



فهرست مطالب مجموعه آموزشی

Asp.Net Core MVC with C# 6.0

ماژول ۱: معماری

- ✓ مقدمه
- ✓ دات نت فریم ورک و اجزای آن
 - مزایای دات نت فریم ورک
 - اجزای دات نت فرم ورک
 - روند اجرای برنامه های دات نت
 - تاریخچه دات نت فریم ورک و ویژوال استودیو
- ✓ مقدمه ای بر ویژوال استودیو ۲۰۱۵
 - دانلود و نصب ویژوال استودیو
 - بررسی اجمالی محیط ویژوال استودیو
 - ✓ اولین برنامه به زبان سی شارپ
 - ایجاد پروژه
 - کامپایل و اجرای پروژه
 - پنجره Error List و خطاهای پروژه

ماژول ۲: مقدمات زبان برنامه نویسی سی شارپ

- ✓ آماده سازی پروژه
 - بررسی فایل sln
 - ✓ کار با کلاس Console
 - نوشتن در پنجره کنسول
 - خواندن از پنجره کنسول
 - Trace کردن برنامه
 - ✓ کار با ویرایشگر ویژوال استودیو
 - پنجره Options





- تغییر ظاهر ویرایشگر
- تنظیم نمایش / عدم نمایش لیست اعضای اشیاء در زمان کد نویسی
- نحوه استفاده از Code Snippet ها در زبان سی شارپ
 - مشاهده محتویات یک فایل Code Snippet
- ✓ توضیحات در زبان سی شارپ
 - توضیحات تک خطی
 - توضیحات چند خطی
 - کاربرد Xml Document ها
- ✓ متغیرها
 - نحوه تعریف یک متغیر و مقدار دهی آن
 - استفاده از کلمه کلیدی var
 - محدوده متغیرها
 - متغیرهایی با مقدار ثابت (استفاده از کلمه کلیدی const)
- ✓ انواع داده
 - بررسی روند ذخیره سازی انواع مقداری و ارجاعی
 - انواع مقداری
 - انواع عددی در زبان سی شارپ
 - اعداد صحیح
 - اعداد اعشاری
 - کاراکترها در زبان سی شارپ
 - نوع داده Boolean
 - انواع تبدیل
 - تبدیل ضمنی
 - تبدیل صریح
 - کاربرد عملگرهای checked و unchecked
 - نوع داده عدد بزرگ (BigInteger)
 - انواع ارجاعی
 - کار با رشته ها (کلاس String)
 - الحاق رشته ها





- جایگزینی در رشته ها (Replace)
- Trim کردن رشته ها
- زیر رشته ها (SubString)
- فرمت بندی رشته ها
- عملگر Interpolation عملگر (\$))
 - کار با کلاس StringBuilder
 - بررسی اجمالی کلاس Object
- ✓ شرط ها
 - دستور if
 - شرط if ساده
 - شرط if همراه با یک else
 - شرط های if – else تو در تو
 - دستور switch
 - ✓ حلقه های تکرار
 - حلقه for
 - حلقه while
 - حلقه do-while
 - حلقه in – foreach
 - ✓ کار با تاریخ و زمان
 - ایجاد یک DateTime
 - بررسی خواص یک DateTime
 - تغییر بر روی یک شی DateTime با استفاده از شی TimeSpan
 - تغییر بر روی یک شی DateTime با استفاده از متد های AddXXX
 - مقایسه دو شی DateTime
 - فرمت بندی خروجی DateTime
 - کار با کلاس PersianCalendar
 - تبدیل تاریخ و زمان فعلی سیستم به معادل شمسی
 - ✓ مدیریت استثنا ها – بخش اول
 - بلاک try





▪ بلاک catch

▪ catch های عمومی و خصوصی

▪ بلاک finally

✓ ایجاد یک برنامه ساده (پژوه ماشین حساب)

ماژول ۳: شی گرایی - بخش اول

✓ آماده سازی پروژه

▪ بررسی فایل sln

✓ استفاده از کلاس ها

▪ تعریف کلاس

▪ روش نمونه گیری از کلاس (Instance)

▪ روش دستیابی به اعضای یک کلاس

✓ فیلد ها و خواص در کلاس

▪ تعریف فیلد ها

• مقدار اولیه فیلد ها در کلاس

• استفاده از Access Modifier (کنترل سطح دسترسی اعضای کلاس)

▪ private

▪ public

▪ تعریف Property ها

• خواص خواندنی / نوشتنی

• خواص خواندنی

• خواص خودکار

• خواص خواندنی در خواص خودکار

✓ متد ها

▪ تعریف متد در کلاس

▪ فراخوانی متد ها

▪ سربار گذاری متد ها

▪ فراخوانی با نام آرگومان ها

▪ پارامتر های اختیاری در متد ها

▪ فراخوانی با ارجاع (استفاده از کلمه کلیدی ref)





■ خلاصه نویسی بدنه یک متد (Expression Body)

✓ کانستراکتورها

- کانستراکتور پیش فرض
- کانستراکتور های دارای آرگومان
- نمونه گیری از کلاس با کانستراکتور های مختلف
- فراخوانی کانستراکتور ها توسط یکدیگر
- فیلد های readonly در کلاس و مقدار دهی آن ها
- ✓ کلاس ها و اعضای استاتیک
 - تعریف یک عضو استاتیک
 - فراخوانی اعضای استاتیک
 - کار با کلاس Math (به عنوان مثالی در زمینه کلاس ها و اعضای استاتیک)

✓ انواع شمارشی

- تعریف یک نوع شمارشی
 - تعریف متغیر از انواع شمارشی
 - استفاده از انواع شمارشی توسط مقادیر عددی آن ها
- استفاده از کلاس Enum و Parse کردن یک نوع شمارشی
- استفاده از انواع شمارشی در دستور switch
- استفاده از code Snippet جهت تعریف دستور switch همراه با انواع شمارشی

✓ ساختارها

- تعریف یک ساختار
- فیلد ها در ساختار ها
- کانستراکتورها در ساختار ها
 - نمونه گیری از ساختار ها
- تعریف متد در ساختار
- سربرگزاری عملگرها در یک ساختار
 - سربرگزاری عملگر جمع (+)
 - سربرگزاری عملگر تفریق (-)

✓ انواع بی نام

- تعریف یک متغیر از نوع بی نام





- استفاده از متغیر های تعریف شده از نوع بی نام
 - انتساب دو متغیر از نوع بی نام به یکدیگر
 - ✓ کلاس ها و اعضای جزئی (partial)
 - تعریف کلاس partial
 - تعریف متد partial
 - ✓ متدهای الحاقی
 - تعریف یک متد الحاقی برای شی DateTime
 - استفاده از متد الحاقی

ماژول ۴: شی گرایی - بخش دوم

- ✓ آماده سازی پروژه
- ✓ ارث بری در زبان سی شارپ
 - تعریف کلاس پایه (پدر) و کلاس های فرزند
 - نمونه گیری از کلاس های فرزند و استفاده از اعضای کلاس پدر
 - ✓ اعضای virtual
 - تعریف یک عضو virtual در کلاس پایه (پدر)
 - دوباره نویسی اعضای virtual در کلاس های فرزند
 - دوباره نویسی متد ToString از کلاس Object
 - ✓ کلاس ها و اعضای abstract
 - تفاوت اعضای virtual و abstract
 - تعریف یک عضو abstract در کلاس پدر
 - دوباره نویسی اعضای abstract در کلاس های فرزند
 - چرا اعضای abstract می بایستی درون کلاس های abstract قرار بگیرند؟
 - ✓ کلاس ها و متد های sealed
 - تعریف کلاس sealed و خصوصیات آن
 - متد های sealed (چرا و چگونه)
 - ✓ کانستراکتورها در ارث بری
 - بررسی کانستراکتورها در زمان ارث بری
 - فراخوانی کانستراکتور کلاس پدر توسط کانستراکتور کلاس های فرزند





✓ اینترفیس ها

- تعریف یک اینترفیس
- پیاده سازی اینترفیس
- استفاده از کلاس های دارای اینترفیس
- چرا اینترفیس ها؟
- پیاده سازی صریح اینترفیس ها
- ✓ آرایه ها در سی شارپ

- تعریف یک آرایه
- آرایه های تک بعدی
- تعریف، مقدار دهی و استفاده
- آرایه های چند بعدی
- تعریف، مقداردهی و استفاده
- آرایه های دنداندار
- تعریف مقدار دهی و استفاده

✓ کلاس Array

- ایجاد یک آرایه با استفاده از کلاس Array
- خواندن و نوشتن در آرایه
- مرتب سازی داده های آرایه
- معکوس کردن داده های آرایه
- محدودیت های آرایه ها

✓ کار با کلاس ArrayList

- نمونه گیری از کلاس ArrayList
- خواندن و نوشتن در یک ArrayList
- مشکلات کلاس ArrayList
- عملیات Boxing و Unboxings
- عدم تشخیص نوع داده ها در زمان کمپایل

✓ ژنریک ها

- کلاس های ژنریک
- تعریف یک کلاس ژنریک





- نمونه گیری از کلاس های ژنریک

- متد های ژنریک

- تعریف یک متد ژنریک

- استفاده از متد های ژنریک

- اینترفیس های ژنریک

- تعریف و پیاده سازی

- اعمال محدودیت ها بری نوع داده ای کلاس و متد های ژنریک

- ✓ مجموعه های ژنریک

- تعریف و استفاده از کلاس `List<T>`

- خواندن و نوشتن داده ها در لیست

- مرتب سازی داده های لیست

- تعریف و استفاده از لیستی از کلاس ها (لیستی از کلاس `Person`)

- اینترفیس `IComparable<T>` و کاربرد آن

- اینترفیس `IComparer<T>` و کاربرد آن

- دلیگیت `Comparision<T>` و کاربرد آن

- کار با کلاس `Dictionary`

- تعریف و استفاده از دیکشنری با انواع اولیه

- تعریف یک دیکشنری با یک شی (کلاس `Person`)

- دوباره نویسی متد `Equals` (چرا و چگونه)

- دوباره نویسی متد `GetHashCode` (چرا و چگونه)

- `Delegate` ها

- تعریف یک `Delegate`

- نمونه گیری از یک `Delegate`

- نمونه ای از کاربرد `Delegate` ها

- عملگرهای `+=` و `--` در `Delegate` ها

- ✓ متد های بی نام

- تعریف صریح متد های بی نام

- تعریف و استفاده با استفاده از کلمه کلیدی `delegate`

- ✓ عبارات لامبدا





- استفاده از عملگر لامبدا (\Rightarrow)
- متدهای چند پارامتری و عبارات لامبدا
- ✓ کلاس های `Func<T>` و `Action<T>`
- تعریف و استفاده
- ✓ رویداد ها
- خصوصیات `Delegate` ای که برای رویداد ها به کار می روند
- تعریف رویداد
- انتشار رویداد
- انتظار برای رخ دادن رویداد(استفاده از رویداد)
- آرگومان های سفارشی در رویداد ها
- استفاده از کلاس `EventHandler<T>` در تعریف رویداد ها
- ✓ مدیریت استثنا ها – بخش دوم
- استفاده از کلمه کلیدی `throw` برای ایجاد(پرتاب) یک استثنا
- کلاس `Exception` و برخی از خواص آن
- تعریف و استفاده از یک کلاس `Exception` سفارشی

ماژول ۵: Language Integrated Query

- ✓ آماده سازی پروژه
- ✓ مقدمه و آماده سازی داده ها
- ✓ کار با عملگر های استاندارد کوئری در LINQ
- عملگر `from – in`
- عملگر `select`
- عملگر `where`
- تعریف و استفاده از یک کوئری
- عملگر `let`
- عملگر `group – by – in`
- ✓ کار با عملگر های بر پایه متد در LINQ
- متد `Where`
- متد `OrderBy`





- OrderByDescending متد
- ThenBy متد
- ThenByDescending متد
- Skip متد
- SkipWhile متد
- Take متد
- TakeWhile متد
- Average و Sum ، Count متد های
- اینترفیس <T>IEqualityComparer و کاربرد آن در متد Distinct

ماژول ۶: Input – Output

- ✓ آماده سازی پروژه
- ✓ مروری سریع بر فضای نام System.IO
- ✓ کار با دایرکتوری ها، فایل ها و درایو ها
- کار با کلاس Directory و اعضای آن
- کار با کلاس DirectoryInfo و اعضای آن
- کار با کلاس File و FileInof
- کار با کلاس DriveInfo
- ✓ خواندن (نوشتن) از (در) فایل های متنی
- کار با متد های کلاس File
 - متد ReadAllText
 - متد ReadAllLines
 - متد WriteAllText
 - متد WriteAllLines
- کار با کلاس StreamReader
- استفاده از عملگر using جهت dispose کردن اشیاء
- کار با کلاس StreamWriter
- کلاس های StringReader و StringWriter





- ✓ سریال کردن داده ها
- کار با کلاس BinaryFormatter
- کار با کلاس XmlSerializer
- ✓ کار با فایل های XML
- تعریف ی فایل XML
- کار با کلاس XmlReader
- کار با کلاس XmlWriter
- کار با کلاس XmlDocument
- دستورات LINQ برای فایل های XML
- استفاده از کلاس XDocument

ماژول ۷: کتابخانه کلاس های اختصاصی

- ✓ آماده سازی پروژه
- ✓ کار با قالب پروژه های Class Library
- ایجاد یک کتابخانه کلاس
- ایجاد ارجاع به یک کتابخانه کلاس
- استفاده از اشیای یک کتابخانه کلاس

ماژول ۸: مقدمه بر ASP.Net Core MVC

- ✓ مروری بر برنامه نویسی تحت وب
- برنامه نویسی Windows Forms Application
- برنامه نویسی ASP.Net Web Forms
- برنامه نویسی ASP.Net MVC
- ✓ برنامه نویسی سمت کلاینت
- ✓ آماده سازی سیستم و ایجاد اولین برنامه
- نصب .Net Core
- ایجاد اولین پروژه
- بررسی اجمالی روند اجرایی پروژه
- ✓ ایجاد یک نمونه پروژه دیگر





- فرخوانی View ها
- استفاده از ViewBag جهت ارسال داده ها از کنترلر به ویو
- TagHelper ها
- ViewModel ها
- درخواست های HttpGet و HttpPost
- ✓ مروری بر تکنولوژی MVC
- اجزای MVC
 - Model
 - View
 - Controller
- ViewModel ها و DomainModel ها
- روند درخواست و پاسخ در MVC

ماژول ۹: ایجاد پروژه CodeWayBlog – خروجی نهایی و ایجاد پروژه

- ✓ مشاهده خروجی
- ✓ ایجاد پروژه
- مروری بر ساختار پروژه
 - فایل های json
 - فایل global.json
 - فایل lanuchSettings.json
 - فایل project.json
 - کلاس Program و عملکرد آن
 - کلاس Startup و عملکرد آن
- ایجاد یک کنترلر

ماژول ۱۰: ایجاد پروژه CodeWayBlog – مدل داده ای

- ✓ ایجاد مدل داده ای
- ایجاد پوشه Models
 - ایجاد کلاس Post
 - ایجاد کلاس Link





- ایجاد کلاس BlogSetting
- ایجاد کلاس BlogRepository
- تولید داده های آزمایشی
- نمایش لیست پست ها در خروجی
- ✓ استفاده از Dependency Injection
- تعریف اینترفیس IBlogRepository
- تعریف و استفاده از کلاس BlogFactory
- استفاده از سرویس ها در کلاس Startup و ثبت اینترفیس IBlogRepository
- متد AddScoped
- متد AddSingleton
- متد AddTransient
- استفاده از اینترفیس IBlogRepository در کانستراکتور کنترلر Home
- بررسی روند اجرایی سرویس های (DI)
- تغییر روت پیش فرض

ماژول ۱۱: ایجاد پروژه CodeWayBlog – افزودن ویو به پروژه

- ✓ افزودن ویو به پروژه
- پوشه Views و روال جستجوی ویو ها در MVC
- افزودن ویوی Index به پروژه
- ارسال لیست پست ها از کنترلر به ویو
- مقید سازی نوع داده Post به ویوی Index و نمایش لیست پست ها
- استفاده از Layout_ها در پروژه
- کاربرد _ViewStart

ماژول ۱۲: ایجاد پروژه CodeWayBlog – افزودن کتابخانه Bootstrap

- ✓ افزودن کتابخانه Bootstrap
- نصب ابزار git
- استفاده از Bower و نصب پکیج های :
- Bootstrap
- jquery





- jquery-validation
- jquery-validation-unobtrusive
- بررسی اجمالی کتابخانه Bootstrap
- تغییر در Layout ها
- _LayoutBase
- _UserLayout
- _AdminLayout
- لینک کردن پروژه به Bootstrap
- تولید ظاهر خروجی پروژه
- واکنشگرا (Responsive) بودن خروجی پروژه

ماژول ۱۳: ایجاد پروژه CodeWayBlog – افزودن PartialViews

✓ افزودن PartialViews

- افزودن ویوی جزئی Post_
- مقید سازی نوع Post به ویوی Post_
- فراخوانی Post_ در ویوی Index

ماژول ۱۴: ایجاد پروژه CodeWayBlog – افزودن پایگاه داده

✓ افزودن پایگاه داده

- افزودن پکیج های مربوط به EntityFrameworkCore از نیوگت
- ایجاد کلاس BlogDbContext
- ایجاد DbSet ها برای Post ، link و BlogSetting
- افزودن فایل appSettings.json
- تعریف ConnectionString جهت اتصال به پایگاه داده
- استفاده از IConfigurationRoot و ConfigurationBuilder جهت کار با فایل appSettings.Json
- معرفی کلاس BlogDbContext به سرویس های پروژه توسط AddDbContext
- استفاده از Migration جهت ایجاد پایگاه داده
- دستور Add-Migration
- دستور Update-Database





- ایجاد کلاس BlogSeedData جهت مقدار دهی اولیه پایگاه داده
- استفاده از کلاس BlogDbContext در کلاس BlogRepository
- نمایش لیست پست های درون پایگاه داده در خروجی

ماژول ۱۵: ایجاد پروژه CodeWayBlog – افزودن View Component ها

✓ افزودن View Component ها

- تعریف کلاس BlogTitleViewComponent و ارث بری از کلاس ViewComponent
 - متد Invoke و عملکرد آن در ViewComponent ها
 - استفاده از IBlogRepository در BlogTitleViewComponent
 - فراخوانی ViewComponent ها در ویو
- تکمیل کلاس BlogSeedData
- تعریف سایر کلاس های ViewComponent
- کلاس LinkViewComponent
 - ایجاد خروجی بر اساس ویو
- نحوه جستجوی ویو های مربوط به ViewComponent ها

ماژول ۱۶: ایجاد پروژه CodeWayBlog – TagHelper ها

✓ افزودن TagHelper ها

- افزودن پکیج های مربوطه از نیوگت به پروژه
- TagHelper های پیش فرض
 - ویوی ViewImport_ و کاربردهای آن
 - معرفی TagHelper های پیش فرض
 - استفاده از دایرکتیو @addTagHelper
 - استفاده از HtmlHelper جهت تولید لینک ها
 - استفاده از TagHelper های :
 - asp-controller
 - asp-action
 - asp-route-
 - TagHelper ها سفارشی
 - صفحه بندی لیست پست ها





• TagHelper سفارشی جهت تولید لینک های صفحه بندی پست ها

- تعریف کلاس PageLinkTagHelper و ارث بری از کلاس TagHelper
- تغییر ظاهر لینک های به شیوه Bootstrap

ماژول ۱۷: ایجاد پروژه CodeWayBlog – افزودن بخش مدیریت وبلاگ

✓ افزودن بخش مدیریت وبلاگ

- ایجاد پوشه های Admin و Areas
- استفاده از صفت Area برای معرفی کنترلر های یک Area
- تعریف روت مربوط به Area ها
- ویو های بخش ادمین و AdminLayout_
- دستور @RenderSection و کاربرد آن در پروژه

ماژول ۱۸: ایجاد پروژه CodeWayBlog – مدیریت پست ها

✓ مدیریت پست ها

- تعریف کنترلر Post
- تعریف متد List
- تعریف ویوی List و نمایش لیست پست ها
- افزودن متد Add و ویوی مربوطه جهت درج اطلاعات
- افزودن متد های Edit و ویوی مربوطه جهت ویرایش اطلاعات
- حذف اطلاعات
- تست برنامه

ماژول ۱۹: ایجاد پروژه CodeWayBlog – افزودن بخش ورود به برنامه توسط کتابخانه

Identity

✓ افزودن بخش ورود به برنامه توسط کتابخانه Identity

- افزودن پکیج های مربوطه
- تغییر کلاس پایه ی کلاس BlogDbContext به کلاس IdentityDbContext
- افزودن سرویس Identity به لیست سرویس های پروژه
- کلاس IdentitySeedData و افزودن داده های اولیه مربوط به کاربر





- بروز رسانی پایگاه داده
- محدودیت دسترسی به اکشن متد ها با صفت Authorize
- تعریف کنترلر AccountController
- تعریف ویوی Login و فرم ورود به برنامه
- تعریف کلاس LoginViewModel به عنوان مدل داده ای ویوی Login
- کلاس های UserManager<T> و SignInManager<T> و کاربرد های آن
- محدودیت دسترسی به کنترلر ها با صفت Authorize
- صفت AllowAnonynouse و کاربرد آن

ماژول ۲۰: ایجاد پروژه CodeWayBlog – اعتبار سنجی داده ها

✓ اعتبار سنجی داده ها

- بررسی داده های کاربر به صورت دستی و توسط دستورات زبان سی شارپ
- ModelState و کاربرد آن
- استفاده از asp-validation-summary جهت نمایش پیام های اعتبار سنجی به کاربر
- استفاده از asp-validation-for جهت نمایش پیام های اعتبار سنجی به کاربر(برای هر فیلد)
- استفاده از صفات مربوط به اعتبار سنجی
- اعتبار سنجی سمت کاربر و روش پیاده سازی آن

ماژول ۲۱: ایجاد پروژه CodeWayBlog – مدیریت خطا ها در وب

✓ مدیریت خطاها در وب

- میان افزار UseDeveloperExceptionPage
- مدیریت خطا ها بر اساس محیط کاری
- تعریف اکشن متد ErrorHandler و تنظیم روت مربوط به آن
- میان افزار UseStatusCodePages و کاربرد آن

