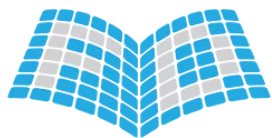
VB.NET

- ایجاد Windows/Web/Console Applications ها
- استفاده از سرویس ها و ویژگی های .NET Framework. که قبلاً گفته شد.

مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

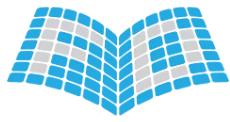
• و...

faradars.org/fvcs9311



فرادرس

فرا تر از یک کلاس درس
www.faradars.org



فرادرس

فراتر از یک کلاس درس
www.faradars.org

روند پیشرفت Visual Studio و .NET Framework

✓ در حقیقت هدف اصلی و دراز مدت مایکروسافت متحد کردن ابزارهای برنامه‌نویسی تحت یک محیط واحد با در دسترس بودن ابزار و امکانات برنامه‌نویسی آسانتر و کارتر برای برنامه‌نویس بوده است.

✓ ارائه نسخه اول ویژوال استودیو به نام Visual Studio 97 در سال ۱۹۹۷

✓ ارائه Visual Studio 98 در سال ۱۹۹۸ با نام Visual Studio 6 (شامل Visual Basic 6، Visual C/C++ 6 و..)

✓ ارائه Visual Studio.NET 2001 همراه با .NET Framework 1.0

✓ ارائه Visual Studio.NET 2003 همراه با .NET Framework 1.1

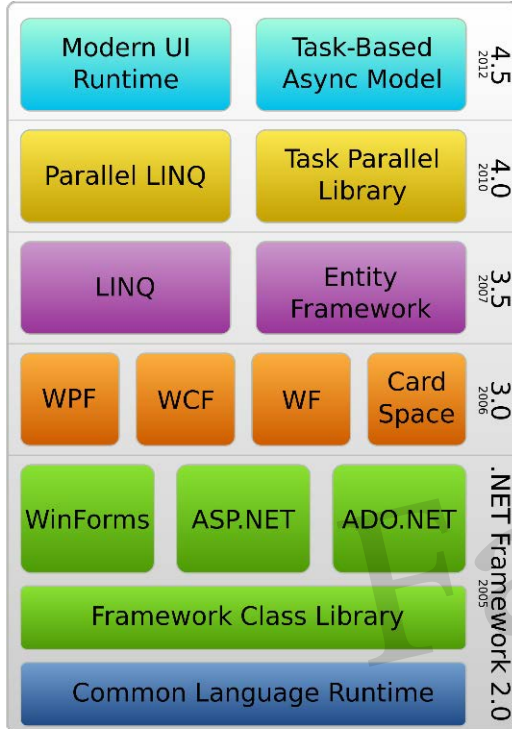
✓ ارائه Visual Studio.NET 2005 همراه با .NET Framework 2.0

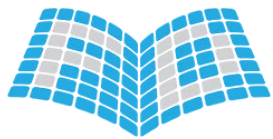
✓ ارائه Visual Studio.NET 2008 همراه با .NET Framework 3.0

✓ ارائه Visual Studio.NET 2010 همراه با .NET Framework 4.0

✓ ارائه Visual Studio.NET 2012 همراه با .NET Framework 4.5

مجموعه آموزش های کاربردی برنامه نویسی C# (لمی شارپ)





فرادرس

فرا تراز یک کلاس درس
www.faradars.org

Visual Studio 2015

شرکت مایکروسافت در آخرین مراسم خود در نیویورک از امکانات عالی Visual Studio 2015 خبر داد و این خبر طیف وسیعی از برنامه نویسان را خوشحال نمود، حتی برنامه نویسانی که با زبان های غیر مایکروسافتی برنامه می نوشتند! اولین خبر خوشحال کننده این است که مایکروسافت ویژوال استودیو ۲۰۱۵ بصورت رایگان در اختیار کاربران قرار می گیرد.

یکی از مهمترین ویژگی های ویژوال استودیو ۲۰۱۵ این است که توسعه دهندگان می توانند برای ساخت برنامه های iOS، Linux و حتی Android در محیط ویندوز (در ویژوال استودیو ۲۰۱۵) به راحتی و بدون مشکل به برنامه نویسی بپردازند.

به نظر میرسد در آینده ای چندان دور شاهد آن خواهیم بود که بسیاری از برنامه نویسان Android، iOS، Mac، Linux و ... نیز در ویژوال استودیو مایکروسافت، و با زبان های مایکروسافتی برنامه هایشان را بنویسند.

مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

faradars.org/fvcs9311

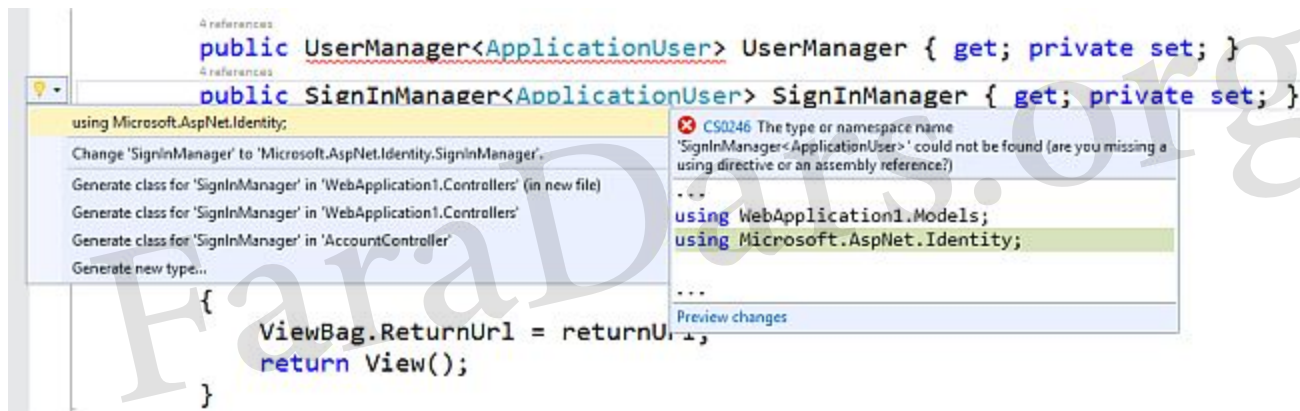


فرادرس

فرا تراژیک کلاس درس
www.faradars.org

Roslyn, .NET compiler platform is integrated in Visual Studio 2015

New feature includes better code fixing as developers write code in editor. Visual Studio 2015 has support for breakpoint configuration and debugging lambda expression. Smart Unit Tests feature is also added to Visual Studio 2015.



مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

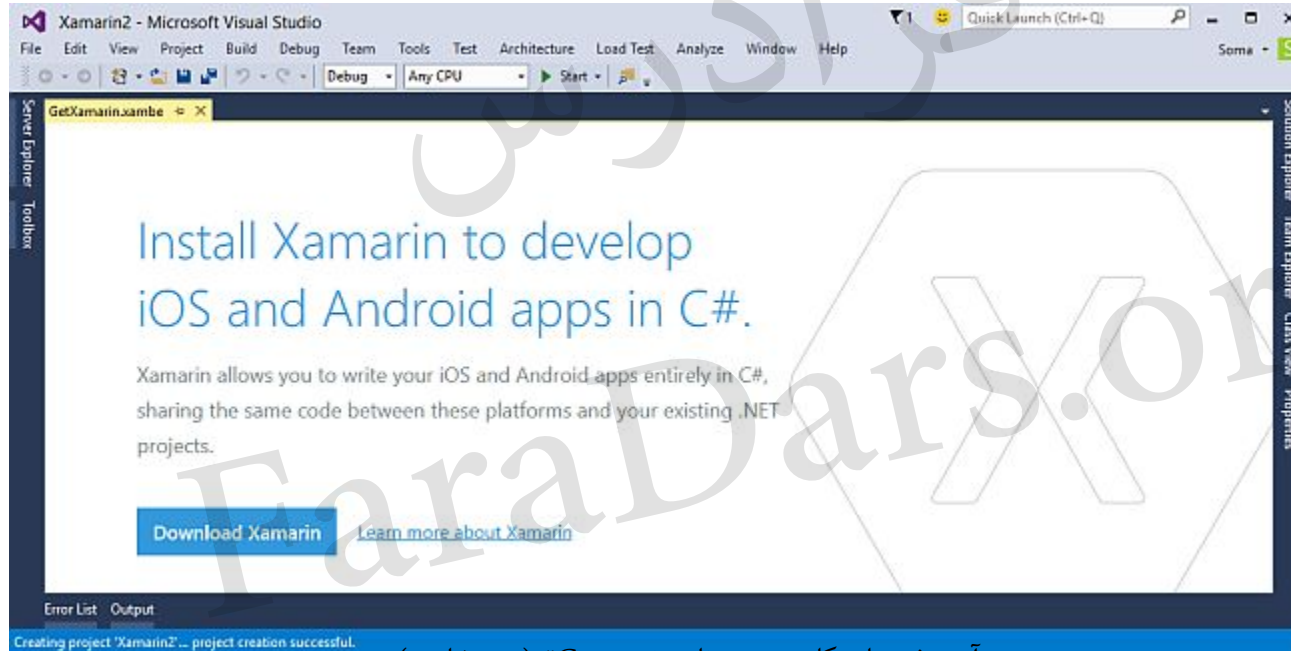
faradars.org/fvcs9311



فرادرس

فراتر از یک کلاس درس
www.faradars.org

Developers can install Xamarin from Visual Studio and vice versa Visual Studio support will be available from Xamarin.



مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

faradars.org/fvcs9311



فرادرس

فرا تراژیک کلاس درس
www.faradars.org

Microsoft Visual Studio 2015 now has options for Android development: C++, Cordova, and C# with Xamarin. When choosing one of those Android development options, Visual Studio will also install the brand new Visual Studio Emulator for Android to use as a target for debugging your app.



مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

faradars.org/fvcs9311



فرادرس

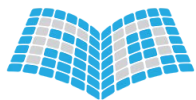
فراتر از یک کلاس درس
www.faradars.org

we will be creating apps for potentially 6 platforms in one go (and 8 platforms if we include Android and iOS, assuming Xamarin turns out the way I'm hoping).



مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

faradars.org/fvcs9311



فرادرس

فراتر از یک کلاس درس
www.faradars.org

Surface Hub

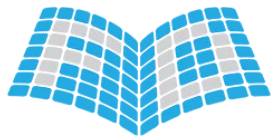
محصول جدیدی دیگر از مایکروسافت که در تاریخ ۲۱ ژانویه ۲۰۱۵ رونمایی شد. Surface Hub در واقع یک تبلت بزرگ است که توانایی های بسیاری در زمینه انجام کارهای گروهی دارد. از جمله توانایی های Surface Hub می توان به:



- پشتیبانی از انواع روش اتصال از جمله wireless projection, HDMI, NFC, Bluetooth
- مانیتور در دو سایز ۵۵ و ۸۴ اینچ لمسی با کیفیت عالی 4k
- دو قلم نوری بسیار عالی با قابلیت شارژ مجدد
- وجود میکروفون و اسپیکرهای بسیار با کیفیت
- بروی دستگاه
- دارای آخرین نسخه آفیس با قابلیت لمسی
- ویندوز ۱۰
- پشتیبانی از کلیه برنامه های Universal

مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

faradars.org/fvcs9311



فرادرس

فرا ترا از یک کلاس درس
www.faradars.org

HoloLens



مایکروسافت در کنار ویندوز ۱۰ خود از عینک هولولنز (HoloLens) خود پرده برداشت. هولولنز هدستی است که دنیای واقعی را با دنیای مجازی شما یکی خواهد کرد. یعنی شما می توانید اشکال خود را در دنیای واقعی با این عینک بسازید و با آن زندگی کنید. برای مثال شما با هدست هولولنز (HoloLens) می توانید به راحتی یک ویدئو را بر روی دیوار اتاقتان بدون هیچ تلویزیونی تماشا کنید و اینها همه از ترکیب دنیای مجازی هولولنز

(HoloLens) با دنیای واقعی خودمان اتفاق می افتد. آموزش های کاربردی برنامه نویسی C# (سی شارپ)

faradars.org/fvcs9311



مثال: برنامه‌ای بنویسید که نام و مقدار حقوق یک کارمند را دریافت کرده، مقدار مالیات و مقدار حقوق خالص وی را محاسبه کند و روی فرم نشان دهد. فرض کنید فرمول محاسبه مالیات به شکل زیر است:

مالیات	حقوق بر حسب تومان
0% ←	Salary<500000
5% ←	500000<=salary<1000000
10% ←	1000000<=salary<2000000
15% ←	2000000<=salary<3000000
20% ←	Salary>=3000000

ظاهر برنامه به شکل زیر باشد.

Form1

Enter your name:

Enter your salary:

Result



فرادرس

فرا تار از یک کلاس درس
www.faradars.org

```
if (شرط){  
    مجموعه دستورات ۱  
}  
  
if(شرط){  
    مجموعه دستورات ۱  
}  
else{  
    مجموعه دستورات ۲  
}  
  
if(شرط ۱){  
    مجموعه دستورات ۱  
}  
else if(شرط ۲){  
    مجموعه دستورات ۲  
}  
else if(شرط ۳){  
    مجموعه دستورات ۳  
}  
....  
else{  
    مجموعه دستورات ۴  
}
```

switch (متغیر یا عبارت) {

case مقدار ۱ :

مجموعه دستورات ۱ ;

break;

case مقدار ۲ :

مجموعه دستورات ۲ ;

break;

case مقدار ۳ :

مجموعه دستورات ۳ ;

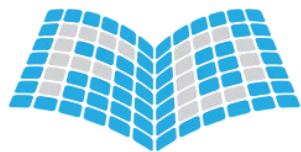
break;

default:

مجموعه دستورات ;

break;

}

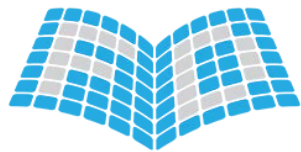


فرادرس

فرا تراژیک کلاس درس
www.faradars.org

تمرین: برنامه ماشین حساب را به نحوی تغییر دهید که دکمه Enter نیز بجای کلید = کار کنند.

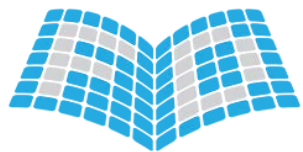
هدف اصلی در این تمرین درک تفاوت رخداد Click با رخداد MouseClick می باشد



فرادرس

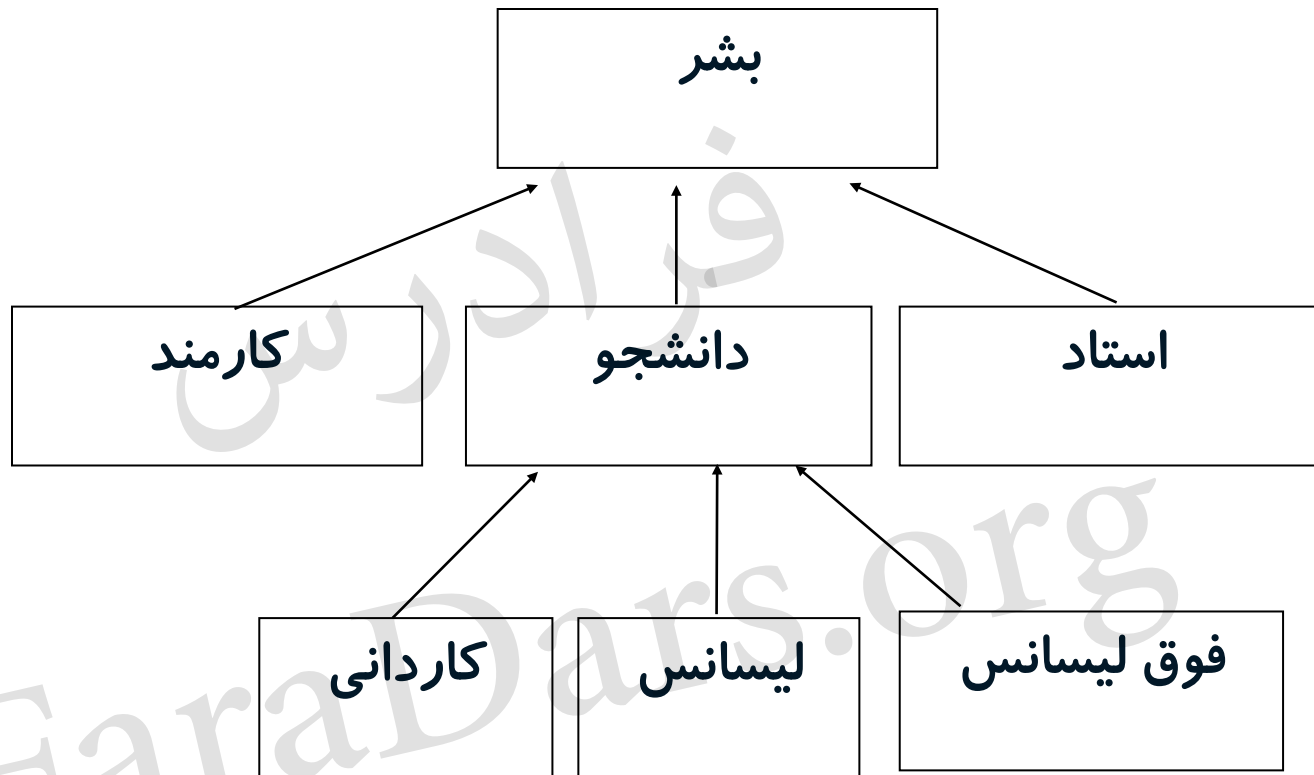
فراتر از یک کلاس درس
www.faradars.org

تمرین: برنامه ماشین حساب را به نحوی تغییر دهید که دکمه Btnonoff نیز کارکنند. در شروع اجرا باید متن این دکمه On باشد به این معنی که با کلیک کردن می توانید ماشین حساب را روشن کنید، در این حالت تمام دکمه های ماشین حساب خاموش باشند. با کلیک بر این دکمه ماشین حساب روشن شده و تمام دکمه ها فعال باشند. در این حالت متن دکمه به Off تبدیل گردد به این معنی که با کلیک کردن می توانید ماشین حساب را خاموش کنید. با هر بار کلیک کردن روی این دکمه ماشین حساب بین دو حالت مذکور تغییر وضعیت دهد.



فرادرس

فرا تراژیک کلاس درس
www.faradars.org



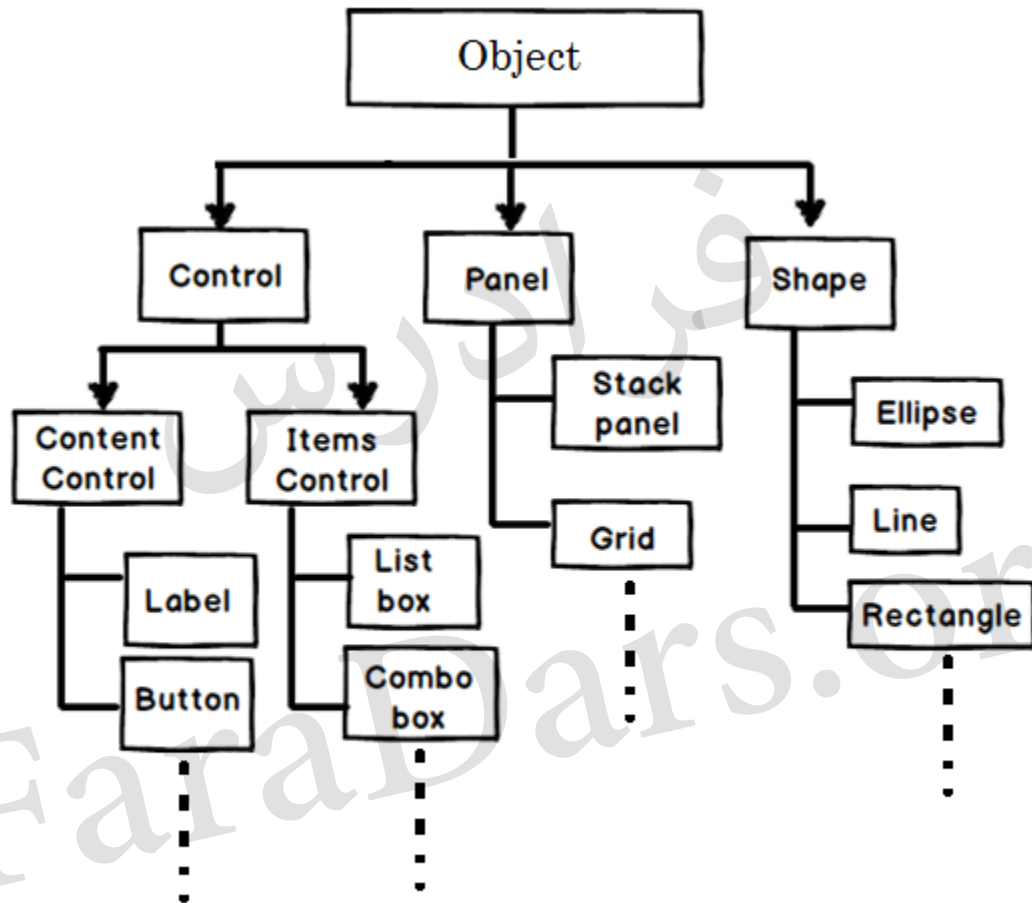
مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

faradars.org/fvcs9311



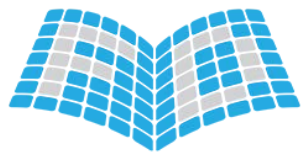
فرادرس

فرا تراز یک کلاس درس
www.faradars.org



مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

faradars.org/fvcs9311



فرادرس

فراتر از یک کلاس درس
www.faradars.org

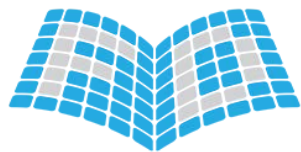
طراحی و پیاده‌سازی برنامه کاربردی Notepad

اهداف آموزشی:

- معرفی منوها ، ایجاد منو و برنامه نویسی آن
- معرفی کادرهای محاوره‌ای متداول و استفاده از آن
- آرایه مفهوم فایل و روش استفاده از آن
- آرائه مفهوم Private و Public و ایجاد اجزا خصوصی و عمومی برای کلاس ها
- آرایه مفهوم سازنده کلاس ها
- ایجاد برنامه های چند فرمی و نحوه دسترسی به اشیاء و کنترل های یک فرم از سایر فرم ها (سازنده و ارسال پارامتر توسط سازنده)

مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

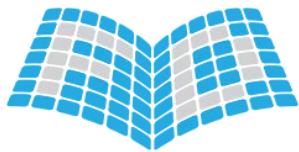
faradars.org/fvcs9311



فرادرس

فراتر از یک کلاس درس
www.faradars.org

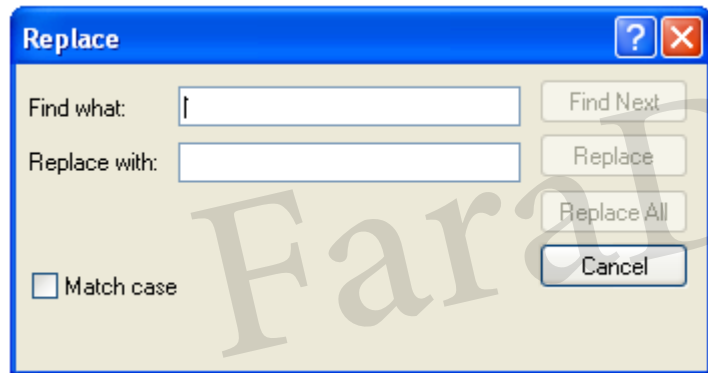
تمرین: برنامه Notepad را به نحوی تغییر دهید که بجز رنگ پس زمینه، فونت جاری و ابعاد فرم (طول و عرض) را دائمی کند (تنها از یک فایل استفاده کنید و به کمک آرایه لیستی از داده‌ها را در فایل ذخیره و بازیابی کنید).



فرادرس

فراتر از یک کلاس درس
www.faradars.org

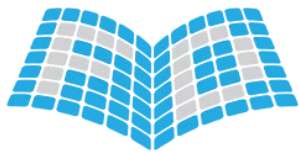
تمرین: یک فرم به نام FrmReplace به برنامه فوق اضافه کنید و ظاهر آنها را همانند شکل زیر طراحی کنید. سپس رخدادهای مورد نیاز را به نحوی برنامه‌نویسی کنید که کارکرد تمام اجزاء آن همانند فرم Replace در برنامه Notepad ویندوز گردد.



در صورتیکه دقت کنید مشخص می شود که تمام بخش های فرم FrmFind در این فرم نیز موجود می باشد و لازم نیست این بخش ها مجددا طراحی شوند و کافیهست فرم FrmReplace فرم FrmFind را به ارث ببرد.

مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

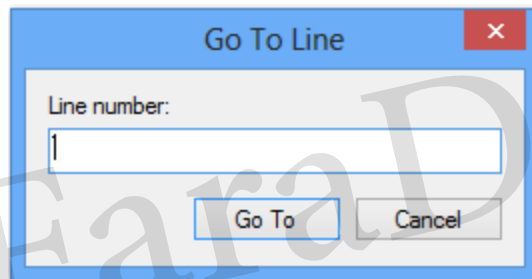
faradars.org/fvcs9311



فرادرس

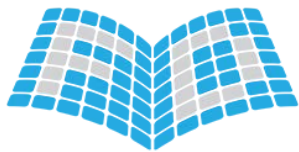
فرا تراژیک کلاس درس
www.faradars.org

تمرین: یک فرم به نام FrmGoto به برنامه فوق اضافه کنید و ظاهر آنها را همانند شکل زیر طراحی کنید. سپس رخدادهای مورد نیاز را به نحوی برنامه نویسی کنید که کارکرد تمام اجزاء آن همانند فرم Goto در برنامه Notepad ویندوز گردد.



مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

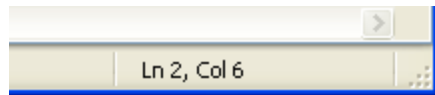
faradars.org/fvcs9311



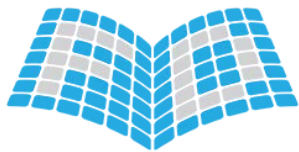
فرادرس

فراتر از یک کلاس درس
www.faradars.org

تمرین: برنامه Mynotepad را به نحوی تغییر دهید که در هر لحظه سطر و ستون محل جاری مکان نما همانند شکل زیر در میله وضعیت نشان داده شود.



دقت کنید که تغییر مقادیر سطر و ستون جاری در میله وضعیت در همه حالتها درست کار کند. (با تایپ کردن، با کلیک کردن موس، با کلیدهای جهت نماو....)



فرادرس

فرا تراژیک کلاس درس
www.faradars.org

تمرین: برنامه Mynotepad را به نحوی تغییر دهید که در هر لحظه وضعیت Enabled گزینه های منو صحیح تنظیم شده باشد. به عنوان مثال زمانی که متنی انتخاب نشده باشد گزینه Copy غیر فعال و در غیرانصورت فعال باشد.

دقت کنید که زمانی که با استفاده از موس یا کیبرد با منو کار می کنید باید گزینه های منو صحیح تنظیم شده باشند.



Firstname	lastname	Phone
Ali	Ahmadi	123
Rashid	Shojaei	12345
babak	amni	21232
alireza	amrani	4342
Nafas	babash	9898

طراحی و پیاده‌سازی یک دفترچه تلفن

اهداف آموزشی:

- آشنایی با روش ایجاد بانک اطلاعاتی به کمک SQL Server
 - آشنایی با تکنولوژی ADO.NET
 - آشنایی با کلاس‌های موجود در ADO.NET
 - معرفی نحوه ارتباط برقرار کردن C#.Net با بانک اطلاعاتی به روش ADO.NET
 - آشنایی با نحوه ذخیره کردن، ویرایش کردن و حذف کردن داده‌ها
 - ایجاد کردن یک جستجوگر پویا
- مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)



فرادرس

فراتر از یک کلاس درس
www.faradars.org

بانک اطلاعاتی چیست؟

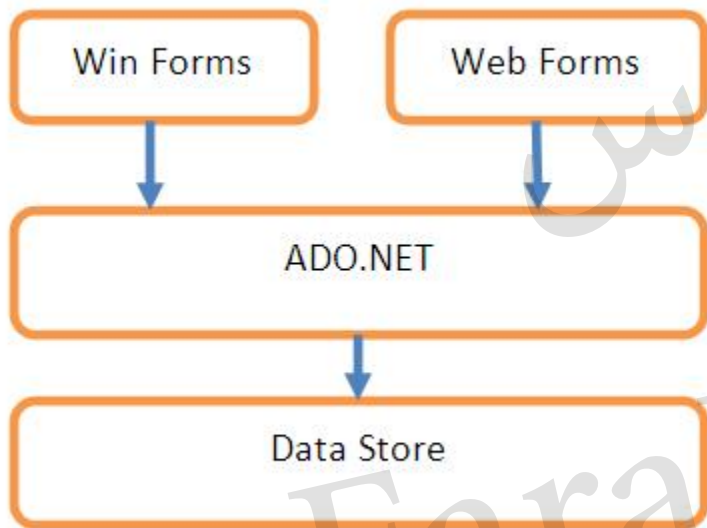
ذخیره و بازیابی اطلاعات یکی از مهمترین بخش‌ها در تولید یک محصول نرم افزاری می‌باشد. امروزه برای ذخیره و بازیابی اطلاعات در حجم زیاد از نرم افزارهای بانک اطلاعاتی همانند Access ,Sql Server Oracle و غیره استفاده می‌شود. در .Net روش‌های مختلفی برای ارتباط با بانک‌های اطلاعاتی ایجاد شده است که از جمله این تکنولوژی‌ها می‌توان ADO.NET، LINQ و... را نام برد. در این بخش ابتدا اجزاء ADO.NET را بررسی کرده و در ادامه با استفاده از این تکنولوژی یک برنامه کاربردی را بصورت کامل پیاده‌سازی خواهیم کرد.

مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

faradars.org/fvcs9311

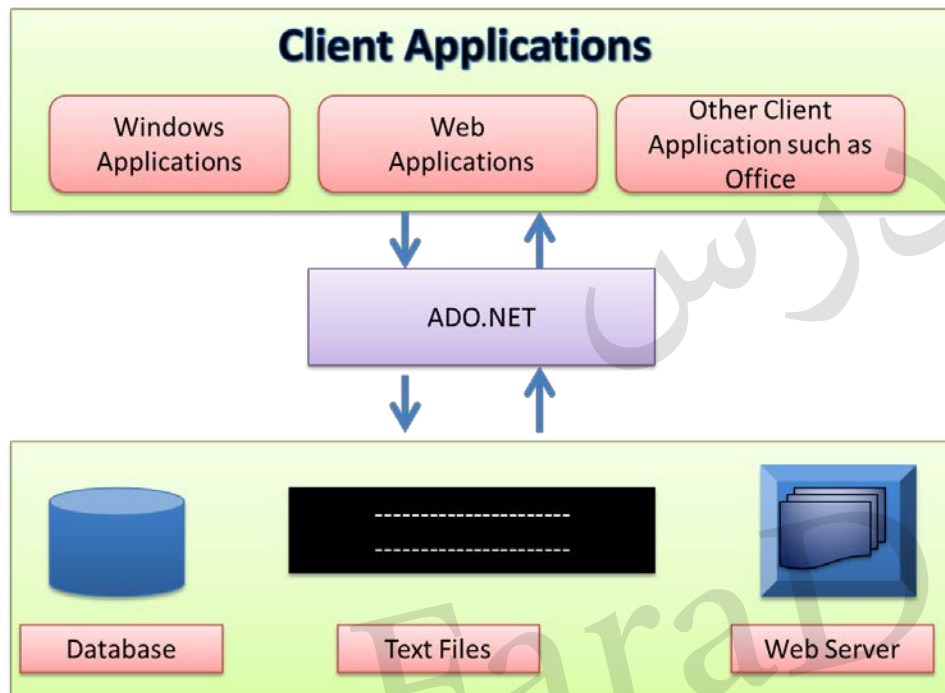


ADO.NET چیست؟



به مجموعه کلاس‌هایی که اشیائی از آن کلاس‌ها برای دسترسی به داده‌های یک بانک اطلاعاتی در .Net استفاده می‌شود، ADO.NET گفته می‌شود.

کاربرد ADO.NET

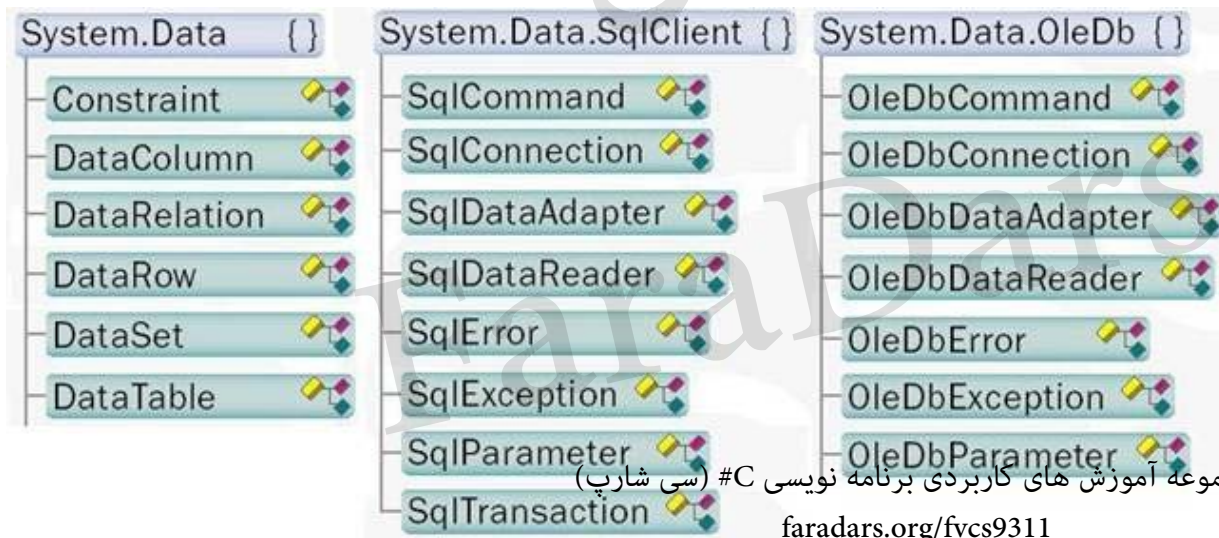


تکنولوژی ADO.NET برای ذخیره و بازیابی داده ها در هر نوع منبع داده ای کاربرد دارد. همچنین این تکنولوژی برای ذخیره بازیابی داده ها در هر نوع برنامه نویسی از تحت وب تا تحت ویندوز کاربرد دارد.



فضای نام های ADO.NET هستند؟

کلاس های اصلی و مشترک ADO در فضای نام **System.Data** قرار دارند. این فضای نام خود نیز شامل چند فضای نام دیگر است که مهمترین آنها عبارتند از **System.Data.SqlClient** و **System.Data.OleDb**. فضای نام **System.Data.SqlClient** حاوی کلاس هایی است که برای دسترسی به بانک های اطلاعاتی از نوع **Sql Server** مورد استفاده قرار می گیرد. فضای نام **System.Data.OleDb** نیز حاوی کلاس هایی است که برای دسترسی به بانک های اطلاعاتی از نوع **OLE** (مانند بانک های اطلاعاتی **Access**) استفاده می شود. در طی این فصل با استفاده از فضای نام **SqlClient** به بانک های اطلاعاتی از نوع **SQL Server** دسترسی خواهیم داشت.



مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

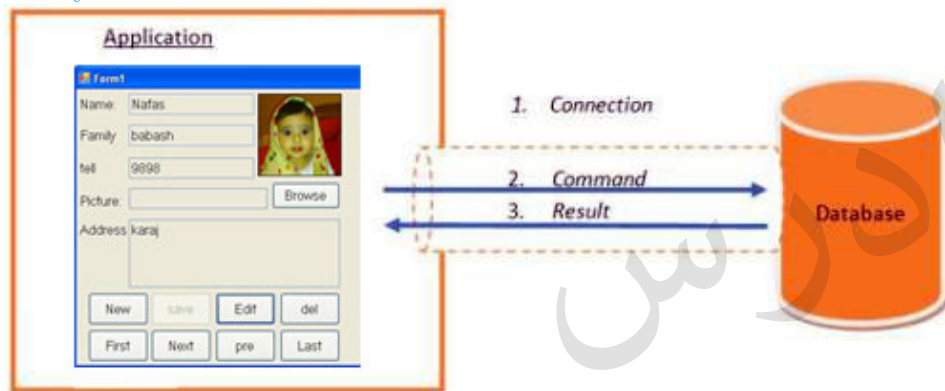
کلاس های بکار برده شده در این بخش

ابتدا با تعدادی از مهمترین کلاس های ADO.NET که در طی مثال های این بخش به کار رفته اند آشنا می شویم.
این کلاس ها عبارتند از:

SqlConnection – SqlCommand – SqlDataAdapter – SqlParameter

این کلاس ها فقط برای ارتباط با بانک های اطلاعاتی SQL Server مورد استفاده قرار می گیرند و در فضای نام System.Data.SqlClient قرار دارند. همچنین از کلاس DataSet استفاده می کنیم که در فضای نام System.Data قرار دارند و برای تمام سرویس دهنده ها قابل استفاده هستند. بنابراین به کمک دستور using فضای نام های مذکور را به شکل زیر به برنامه اضافه می کنیم.

```
using System.Data;  
using System.Data.SqlClient;
```



کلاس SqlConnection

تقریباً می‌توانیم بگوییم که کلاس **SqlConnection** در قلب کلاس‌هایی قرار دارد که در این قسمت مورد استفاده قرار می‌گیرد، زیرا این کلاس وظیفه برقراری ارتباط بین برنامه و بانک اطلاعاتی برنامه را بر عهده دارد.

یک نمونه از این کلاس دارای جزء داده‌ای به نام **ConnectionString** می‌باشد. این خصوصیت به نام رشته اتصال معروف است که از نوع رشته‌ای می‌باشد و تمام داده‌های مورد نیاز برای اتصال به یک بانک اطلاعاتی را شامل می‌شود.

بخش‌های مختلف ConnectionString

ساختار متنی که باید برای ConnectionString مورد استفاده قرار گیرد، بستگی به سرویس‌دهنده‌ی اطلاعاتی دارد که استفاده می‌کنیم. بخش‌های مختلف ConnectionString در جدول مقابل توضیح داده شده است. برای ایجاد امنیت در بانک‌های اطلاعاتی ایجاد شده بوسیله‌ی SQL Server، باید هنگام دسترسی به آنها ابتدا هویت استفاده کننده توسط SQL Server را مشخص کرد. بنابراین اگر بخواهیم توسط یک برنامه به داده‌های موجود در یک بانک اطلاعاتی دسترسی داشته باشیم، باید اطلاعات لازم برای این تعیین هویت را همراه با دیگر اطلاعات در متن ConnectionString مشخص کنیم.

پارامتر	توضیح
Server	نام سرور بانک اطلاعاتی که می‌خواهید از آن استفاده کنید. این پارامتر معمولاً حاوی نام کامپیوتری است که موتور بانک اطلاعاتی SQL Server در آن نصب شده است. اگر SQL Server بر روی همان کامپیوتری که برنامه را اجرا می‌کند نصب شده است، می‌توانید از مقادیری مانند local یا localhost برای این پارامتر استفاده کنید. اما اگر از SQL Server که در کامپیوتر دیگری در شبکه نصب شده است استفاده می‌کنید، لازم است که مقدار این پارامتر را برابر با نام آن کامپیوتر در شبکه قرار دهید. همچنین اگر در آن کامپیوتر بیش از یک SQL Server قرار داشته باشد، باید بعد از نام کامپیوتر یک علامت \ قرار داده و سپس نام SQL Server که می‌خواهید مورد استفاده قرار دهید را ذکر کنید.
Database	نام بانک اطلاعاتی که می‌خواهید مورد استفاده قرار دهید در این پارامتر قرار می‌گیرد (برای مثال بانک اطلاعاتی Phonebook).
User ID	این پارامتر باید حاوی نام کاربری باشد که می‌خواهیم برای اتصال به بانک اطلاعاتی از آن استفاده کنیم. برای اینکه بتوانیم با استفاده از این روش از بانک اطلاعاتی استفاده کنیم، باید یک حساب کاربری به این نام در SQL Server ایجاد شده و اجازه‌ی دسترسی به داده‌های مورد نیاز نیز به آن داده شود.
Password	کلمه‌ی عبوری که برای این نام کاربری مورد استفاده قرار می‌گیرد.

مجموعه آموزش‌های کاربردی برنامه نویسی C# (سی شارپ)

مثالی از نحوه ایجاد یک شی اتصال و نحوه استفاده از آن

بعد از اینکه با ایجاد `ConnectionString` نحوه‌ی برقراری ارتباط با بانک اطلاعاتی را مشخص کردیم، می‌توانیم با استفاده از متدهای `Open` و `Close` در کلاس `SqlConnection` به بانک اطلاعاتی متصل شده و یا اتصال خود را قطع کنیم. یک نمونه از این کار در قطعه کد زیر نشان داده شده است:

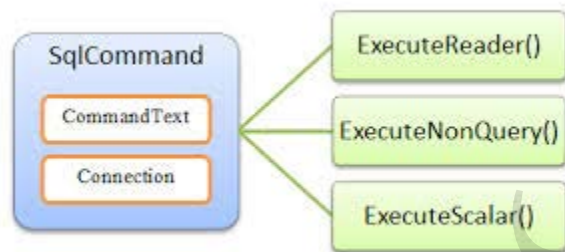
البته متدها و خواص فراوان دیگری در کلاس `SqlConnection` وجود دارند که می‌توانیم در برنامه از آنها استفاده کنیم، اما مواردی که در اینجا با آنها آشنا شدیم پرکاربردترین آنها به شمار می‌روند.

```
SqlConnection conn = new SqlConnection();  
conn.ConnectionString = "رشته اتصال";  
conn.Open();  
.  
.  
.  
conn.Close();
```

مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)



کلاس SqlCommand

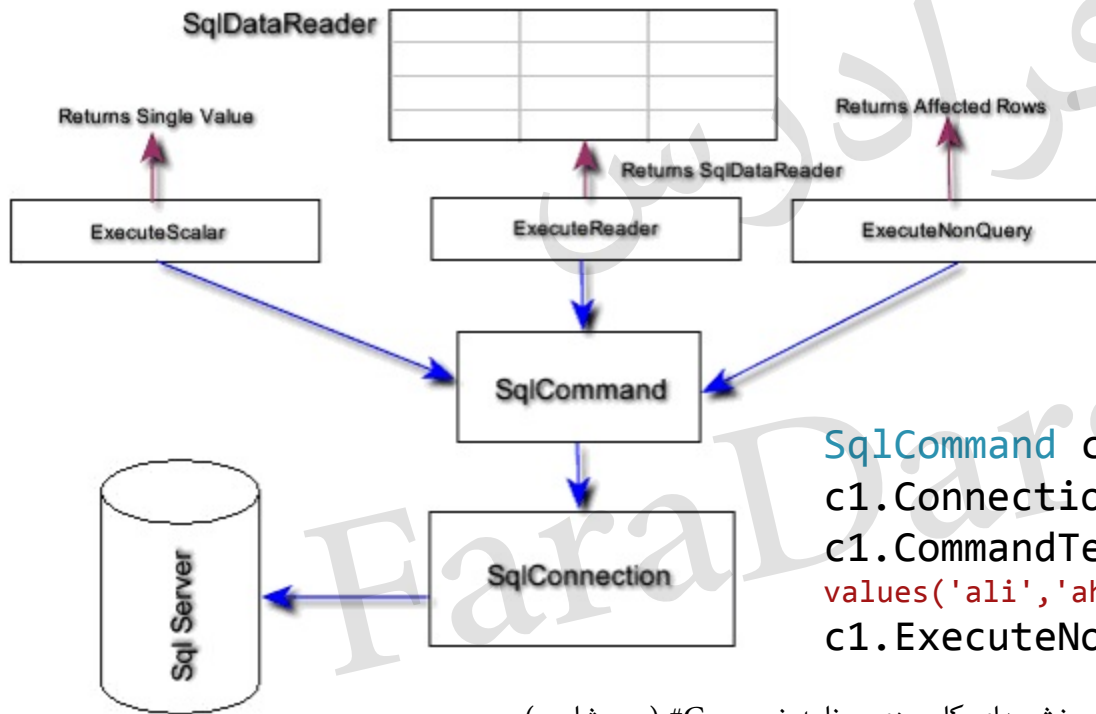


کلاس `SqlCommand` حاوی یک دستور `SQL` برای اعمال تغییرات بر روی داده‌های دریافت شده از بانک اطلاعاتی است. این دستور می‌تواند یک دستور `SELECT` برای انتخاب داده‌هایی خاص، یک دستور `INSERT` برای درج داده‌های جدید در بانک اطلاعاتی، یک دستور `DELETE` برای حذف داده‌ها از بانک اطلاعات و یا حتی فراخوانی یک پروسیجر ذخیره شده در بانک اطلاعاتی باشد. دستور `SQL`ی که در این کلاس نگهداری می‌شود می‌تواند شامل پارامترهایی نیز باشد. به شکل مقابل می‌توانیم یک شی از نوع دستور تعریف کنیم.

خاصیت `CommandText` حاوی متنی است که می‌تواند یک دستور `SQL` و یا نام یک پروسیجر ذخیره شده در بانک اطلاعاتی باشد که باید بر روی داده‌ها اجرا شود. خاصیت `Connection` مرجع به یک شی اتصال باز می‌باشد که می‌خواهیم دستوری در بانک اطلاعاتی مرتبط با آن شی اتصال اجرا گردد.

```
SqlCommand c1 = new SqlCommand();
c1.Connection = conn;
c1.CommandText = "insert into Phones values('ali','ahmadi',0912..','tehran..')";
```

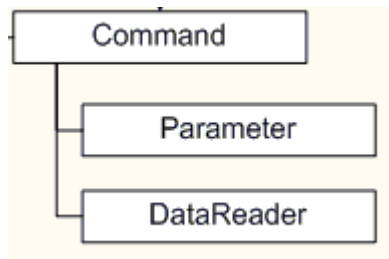
متدهای کلاس SqlCommand



`ExecuteNonQuery();`
`ExecuteReader();`
`ExecuteScalar();`

نمونه ای از برنامه دکمه ذخیره

```
SqlCommand c1 = new SqlCommand();
c1.Connection = conn;
c1.CommandText = "insert into Phones
values('ali','ahmadi',0912..','tehran..')";
c1.ExecuteNonQuery();
```



پارامترها متغیرهایی هستند که در یک دستور SQL قرار می‌گیرند و می‌توانند در زمان اجرای برنامه جای خود را با عباراتی خاص عوض کنند. این متغیرها با علامت @ در یک دستور مشخص می‌شوند.

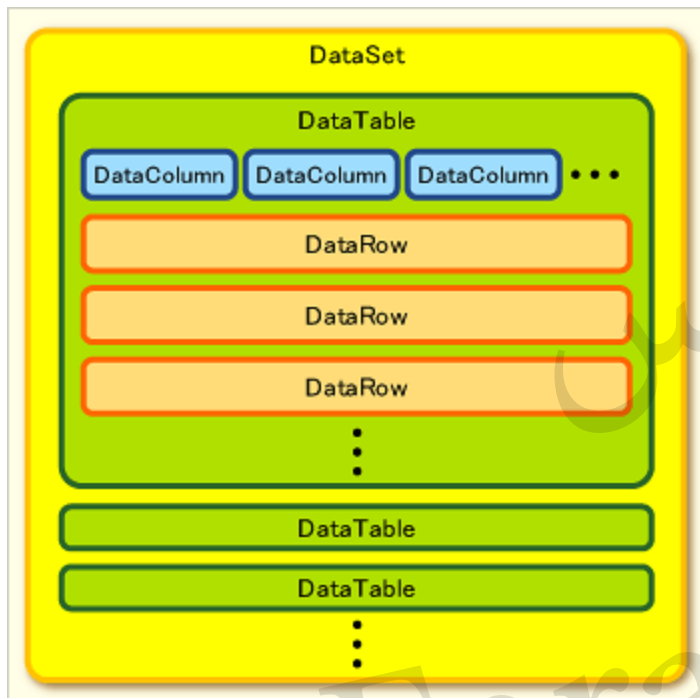
```

SqlCommand c1 = new SqlCommand();
c1.CommandText = "insert into Phones values(@p1,@p2,@p3,@p4)";
c1.Connection = conn;
c1.CommandType = CommandType.Text;
c1.Parameters.AddWithValue("p1", txtname.Text);
c1.Parameters.AddWithValue("p2", txtfamily.Text);
c1.Parameters.AddWithValue("p3", txttoll.Text);
c1.Parameters.AddWithValue("p4", txtaddress.Text);
c1.ExecuteNonQuery();
  
```

متد **AddWithValue** نام یک پارامتر و متغیری که مقدار مربوط به آن را در زمان اجرای برنامه نگهداری می‌کند به

عنوان پارامتر دریافت کرده و آن را به لیست **Parameters** اضافه می‌کند. # (سی شارپ)

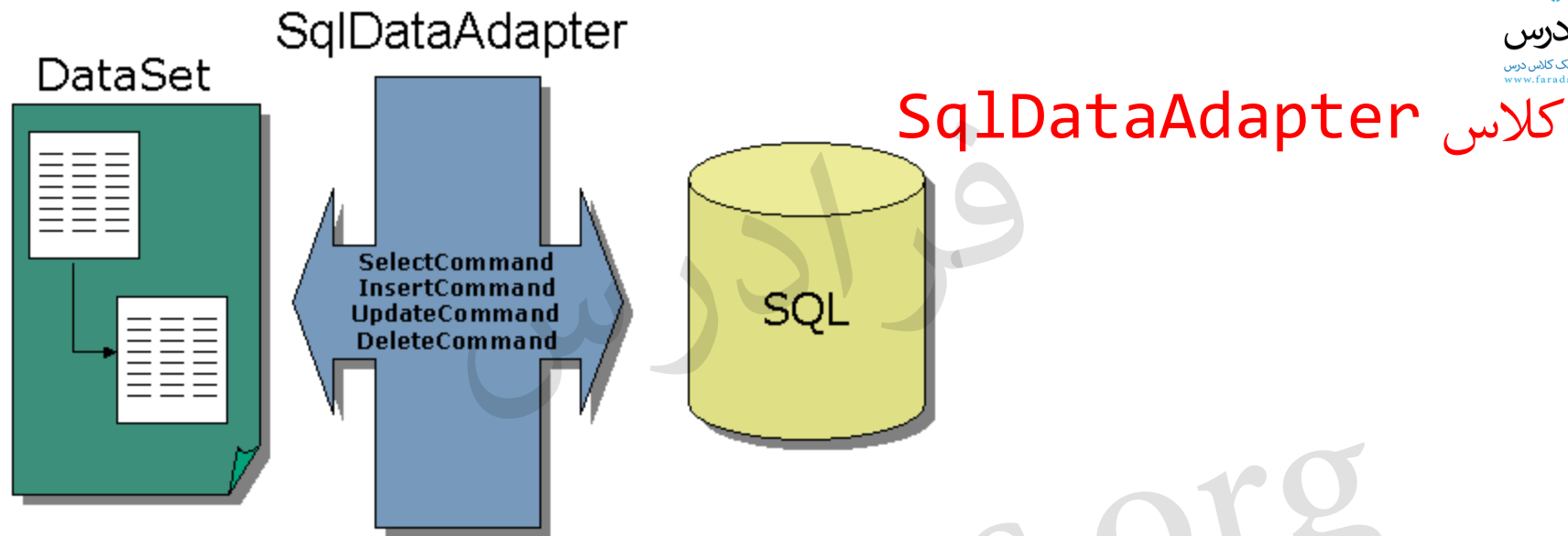
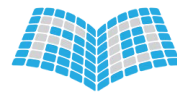
کلاس DataSet



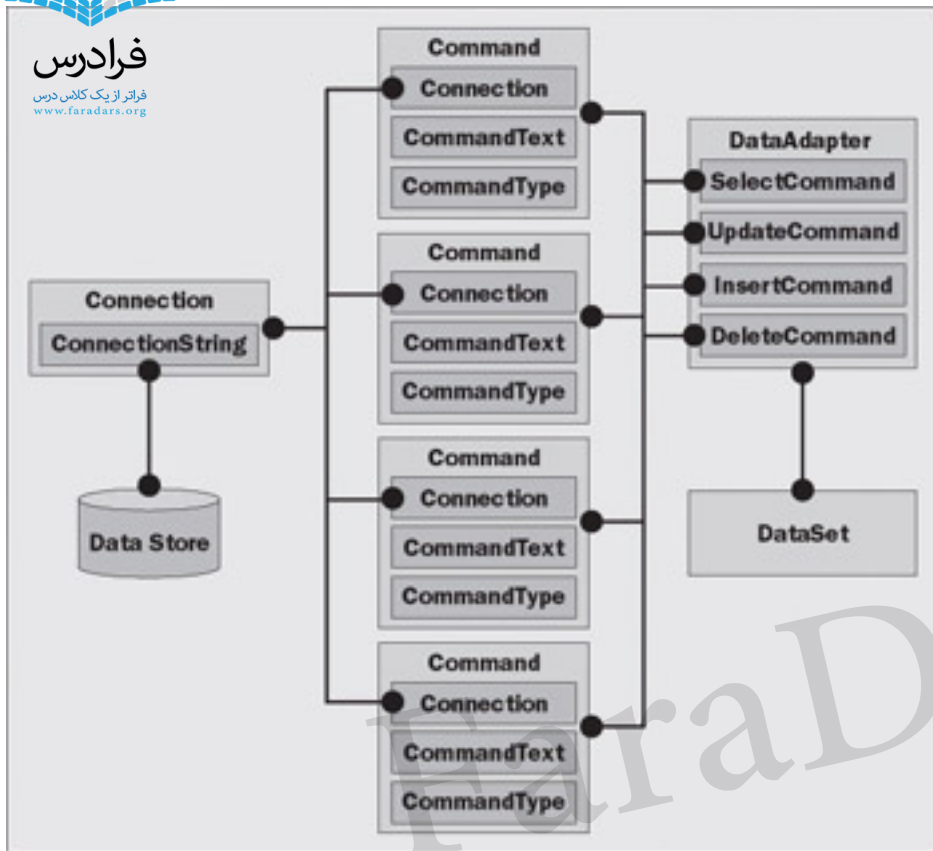
همانطور که مشخص است یک شی DataSet می تواند شامل چندین جدول باشد که هر یک از آنها به وسیله یک کنترل DataTable مشخص می شوند. دقت کنید که ابتدای کلاس DataSet کلمه ی Sql وجود ندارد. دلیل این مورد هم این است که این کلاس متعلق به فضای نام System.Data.SqlClient نیست بلکه در فضای نام System.Data قرار دارد. به عبارت دیگر، کلاس DataSet به سرویس دهنده ی اطلاعاتی خاصی از قبیل SqlConnection یا OleDb تعلق ندارد و وظیفه ی آن نگهداری اطلاعات به دست آمده به هر نحوی در حافظه است. هنگامی که اطلاعات را در حافظه قرار دادیم دیگر نیازی

```
DataSet ds = new DataSet();
```

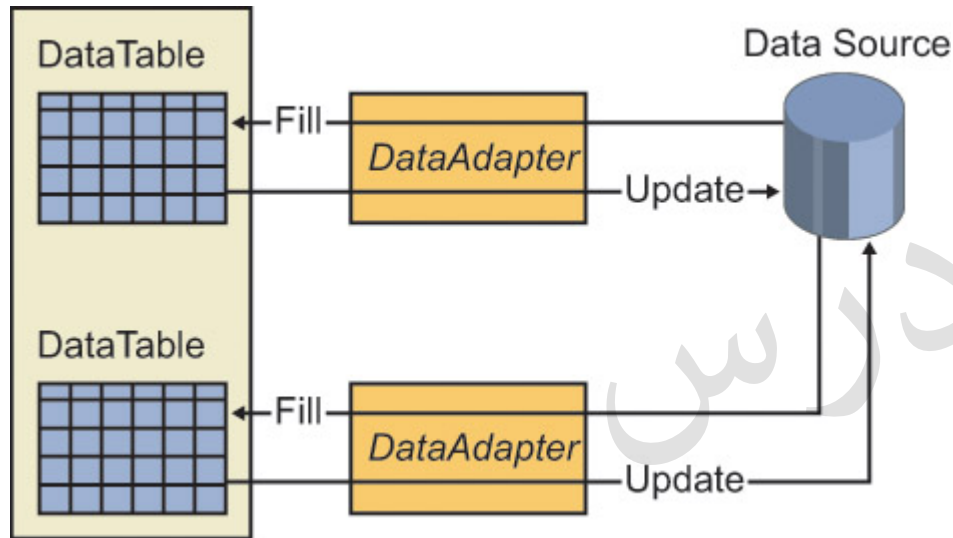
نیست بدانیم که این اطلاعات از کجا به دست آمده اند (سی شارپ)



کلاس **DataAdapter** در برنامه‌های بانک اطلاعاتی، همانند پلی بین جداول بانک اطلاعاتی و نیز داده‌های موجود در حافظه که به وسیله‌ی **DataSet** نگهداری می‌شوند، عمل می‌کند. می‌توانیم بگوئیم که کلاس **DataAdapter** برای دسترسی به بانک اطلاعاتی از کلاس **SqlConnection** و **SqlCommand**



کلاس **DataAdapter** دارای خاصیتی به نام **SelectCommand** است که مرجع به شیئی از نوع دستور می باشد که **DataAdapter** آنرا بر روی بانک اطلاعاتی اجرا کرده و نتایج آن را در کلاس هایی مانند **DataSet** و یا **DataTable** قرار می دهد تا در برنامه ها مورد استفاده قرار گیرند. کلاس **DataAdapter** دارای خاصیت هایی به نام **DeleteCommand**، **InsertCommand** و **UpdateCommand** است که هر یک شیئی از نوع **SqlCommand** را قبول می کنند و **DataAdapter** از دستور ذخیره شده در هر یک از آنها به ترتیب برای حذف، درج و یا ویرایش داده ها در بانک اطلاعاتی استفاده می کند. در حقیقت هنگامی که ما در طی برنامه تغییراتی را در درون داده های موجود در حافظه نگهداری می کنیم، **DataAdapter** با استفاده از دستورات موجود در این خاصیت ها تغییرات ما را از داده های حافظه به داده های موجود در بانک اطلاعاتی منتقل می کند.



متد Fill از کلاس DataAdapter

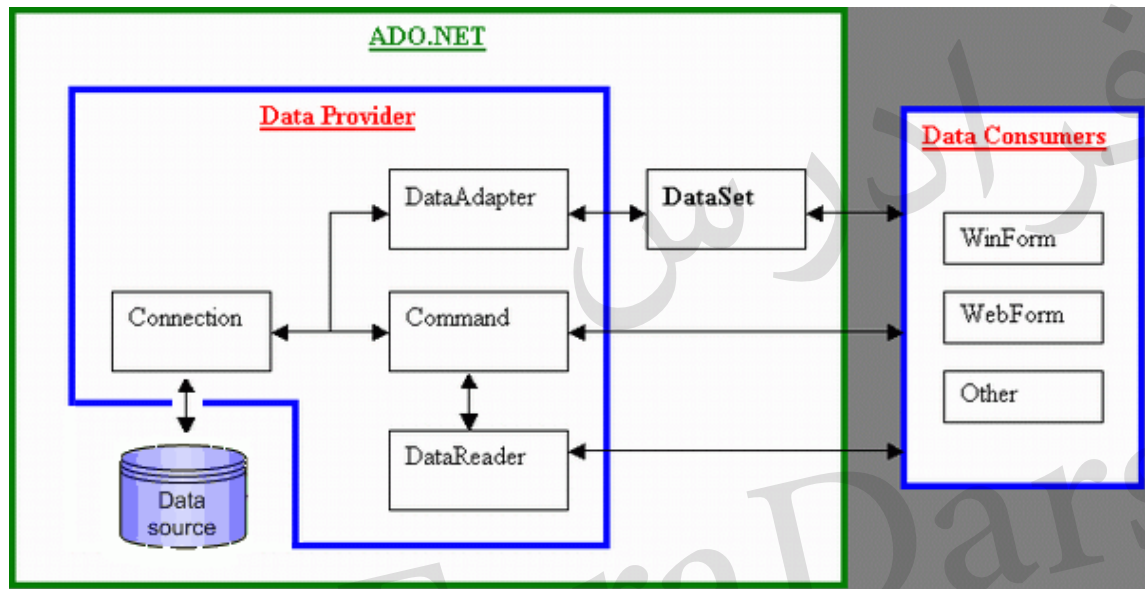
با استفاده از متد Fill از کلاس DataAdapter می‌توان دستور SQL موجود در خاصیت SelectCommand را در بانک اطلاعاتی اجرا کرده و سپس داده‌های برگشتی از اجرای این دستور را درون یک DataSet در حافظه قرار دهیم. هنگامی که با استفاده از DataAdapter داده‌هایی را درون یک DataSet قرار می‌دهید، ابتدا یک شیء جدید از نوع DataTable ایجاد شده، داده‌ها را درون آن قرار داده میشوند و DataSet اضافه میشوند.

```

SqlConnection conn = new SqlConnection();
conn.ConnectionString = "رشته اتصال";
SqlDataAdapter da = new SqlDataAdapter();
DataSet ds = new DataSet();
SqlCommand c1 = new SqlCommand();
c1.Connection = conn;
c1.CommandText = "Select * from Telbook";
da.SelectCommand = c1;
da.Fill(ds, "T1");
  
```

مجموعه آموزش‌های کاربردی برنامه نویسی C# (سی شارپ)

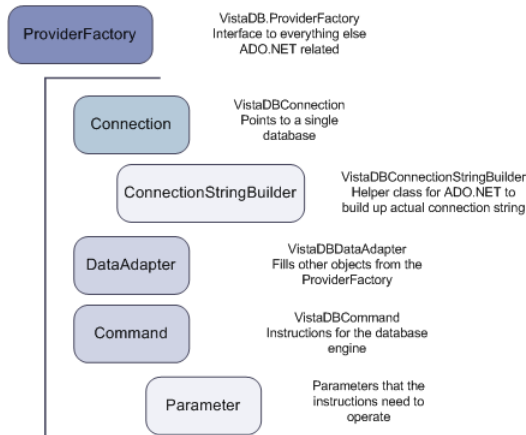
تصور ساده‌ای از کلاس‌های ADO.NET



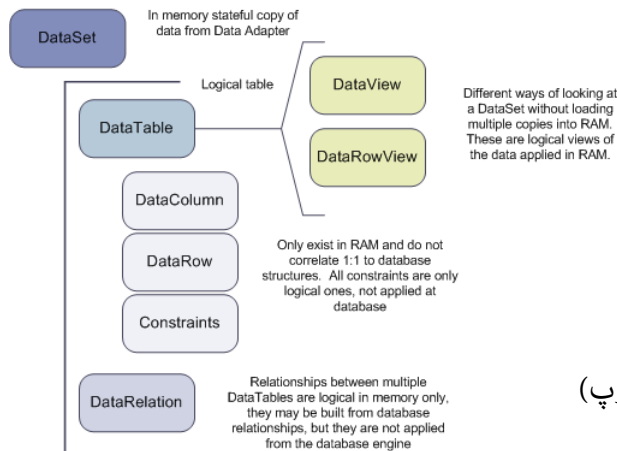
کلاس‌های موجود در ADO.NET به حدی زیاد هستند که نمی‌توانیم تمام آنها را در این بخش معرفی کنیم. با این وجود تمام کلاس‌های مهم که برای مثال کاربردی در ادامه نیاز داریم در این بخش معرفی شده‌اند و ارتباط آنها را در شکل مقابل مشاهده می‌کنید..



Connected ADO.NET Classes



Disconnected ADO.NET Classes



استفاده ADO.NET از معماری غیرمتصل

ADO.NET برای دسترسی به داده‌ها از معماری غیرمتصل استفاده می‌کند. غیرمتصل به این معنی است که ابتدا برنامه به موتور بانک اطلاعاتی مورد نظر متصل شده، داده‌های مورد نیاز خود را از بانک اطلاعاتی دریافت کرده و آنها را در حافظه کامپیوتر ذخیره می‌کند. سپس اتصال برنامه از بانک اطلاعاتی قطع می‌شود و تغییرات مورد نظر خود را در داده‌های موجود در حافظه انجام می‌دهد. هر زمان که لازم باشد تغییرات ایجاد شده در بانک اطلاعاتی ذخیره شوند، برنامه یک اتصال جدید را با بانک اطلاعاتی ایجاد کرده و از طریق این اتصال تغییراتی که در داده‌ها اعمال کرده بود را در جدول اصلی پیاده‌سازی می‌کند. شیئی اصلی که داده‌های دریافتی از بانک اطلاعاتی را در حافظه نگهداری می‌کند، شیئی DataSet است.

مجموعه آموزش‌های کاربردی برنامه نویسی C# (سی شارپ)



فرادرس

فراتر از یک کلاس درس
www.faradars.org

استفاده از کلاس‌های ADO.NET در عمل

تا این قسمت از کار تنها بصورت تئوری کلاس‌های موجود در ADO.NET را معرفی کردیم. در ادامه با طراحی و پیاده‌سازی یک دفترچه تلفن مطمئن می‌شویم که نحوه استفاده از این کلاس‌ها، متدها، خاصیت‌ها... را درست درک کرده‌ایم.

Form1

Name:

Family:

Tell:

Address:

search by: LastName

Find:

Search

New Save Edit DEI

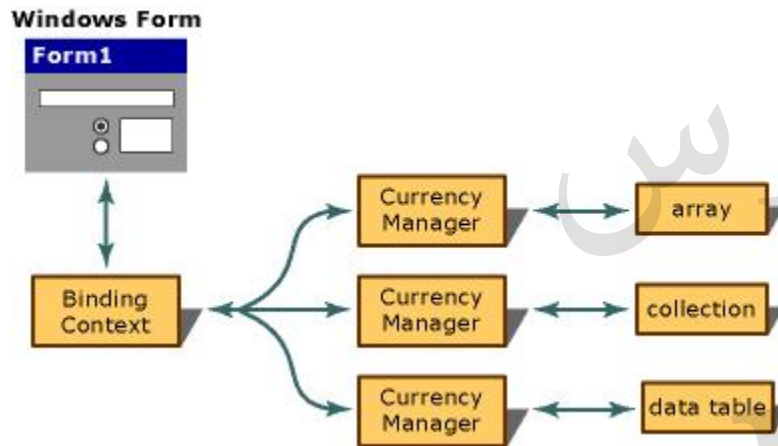
Next Pre Last

مجموعه آموزش‌های کاربردی برنامه نویسی #C (سی شارپ) Faradars.org

faradars.org/fvcs9311



BindingContext و CurrencyManager



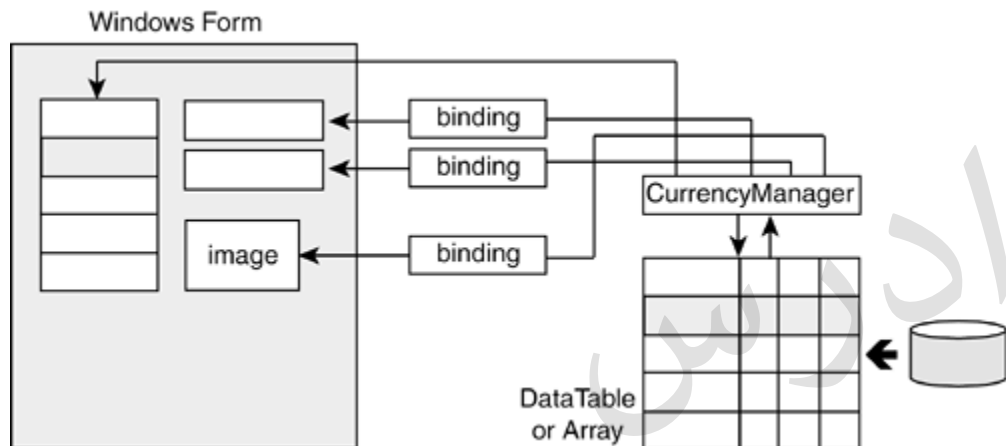
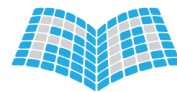
هر فرم دارای شیئی از نوع BindingContext است که اتصالات کنترل‌های درون فرم را مدیریت می‌کند. بنابراین به علت اینکه فرم برنامه شما به صورت درونی دارای چنین شیئی است، نیازی نیست که آن را ایجاد کنید.

شیء BindingContext در حقیقت یک مجموعه از اشیاء از نوع CurrencyManager را مدیریت می‌کند. وظیفه‌ی CurrencyManager نیز این است که بین تمام کنترل‌هایی که به یک منبع داده (مثلاً DataSet) متصل هستند هماهنگی برقرار کند.

CurrencyManager cr;

cr = (CurrencyManager)this.BindingContext[ds, "t1"];

مجموعه آموزش‌های کاربردی برنامه نویسی C# (سی شارپ)



DataBindings

به کمک متد Add از DataBindings کنترل ها می توان برای آنها منبع داده تعیین کرد. شکل کلی این متد به شکل زیر است.

```
txtname.DataBindings.Add(|
```

```
2 of 7 Binding ControlBindingsCollection.Add (string propertyName, object dataSource, string dataMember)  
propertyName: The name of the control property to bind.
```

ورودی اول، نام خصوصی است که می خواهید برای آن منبع داده تعریف کنید. پارامتر دوم، نام DataSet جاری می باشد و پارامتر سوم، نام جزئی از DataSet است که می خواهید به عنوان منبع داده تعریف کنید.

مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)



تمرین: برنامه حذف رکورد را به نحوی تغییر دهید که همانند شکل زیر قبل از حذف رکورد، به کمک یک messagebox از کاربر تأییدیه حذف رکورد مذکور را دریافت کند.





تمرین: برنامه دفترچه تلفن را به نحوی تغییر دهید که تصویر مشترک تلفن را مطابق شکل زیر ثبت کند. دقت کنید که این بخش باید در تمام قسمت‌های برنامه (ویرایش، حذف، بعدی، قبلی و...) درست کار کند.

Firstname	lastname	Phone
Ali	Ahmadi	123
Rashid	Shojaei	12345
babak	amri	21232
alireza	amani	4342
Nafas	babash	9898

مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)



فرادرس

فراگیر از یک کلاس درس
www.faradars.org

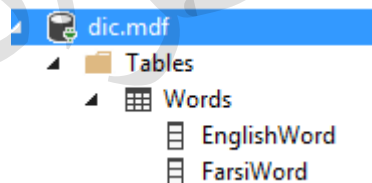
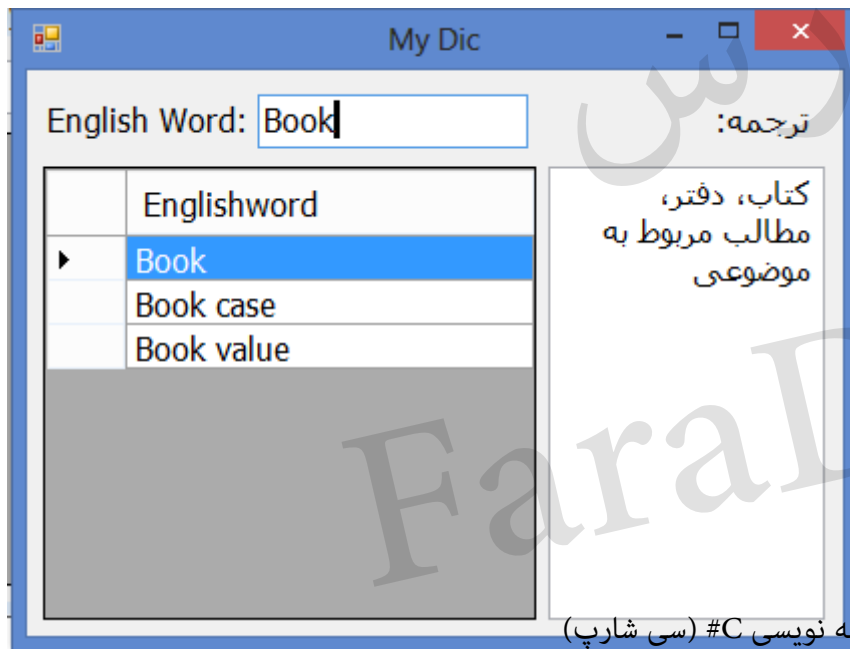
تمرین: برنامه دفترچه تلفن را به نحوی تغییر دهید که با کلیک کردن بر روی تصویر، عکس شخص با ابعاد واقعی نمایش داده شود.



مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)



تمرین: برنامه‌ای بنویسید که یک دیکشنری را بطور کامل پیاده‌سازی کند. بانک اطلاعاتی این دیکشنری حاوی یک جدول بنام Words با شکل زیر است.



EnglishWord	FarsiWord
Book	کتاب دفتر مطالب مربوط به موضوعی
Book case	جا کتابی، کمد کتاب، قفسه کتاب
Book value	ارزش دارایی شرکت، ارزش طبق دفاتر، ارزش هر سهم شرکت
IT	فناوری اطلاعات
Template	قالب
OS	سیستم عامل
Word processor	واژه پرداز

فرم اصلی دیکشنری به شکل مقابل است
مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)



فرادرس

فراگیر از یک کلاس درس
www.faradars.org

معرفی تکنولوژی هایی که می توانید در ادامه با زبان C# آموزش ببینید. هر عنوان معرفی شده یک تخصص می باشد.

- WPF (Windows Presentation Foundation) with C#
- WCF (Windows Communication Foundation) with C#
- WF (Windows Workflow Foundation)
- Socket Programming with C#
- Multithreaded Programming with C#
- Multilayer applications(N-Tier programming) with C#
- ASP.NET MVC Web Application with C#
- LINQ (Language-Integrated Query) with C#
- Programming Entity Framework with C#
- Developing Mobile iOS Applications With Xamarin and C#
- Building Android Apps In C# With Xamarin

پیشنهاد شما برای تهیه آموزش:
اگر هر یک از تکنولوژی های معرفی شده در این بخش را مناسب فعالیتهای آینده خود دیدید، می توانید درخواست تهیه آن را از طریق آدرس زیر مطرح نمایید. ما بر حسب میزان درخواست ها، انتشار آنها را پیگیری خواهیم کرد.

مدرس: مهندس رشید شجاعی

مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

faradars.org/fvcs9311

<http://faradars.org/course-request/>



فرادرس

فراتر از یک کلاس درس
www.faradars.org

مدرس: مهندس رشید شجاعی



منبع آموزش:

کتاب «آموزش مباحث ویژه در C#.Net/مولفین رشید شجاعی، نگار براهویی»

مراکز توزیع کتاب:

فروشگاه تهران: تهران میدان انقلاب، روبروی سینما بهمن، نبش پاساژ اندیشه، کتاب دهخدا. تلفن: ۰۹۱۲۲۱۰۲۸۳۵ (جهت سفارش عمده)
فروشگاه کرج: کرج رجایی شهر روبروی درب دانشگاه آزاد اسلامی، کتاب فروشی خوارزمی و ایران

راه های ارتباطی با مدرس:

Web Page: <http://kiauu.ac.ir/~shojaei>

Email: Shojaei@kiauu.ac.ir

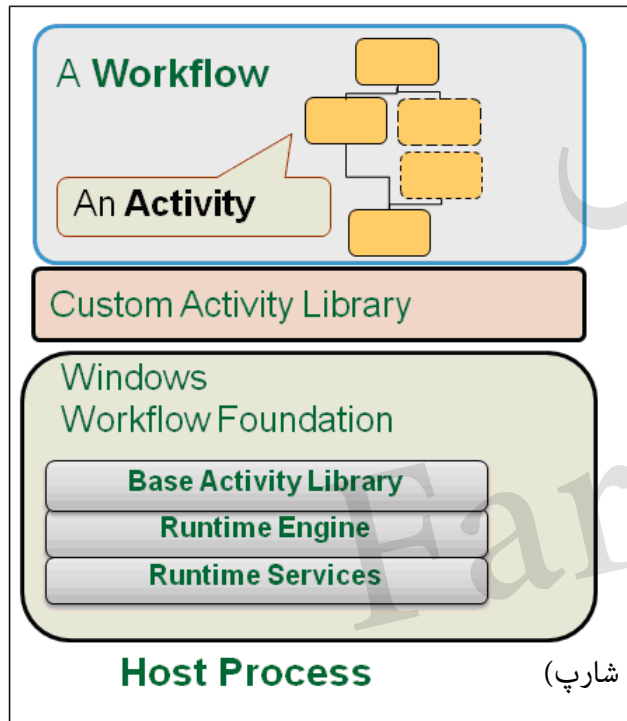
Email: Rashid.Shojaei@gmail.com

مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

faradars.org/fvcs9311

WF (Windows Workflow Foundation)

Visual Designer



Windows Workflow Foundation یا WF برای پشتیبانی از Workflow برنامه‌ای ایجاد شده است. WF کمک می‌کند که هر کدام از تکه‌های خیلی بزرگ نرم‌افزار به یک Activity تبدیل شود. این تکه‌ها بعداً در قالب یک Workflow با هم ارتباط پیدا می‌کنند. نهایتاً این Workflow در جایی host شده و توسط WF Runtime شروع به اجرا می‌کند. WF مدعی است که به برنامه‌نویسان امکان می‌دهد تا بخش‌های مختلف Workflow را به طور موازی و در کامپیوترهای مختلف به اجرا درآورند. در واقع شعار WF نوشتن برنامه‌های بزرگی است که به راحتی برنامه‌های کوچک قابل توسعه و نگهداری هستند.

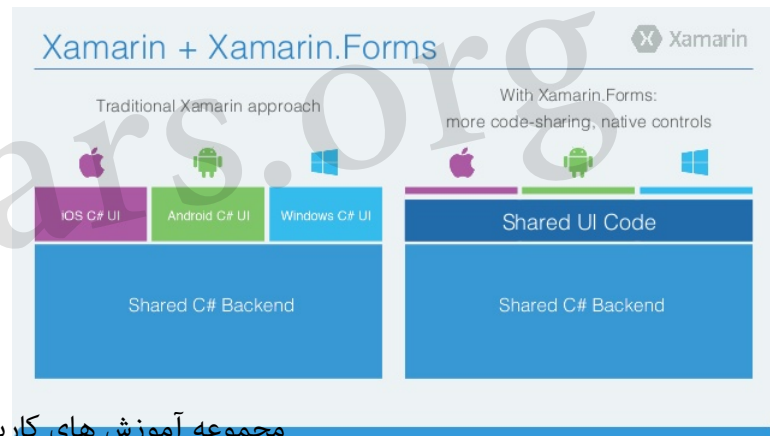
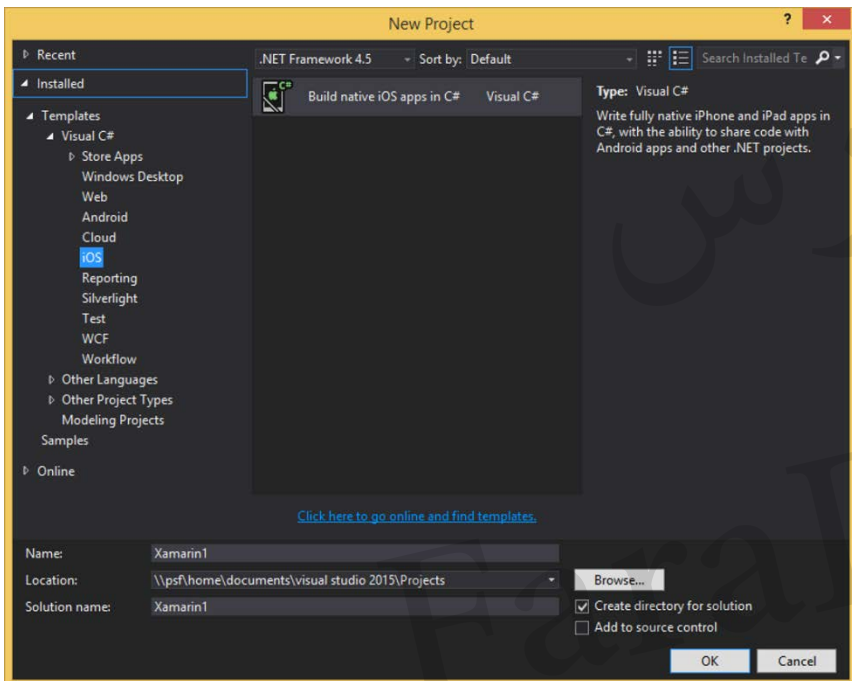
نرم افزار SharePoint و سایر محصولات مایکروسافت که مرتبط با گردش کار می باشند از یک زیرساخت مشترک جهت مدیریت گردش کار با نام Windows Workflow Foundation استفاده می کنند. WF یک چهارچوب جهت تولید گردش کارهای سیستمی و یا انسانی است که شامل مجموعه ای از API ها، موتور گردش کار درون فرایندی، و محیط طراحی می باشد. گردش کار شامل مجموعه ای از اجزا می باشد که در WF به هر یک از این اجزا فعالیت (Activity) گفته می شود. مایکروسافت کتابخانه ای از فعالیت ها را در .Net Framework ایجاد کرده و امکان توسعه فعالیت های شخصی را نیز فراهم کرده است.

مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)



Developing Mobile iOS Applications With Xamarin and C#

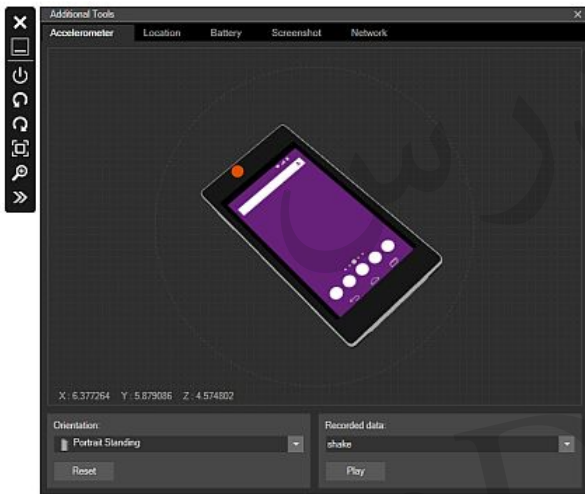
Xamarin را میتوان بهترین ابزار توسعه اندروید با پایتون دانست. با این Framework میتوان به زبانهای C# برای اندروید برنامه نویسی کرد. Xamarin یک شرکت نرم افزاری است که توسط مهندسان Mono، MonoTuch، Mono for Android ایجاد شد. Xamarin هم در CLI اجرا میشود و هم در محیط Visual Studio. در حال حاضر Xamarin به Visual Studio 2015 افزوده شده است.



مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)



Building Android Apps In C# With Xamarin

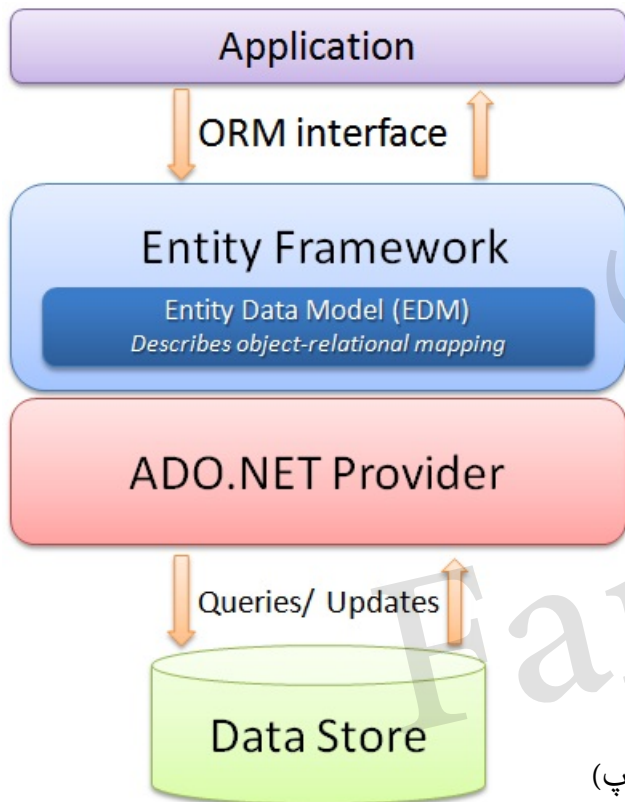


Xamarin را میتوان بهترین ابزار توسعه اندروید با پایتون دانست. با این Framework میتوان به زبانهای C# برای اندروید برنامه نویسی کرد. Xamarin یک شرکت نرم افزاری است که توسط مهندسان Mono، MonoTuch، Mono for Android ایجاد شد. Xamarin هم در CLI اجرا میشود و هم در محیط Visual Studio. در حال حاضر Xamarin به Visual Studio 2015 افزوده شده است.



تصاویری از **Emulator** اندروید که برای تست و دیباگ کردن برنامه های نوشته شده جهت سیستم عامل **Android**. این شبیه ساز همراه نصب **Visual Studio 2015** نصب می شود.

Programming Entity Framework

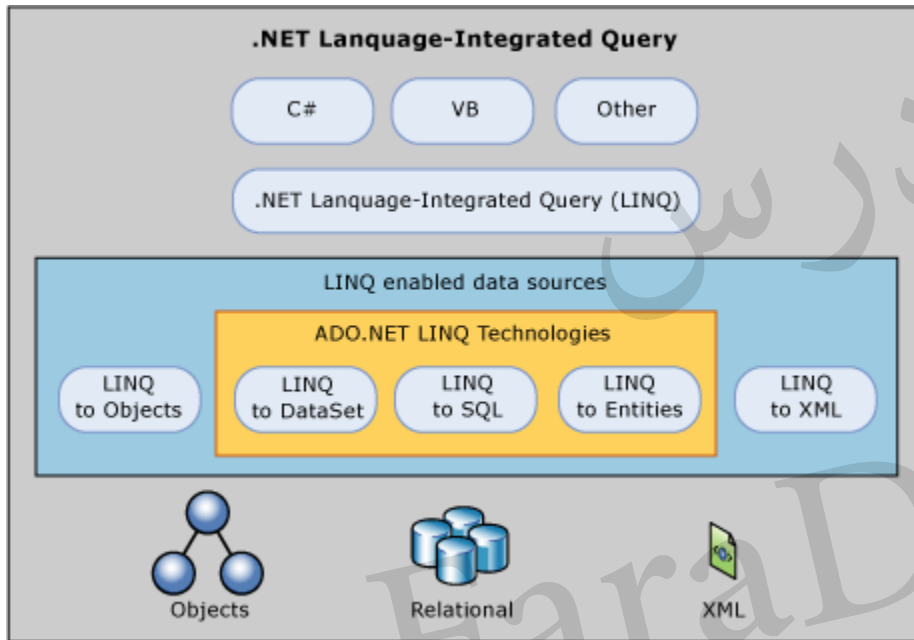


قبل از این که بخواهیم با Entity Framework آشنا شویم، بهتر است مفهوم ORM را بدانیم. ORM یا Object Relational Mapping ابزاری است که به وسیله آن می توان جداول پایگاه داده را پردازش کرد و آنها را به فرمت شیء نگاشت کرد که در کدنویسی خواناتر باشند. در کدنویسی به جای این که به صورت مستقیم با جداول پایگاه داده کار کنیم بر روی اشیاء کار می کنیم. به این اشیاء موجودیت یا Entity گفته می شود.

Entity Framework یک ORM است که توسط شرکت Microsoft توسعه داده میشود.

درواقع Entity Framework تکنولوژی توسعه یافته ADO.Net است که فاصله بین برنامه نویسی شی گرا و بانک اطلاعاتی رابطه ای را پر می کند. این فاصله معمولاً تحت عنوان عدم تطابق شناخته می شود. و یک تکنیک برنامه نویسی برای تبدیل ارتباطات در Database به مفاهیم Object Oriented در برنامه نویسی است. در واقع می توان گفت که کلاس ها را به Table ها map می کند. وقتی که شما می خواهید به Database دسترسی پیدا کنید، یا اطلاعاتی را ذخیره کنید، این کارها را مستقیماً بر روی اشیاء (Object) انجام می دهید. مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

LINQ (Language-Integrated Query)



هدف از ابداع فناوری LINQ، ارائه تسهیلات لازم برای پیاده کنندگان جهت کار با داده در بانک های اطلاعاتی SQL و XML عنوان شده است. در واقع، وجود یک حلقه گمشده بین داده رابطه ای (بانک های اطلاعاتی)، اسناد XML با زبان های برنامه نویسی، ضرورت ابداع فناوری فوق را توجیه کرده بود. پیاده کنندگان مجبور بودند برای کار با هر یک از منابع داده، از روش های مختلفی استفاده نمایند. LINQ را می توان به منزله حلقه ای گمشده بین دنیای داده و زبان های برنامه نویسی همه منظوره تصور کرد. با استفاده از LINQ امکان دستیابی به داده صرف نظر از نوع داده، با روشی مشابه و یکسان فراهم می گردد. پیاده کنندگان نرم افزار در زمان نوشتن برنامه های خود معمولاً از دو زبان مختلف جهت رسیدن به یک هدف مشترک استفاده می نمایند. ما برای نوشتن کدها از یک زبان برنامه نویسی نظیر VB.NET و یا C# و برای گفتگو با سیستم بانک اطلاعاتی از یک زبان دیگر نظیر SQL استفاده می کنیم.

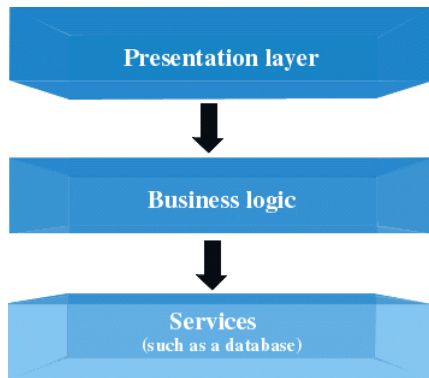
LINQ یک فناوری قدرتمند در زمان کار با داده است که دستاوردهای متعددی دارد:

- روشی ساده جهت نوشتن query

- تسریع در پیاده سازی نرم افزار با توجه به حذف خطاهای زمان اجراء
- امکان استفاده از امکاناتی نظیر اشکال زدائی و IntelliSense در زمان پیاده سازی
- حذف خلاء موجود بین داده رابطه ای و پیاده سازی شی گرا
- استفاده از یک گرامر یکسان جهت نوشتن query صرفنظر از نوع منبع داده

• افزایش بازدهی و راندمان برنامه نویسی، چرا که برنامه نویسان از یک رویکرد یکسان برای نوشتن query و بهنگام سازی داده از طریق زبان برنامه نویسی استفاده خواهند کرد. مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

Multilayer applications



برنامه‌نویسی با معماری چند لایه یکی از بهترین روش‌های برنامه‌نویسی بوده که امروزه شرکت‌های نرم‌افزاری بزرگ جهان مانند ماکروسافت آنرا توصیه نموده و در سیستم‌های برنامه‌سازی خود مانند VS امکانات خاصی برای پیاده‌سازی آن قائل شده‌اند. در این روش، برنامه به سه لایه کلی تقسیم می‌گردد. در این حالت، کاربر که با فرم‌ها در ارتباط می‌باشد به هیچ عنوان با ساختار بانک اطلاعاتی در ارتباط نبوده و این لایه‌های مختلف هستند که خدمات مورد نیاز را تأمین می‌نمایند.

در معماری سه لایه، نرم افزار از سه بخش مجزا به صورت فیزیکی یا منطقی تشکیل شده است که حداکثر استقلال را از یکدیگر دارند و این مهمترین مزیت در توسعه معماری سه لایه نرم افزار است. معمولاً لایه‌ها با اسامی زیر نام گذاری می‌شوند.

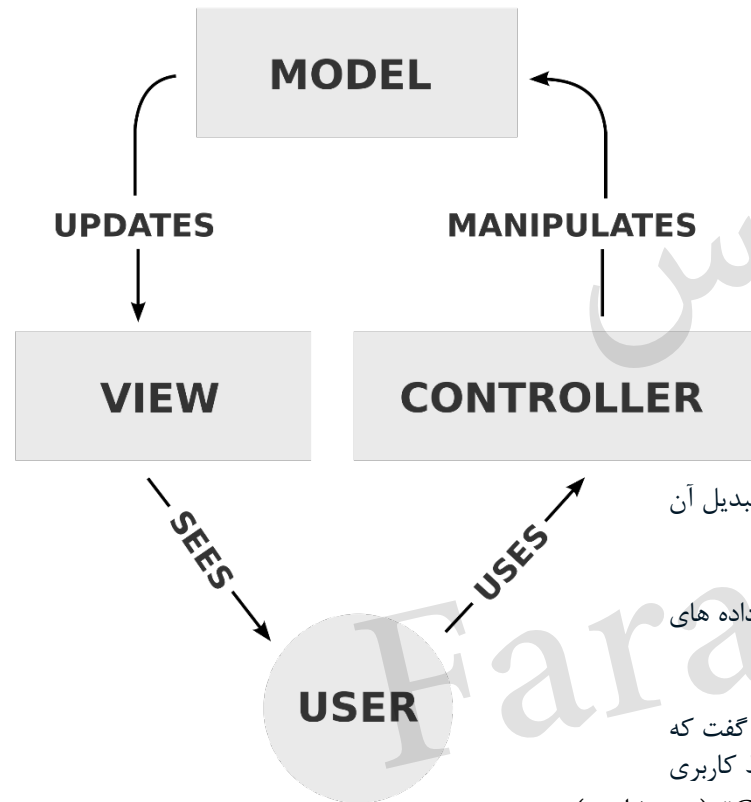
1. Presentation Layer(GUI,UI,User Layer)
2. Business Layer(Business Logic, Business Object)
3. Data Layer(Data Access Layer)

۱- **لایه Presentation(UI):** لایه کاربر در اصل همان برنامه‌هایی است که کاربران با آن ارتباط برقرار می‌کنند تا نیازهای خود را جهت اعمال تغییرات یا گرفتن گزارش به سیستم ارائه دهند. این لایه بطور کلی از قوانین موجود در سیستم مجزا می‌باشد.

۲- **لایه Business(B. L) Business Logic:** وظیفه این لایه انجام پردازش‌های منطقی نرم‌افزار است و در واقع منطق کاری نرم افزار (Business Logic) در آن پیاده‌سازی شده است. این لایه فاقد هرگونه امکان نمایش اطلاعات به کاربر است.

۳- **لایه DAL(DB):** وظیفه این لایه انجام عملیات مرتبط با پایگاه داده نظیر درج کردن(insert)، حذف کردن>Delete)، به روز رسانی(Update) و یا استخراج اطلاعات (Selection/Query) می‌باشد. این لایه نتیجه اجرای فرم‌های کاربر را در پایگاه داده می‌فرستد.

ASP.NET MVC



برخی از برنامه نویسان، همچنان از ASP.NET که بر مبنای فرم های وب و Postback است استفاده می کنند، از ویژگی های MVC سود می برند و برخی دو پلت فرم را با یکدیگر ترکیب می کنند و این موضوع بیانگر آن است که هیچکدام از پلت فرم ها ناقض یکدیگر نیستند. در واقع MVC بر روی معماری های چند لایه جهت تفکیک بخش های مختلف برنامه (بخش های منطقی برنامه مانند داده ها، مجوزها، کنترل صحت داده ها و لایه های مرتبط با کاربر نهایی) قرار می گیرد.

اجزای تشکیل دهنده ی MVC

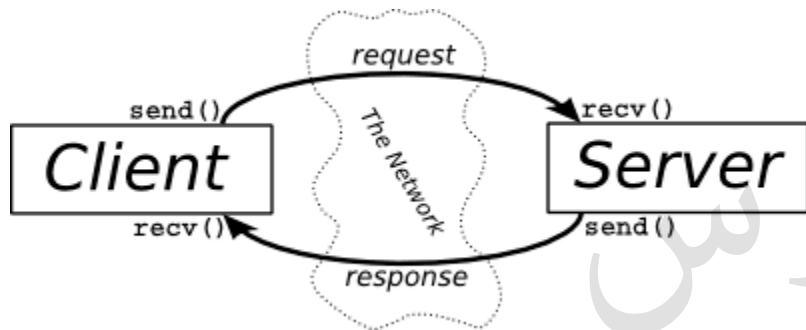
Model-1 (مدل): قسمتی از برنامه کاربردی است که مسئول بازیابی داده از بانک اطلاعاتی، ذخیره آن، تبدیل آن به شی یا آبجکت ها و پیاده سازی منطق برنامه برای داده های دامنه ی مسئله است.

View-2 (نمایشگر): اجزایی از برنامه است که واسط کاربری برنامه (UI) را می سازد. معمولاً این UI از داده های مدل ساخته می شود و در واقع نقطه پایان برنامه کاربردی است.

Controller-3 (کنترلر): اجزایی از برنامه هستند که مدیریت تعامل با کاربر را بر عهده دارند. می توان گفت که واسط بین مدل و نمایشگر می باشند؛ یعنی با مدل کار می کند و در انتها نمایشگری را برای نشان دادن واسط کاربری انتخاب می کند. ورودی کاربر را مدیریت کرده و به آنها پاسخ می دهد و عملیات مورد نیاز کاربر را بر مبنای کدنویسی C# (سی شارپ) انجام می دهد.

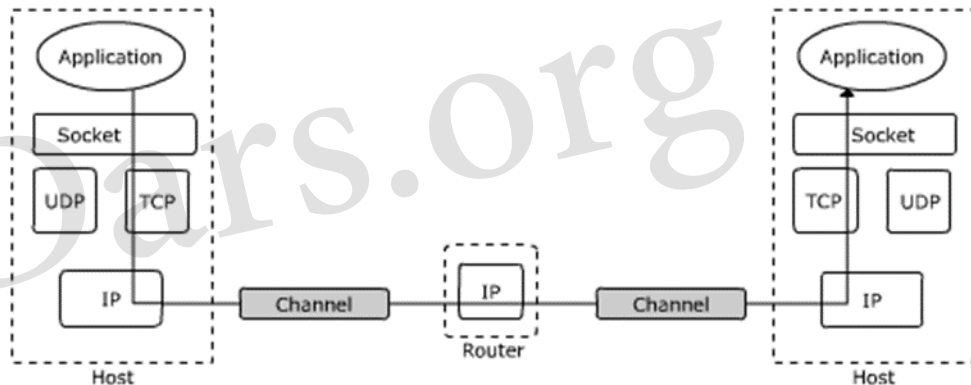


Socket Programming



سوکت یک ارتباط قابل اطمینان جهت انتقال داده ها بین دو host (ماشین مورد نظر) می باشد. سوکت برنامه نویسان را از پیچیده گی های فرآیند برقراری ارتباط بین دو ماشین مانند جزئیات کد کردن بسته ها، فرآیند ارسال داده ها در شبکه، ارسال مجدد بسته های خراب و ... دور می سازد و برنامه نویسان به راحتی قادر به توسعه برنامه های تحت شبکه می باشند.

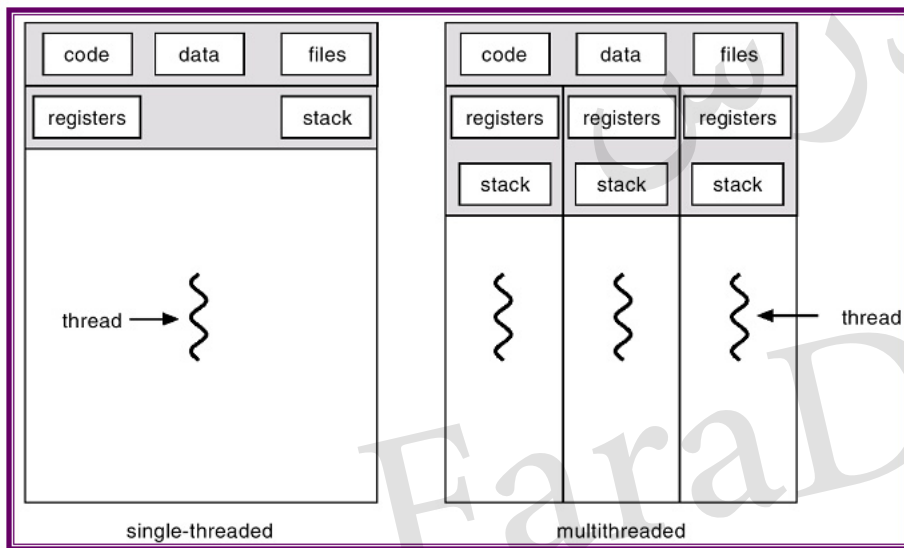
پروتکل TCP/IP و سایر پروتکل ها به صورت لایه ای سازماندهی شده اند. شکل مقابل رابطه بین پروتکل ها، نرم افزارها، و API سوکت ها را در میزبان ها و مسیر یاب ها نشان می دهد. نرم افزار ها از طریق API سوکت ها به سرویس های ارائه شده توسط UDP و TCP دسترسی دارند.



مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

faradars.org/fvcs9311

Multithreaded Programming



تکنیک های چندنخی می تواند پاسخ گویی رابط کاربری برنامه شما را در مقابل کاربر بیشتر کند. در واقع در حالی که یک سری عملیات در پس زمینه اجرا می شوند، رابط کاربری برنامه شما فعال و پاسخ گو باقی می ماند.

در حالی که یک Thread مشغول است پردازشگر می تواند به کار Thread های دیگر بپردازد.

در حالی که پردازش های طولانی در پس زمینه اجرا شده است پردازش های کم مصرف می توانند به راحتی اجرا شوند. پردازش ها در هر لحظه ای از زمان می توانند Pause یا حتی به طور کامل متوقف شوند!

شما می توانید اولویت هر Thread ای را به طور جداگانه با مقدارهای کم و زیاد تنظیم کنید که این کار باعث افزایش کارایی و قدرت برنامه می شود.

© Windows Presentation Foundation

- Hardware Accelerated (Direct 3D)
- Declarative UI with XAML
- Multimedia support
- Skinning support
- xps Documents
- 3D Programming
- Highly Composable
- Animations and Timelines
- Powerful DataBinding
- Resolution Independent
- Vector Based Rendering
- Hardware Accelerated (Direct 3D)

پایه و اساس **WPF** بر **Directx** استوار است. این موضوع سبب می شود که بتوان از بسیاری از جنبه های گرافیکی بدون ایجاد سربار اضافی بر روی برنامه بهره برد و در واقع برنامه با ظاهری بسازید که ساختن آن با برنامه نویسی های پیشین یا غیر ممکن بوده و یا متحمل کار بسیار زیادی بوده است. اگرچه نقطه قوت این تکنولوژی گرافیکی کارایی بیشتر و نویسی با کد کمتر را به بدن معنی نیست که نمی توان با **WPF** اقدام به ایجاد فرم ها و کنترل های سابق نمود.

WCF (Windows Communication Foundation)

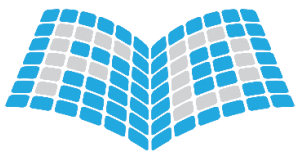
یک قالب کاری برای ساختن برنامه های سرویس گراست. با استفاده از WCF می توانید داده ها را به صورت پیام هایی از یک درگاه سرویس به درگاهی دیگر ارسال کنید. یک درگاه سرویس (service endpoint) می تواند بخشی از سرویس همیشه در دسترس باشد که توسط IIS میزبانی می شود، یا می تواند سرویسی باشد که توسط یک برنامه میزبانی می شود.



زمانی که وب سرویس ها معرفی شدند تا مدتی پاسخگوی نیازهای زمان خودشان بودند، لیکن با رشد تکنولوژیهای نرم افزار نیازها و خواسته های مشتریان نیز بیشتر شد، به طوری که امکانات استاندارد وب سرویسها دیگر کافی نبود. وب سرویسها فاقد امکان تامین امنیت (مانند نیاز به login قبل از استفاده از سرویس، رمزنگاری کانال ارتباطی، رمزنگاری پیام های تبادل شده و ...)، عدم امکان استفاده از Session های سمت سرور هستند.

مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ)

faradars.org/fvcs9311



فرادرس

فرا تراژیک کلاس درس
www.faradars.org

این اسلایدها بر مبنای نکات مطرح شده در مجموعه فرادرس های « آموزش های کاربردی برنامه نویسی C# (سی شارپ) » تهیه شده است.

برای کسب اطلاعات بیشتر در مورد این آموزش به لینک زیر مراجعه نمایید.

faradars.org/fvcs9311