



آموزش کاملاً عملی

LINQ

با اجرای گام به گام
چهار پررژه و بررسی نکات مهم

حسن آقایی

عضو هیئت علمی و دانشکده باهنر شیراز

محمد رحیم نجفی

برنامه نویس و تحلیلگر

صادق هوشیار

تحلیلگر بانکی اطلاعاتی



آموزش کاملاً عملی

LINQ

با اجرای گام به گام
چندین پروژه و بررسی نکات مهم

مولفین

حسن آقایی

محمد رحیم نجفی

صادق هوشیار

طرح جلد

سجاد لدین آقایی

ناشر

فانوس اندیشه

نوبت چاپ

اول ۱۳۹۱

شمارگان

۲۰۰۰

چاپ و صحافی

باقری

قیمت

۷۰۰۰۰ ریال

شابک

۹۷۸-۶۰۰-۵۹۰۱-۲۲-۷

مرکز پخش:

قم - خیابان ارم - روبه روی پاساژ قدس - انتشارات فانوس اندیشه
۰۲۵۱-۷۳۹۳۹۲ - ۷۷۴۳۴۴۵

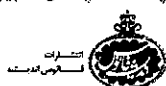
EMAIL: FANOSEANDISHE@YAHOO.COM

تهران - میدان انقلاب - ابتدای خیابان آزادی

خیابان قریب - بعد از خیابان فرصت

پلاک ۷ - پخش کتابیران ۱۵ - ۶۶۵۶۶۵۱۰ - ۰۲۱

پدیدآورنده: آقایی، حسن، ۱۳۶۲ -
عنوان: آموزش کاملاً عملی LINQ IN C# 2010 [لینک این سی شارپ ۲۰۱۰]
تکرار نام پدیدآور: مولفین: حسن آقایی، محمد رحیم نجفی، صادق هوشیار
مشخصات نشر: قم: فانوس اندیشه، ۱۳۹۱
مشخصات فلهاری: ۲۲۲ص:؛ جدول: نمودار
قیمت: ۷۰۰۰۰ ریال
وضعیت فهرست نویسی: فیا
یادداشت: کتابنامه: ص: ۲۲۹
موضوع: سی شارپ (زبان برنامه‌نویسی کامپیوتر) - راهنمای (آموزش (عالمی)
شناسه افزوده: نجفی، محمد رحیم، نویسنده همکار
شناسه افزوده: هوشیار، صادق، نویسنده همکار
رده کنگره: Q۵۷۷۷۳.۱۳۹۱ص:۲۱۷
رده دیویی: ۰۰۵/۱۳۳
شماره کتابخانه ملی: ۳۳۳۷۱۷





سخنی با شما

هرچند که در دنیای کنونی برای دست زدن به هر کاری باید طبق برنامه و هماهنگی شده و در اصطلاح امروز مهندسی شده و مدیریتی عمل کرد تا نتیجه‌ی مطلوب را بدست آورد، بر این باوریم که ما می‌توانیم و قدرت خلاقیت و برنامه ریزی را داریم، اما افسوس که فرهنگ هر کاری را بعد از آموزش علم آن یاد می‌گیریم؛ کاری بس مشکل‌تر از نوع اصولی آن.

اما؛ باید یاد گرفت و یاد داد، باید آموخت و آموزش داد، تا اندیشه‌مان و خاک و طنمان گهربارتر از هر زمان شود، خالی از هر گرایش و منشی. مخصوصاً دانشگاه؛ در محیط سراسر آمیخته به علم، جایی که نامش به تنهایی مایه‌ی بسی افتخار است چه بسا به خدمت در آن.

این آموزه چکیده‌ای از تفکرات، اهداف و نگرش بهینه‌ای است که با همکاری گروهی، خالصانه در جهت آموزش بهتر و بیشتر دانشجویان این مرز و بوم تهیه شده تا فراسوی راهی بس عظیم‌تر بوسیله‌ی جوانانمان باشیم و با ارتباط با بانک‌های اطلاعاتی دنیا بتوانیم مغزهای خلاق را خلاق‌تر و دستان پرتوان را تواناتر بار بیاوریم و دیگر این که به بهانه‌ی دور بودن از ارتباط‌ها و پیشرفت‌های به روز شده‌ی دنیا شانه‌ای خالی نشود.

استاد و دانشجو هر دو، در دو خط موازی قدم برداشته، با فاصله‌ای اندک، چرا که همین دانشجویان، فردا روز با اطلاعات به روز شده‌ی دنیا اساتیدی می‌شوند که نسل دیگر را پرورش می‌دهند. "توسعه و بسط پیشرفت علم برای روزهای خوب‌تر" کاری که در آیات قرآن و احادیث و روایات مختلف از آن بسیار ذکر شده است.

در خاتمه این حاصل؛ استنباط اینجانب از تدریس و پی گیری تحقیق و تفحص های دانشجویانی است که مدتی با آنها در ارتباط بودم، در جهت رفع ایرادها و کاربردی نمودن این تکنولوژی که کاری بس سخت اما شیرین، طولانی اما هموار می باشد با همکاری همفکرانی که مرا یاری نمودند؛ که خداوند در همه ادوار زمان در راه خیر یاریشان کند. قطعاً کتاب دارای نقاط ضعفی است که برخی از آنها از دید ما پوشیده نیست. تذکر در مورد اشکالات کتاب باعث خوشحالی می گردد.

و در اینجا از معاونت محترم پژوهشی دانشکده شهید باهنر شیراز، جناب آقای دکتر به گزین و ریاست محترم بخش کامپیوتر دانشکده، جناب آقای مهندس صادقی، که همکاری صمیمانه ای را در مراحل تالیف کتاب داشتند کمال تشکر را دارم. و از تمامی همکاران و دانشجویان گرامی که در فراهم آوردن این اثر ابراز لطف کردند سپاسگزاری می نمایم. همچنین از مدیریت محترم انتشارات فانوس اندیشه جناب آقای حسینی و همکاران به جهت همکاری و فراهم آوردن تهیات چاپ این کتاب، قدردانی می نمایم.

Linq_group@yahoo.com



۱۳	مقدمه:
۱۵	:Linq (Language Integrated Query)
۱۶	: انواع Linq
۱۷	 فصل اول: عملگرهای LINQ
۱۹	: Linq to object
۱۹	: نوع داده Var
۲۲	: شروع کار با Linq to object
۲۳	: عبارت‌های لامبدا Lambda Expressions
۲۴	: عملگرهای استاندارد پرس و جو
۳۳	: عملگرها و دستورات مربوط به گرامر LINQ
۳۳	: عملگرهای انتخاب:
۴۳	: عملگرهای مرتب سازی:
۴۶	: عملگرهای تولید مجموعه:
۵۵	: عملگرهای تبدیل:
۶۳	: عملگرهای کمیت سنج:
۶۴	: عملگرهای تجمعی:

تمرینات: ۶۷

فصل دوم: کار با بانک اطلاعاتی SQL ۸۳

..... ۸۶ Linq to sql

..... ۸۸ ساخت کلاس های معادل جداول جهت برقراری ارتباط:

..... ۸۸ شکل کلی تعریف کلاس به صورت زیر می باشد:

..... ۹۵ خصوصیات دستور column:

..... ۹۷ Delete Rule

..... ۹۹ Data context:

..... ۱۰۱ کد مربوط به کلاس Leave:

..... ۱۰۶ کد مربوط به کلاس Personnel_Info:

..... ۱۱۳ شکل کلی انجام پرس و جو به صورت زیر می باشد:

..... ۱۱۴ خواندن اطلاعات از پایگاه داده:

..... ۱۱۸ درج اطلاعات در پایگاه داده:

..... ۱۲۱ ویرایش رکورد:

..... ۱۲۵ حذف رکورد:

..... ۱۲۸ ایجاد ارتباط بین کلاس ها:

..... ۱۳۶ بازیابی اطلاعات از کلاس های مرتبط به هم:

..... ۱۴۱ درج، ویرایش و حذف اطلاعات از کلاس های مرتبط:

..... ۱۴۱ درج اطلاعات در کلاس های مرتبط:

..... ۱۴۲ ویرایش اطلاعات کلاس های مرتبط:

..... ۱۴۲ حذف اطلاعات از کلاس های مرتبط:

..... ۱۴۳ مشاهده تغییرات اعمال شده در پایگاه داده:

..... ۱۴۳

۱۴۴.....	متدهای مربوط به اجرای روال‌ها:
۱۴۵.....	کار با متد (Executemethodcall):
۱۵۴.....	کار با متدها:
۱۵۴.....	بازیابی اطلاعات توسط روال‌ها:
۱۵۷.....	ویرایش اطلاعات توسط روال‌ها:
۱۵۹.....	Function ها:
۱۶۰.....	SQLMetal:
۱۶۰.....	پارامترهای SQLMetal:
۱۶۱.....	چگونه یک پروژ را اجرا کنیم:
۱۶۳.....	Linq Designer:
۱۶۳.....	طراح LINQ:
۱۶۹.....	روش آماده استفاده از طراح LINQ:
۱۷۴.....	استفاده از جداول و روال‌ها:
۱۷۶.....	مسئله همزمانی:
۱۸۱.....	بررسی همزمانی توسط متد (submitchanges):
۱۸۲.....	۱- حالت keepChange:
۱۸۵.....	۲- حالت keepcurrentvalue:
۱۸۸.....	۳- حالت Overwritecurrentvalue:
۱۹۰.....	قفل رکورد با transactioncope:
۱۹۱.....	تمرینات:
۱۹۷.....	فصل سوم: LINQ to (Dataset Mysql XML File)
۱۹۹.....	LINQ To Dataset:
۲۰۱.....	LINQ To mysql:

۲۰۲.....	ابزار DBmetal
۲۰۲.....	کار با کلاس ساخته شده:
۲۰۴.....	:LINQ To xml(Extensible Markup Language)
۲۰۵.....	اسناد xml
۲۰۶.....	:XDocument
۲۰۶.....	:XElement
۲۰۷.....	:Xcomment
۲۰۹.....	خواندن اطلاعات بانک و تبدیل به یک سند xml
۲۰۹.....	انجام پرس و جو بر روی سند xml
۲۱۲.....	درج یک عنصر در یک سند:
۲۱۳.....	حذف عنصر از سند:
۲۱۴.....	کار با فایلها:
۲۱۴.....	ساخت، درج اطلاعات در فایل و خواندن اطلاعات:
۲۱۵.....	خواندن اطلاعات فایل و جستجو در فایل:
۲۱۹.....	مقایسه، ادغام و گروه بندی محتویات فایل:
۲۲۲.....	کار با فایل ها و دایرکتوری ها:
۲۲۴.....	خواندن نام و خصوصیات فایل ها:
۲۲۶.....	حجم دایرکتوری و فایل ها:
۲۲۸.....	منابع:



مقدمه

از زمانی که با برنامه‌نویسی آشنا شدیم، اکنون ممکن است واژه شی گرای را زیاد شنیده باشیم و تصور شما چنین باشد که این مبحث یکی از مباحث کارا و در عین حال سخت و پیچیده برنامه‌نویسی است.

شاید این تصور زمانی که این نوع برنامه‌نویسی تازه به وجود آمده بود به خصوص با ظهور ارث بری‌ها و شی گرای‌های پیچیده‌ای چون C++ و زبان‌هایی مانند آن صحیح بود اما امروزه با رشد زبان‌های برنامه‌نویسی این مبحث نه تنها به یکی از مباحث ساده تبدیل شده بلکه یکی از الزامات جهت پیش برد مهندسی و قابل فهم بودن نرم افزارها می‌باشد.

C# زبانی است که از C++ و Java متولد شده است و حاوی بسیاری از جنبه‌های C++ و ویژگی‌های شی گرای جاوا می‌باشد، زبان C# که یک زبان کاملاً شی گرایانه است و پایه و اساس آن بر این اساس گذاشته شده که شما قصد دارید برنامه‌هایتان را شی گرا انجام دهید، همانند C++ ترکیبی از برنامه‌نویسی رویه‌ای (procedural) و همانند جاوا شی گرای ساده می‌باشد. پس بهترین روند برای برنامه نویسان C# تفکر شی گرای است.

با وجود پیشرفت زبانهای برنامه‌نویسی در این زمینه اما برخورد تمام آنها با جداول اطلاعات

(بانکهای اطلاعاتی) کاملاً به دور از شی گزایی بود و با ظهور ADO. Net (ActiveX Data Objects) یک رابط مناسب و کارا و آسان برای کار با بانکهای اطلاعاتی به وجود آمد ولی به دلیل رابطه‌ای بودن اطلاعات در بانک و استفاده از Connection ها و Command ها و روش‌های دیگر هیچ رابطه‌ی شی گزایی به چشم نمی‌خورد به مثال زیر توجه کنید:

```
String strCommand = " Select * From major where code = @ code ";  
SqlCommand objCommand = New SqlCommand(strCommand, " Conection  
string") ;  
objCommand.parameters.Addwithvalue("Code", ID) ,sqlDataAdapter  
objdataadapter = New sqlDataAdapter (objCommand);  
Dataset Objdataset = New dataset () ;  
sqlConnection Connectionstring = New sqlConnection ("Connection String");  
objCommand.ExecuteNonQuery () ;  
objdataadapter.Fill (Objdataset , "major");  
Connection.close () ;
```

تکه کد ۱

به تنها از شی گزایی خبری نیست بلکه یک روند کاملاً پیچیده را باید انجام داد تا رشته‌ای را که کد آن برابر ۱۰ است را برگردانیم باید مراحل زیر را یکی یکی انجام دهیم ابتدا یک شی از نوع Command تعریف می‌کنیم و آنرا با Connection ارتباط می‌دهیم سپس مقدار را به Command ارسال می‌کنیم ، با تعریف یک Dataset و برقراری ارتباط با بانک مقادیری که بازگردانده شده را درون Dataset بریزیم.

پس مشکل زبان‌های شی گزایی در ارتباط با بانک اطلاعاتی در مثال قبل تا حدی بررسی شد. برای ارتباط با هر نوع منبع نیاز به اجرای یک سری قواعد و قوانین متفاوت داریم پس به این نتیجه می‌رسیم که تمام تلاش‌های مایکروسافت تا قبل از سال ۲۰۰۵ با ارائه متدهایی چون Dataset و ADO.Net برای رفع این مشکل (ارتباط رابطه‌ای) کاملاً موفق نبوده و

:Linq (Language Integrated Query)

اما در سال ۲۰۰۵ توسعه دهندگان حرفه‌ای موفق شدند یک روش جدید را ارائه دهند که توسط این روش نه تنها می‌توان یک روش کاملاً شی گرایانه را دنبال کرد بلکه می‌توان هر منبع داده را در Net. با یک روش یکسان بازیابی کرد البته لازم به ذکر است که Linq را می‌توان برای بازیابی اطلاعات از هر شی که واسط IEnumerate را پیاده سازی کرده باشد استفاده کرد که این واسط در ادامه بررسی خواهد شد. به کد زیر توجه کنید:

```
Major m = new Major ();
m.code=10;
m.name="Comp1";
Major.insert (m);
```

تکه کد ۲

حال این قطعه کد را با کد قبلی مقایسه کنید:

```
Var q = From _Major in Major _
Where _Major.code == 10
Select _Major;
```

تکه کد ۳

نه تنها نیاز به دنبال کردن مراحل سخت قبلی نیست بلکه یک روش ساده و کاملاً شی گرا می‌باشد در تکه کد اول یک رشته با کد ۱۰ و نام Comp1 تعریف می‌کند و در کد دوم از بین تمام رشته‌ها رشته‌ای که کد آن برابر ۱۰ باشد را برمی‌گرداند، در ضمن نوع منبع داده را در این گرامر از هر نوعی باشد می‌توان بازیابی کرد یعنی در این قطعه کد هیچ تفاوتی ندارد که Major یک table از بانک اطلاعاتی (هر نوع بانک اطلاعاتی) یا یک لیست و... باشد پس بهتر است نوید این را به شما بدهیم که از این به بعد کافی است به این فکر کنید که چه می‌خواهید و دیگر نیاز نیست به تفکر درباره‌ی چگونگی آن پردازید.

انواع Linq :

با توجه به نوع منبع داده‌هایی که وجود دارد Linq به ۳ دسته اصلی تقسیم می‌شود که در این کتاب برآنیم که این انواع را شرح دهیم:

۱. Linq to object

۲. Linq to SQL

۳. کار با منابع دیگر

www.ketab.ir