

Domain Driven Design



Visual Studio

مدت آموزش: ۵ جلسه (در مجموع ۲۰ ساعت)
مدرس: علیرضا ارومند



سرفصل‌های دوره آموزشی پیشرفته Domain Driven Design

۱- DDD چیست؟

DDD به طور مختصر روشی جهت توسعه نرم‌افزار است که توسط آقای Eric Evans در سال ۲۰۰۲ معرفی شد و از آن زمان به بعد در دنیای توسعه نرم‌افزار مورد استفاده بود. اما در سال‌های اخیر با پیشرفت‌هایی که در حوزه ابزارهای توسعه نرم‌افزار اتفاق افتاده و در کنار آن کاربرد هرچه بیشتر نرم‌افزار و پیچیده شدن آن‌ها استفاده از این روش بیش از پیش مورد توجه توسعه‌دهندگان قرار داده است. مطالبی که در این بخش بررسی می‌کنیم عبارتند از:

- آشنایی با فلسفه DDD
- آشنایی با مشکلات و پیچیدگی‌های توسعه نرم‌افزار
- چگونه DDD به مدیریت پیچیدگی‌ها کمک می‌کند؟
- آیا فقط روشی جهت پیاده‌سازی داریم یا برای سایر قسمت‌های توسعه نیز راهنمایی داریم؟
- اشتباهات رایج در مورد DDD

۲- نحوه برخورد با مسئله

حل مسئله‌های بزرگ و پیچیده بدون تجزیه و تحلیل صحیح آن امری است غیر ممکن. و تجزیه و تحلیل درست به دست نمی‌آید مگر با جلسات و تعاملات بسیار زیاد با افرادی که به محدوده کاری مسلط هستند و سالها با این حوزه کاری آشنا هستند. چگونگی تشکیل جلسات و نحوه تعامل و نحوه ایجاد مدل صحیح برای توسعه از مواردی هستند که در این قسمت به آن‌ها خواهیم پرداخت.



- چرا نیاز به تجزیه مسئله داریم؟
- چگونه به یک دانش و زبان مشترک در مورد صورت مسئله برسیم؟
- Domain Expert کیست و نقش او در توسعه نرم‌افزار چیست؟
- بررسی روش‌های بهینه جهت جمع‌آوری اطلاعات

۳- چگونه بر روی مسئله اصلی تمرکز کنیم؟

در توسعه یک نرم‌افزار بزرگ و پیچیده با قسمت‌های مختلف و صورت مسئله‌های متفاوت فنی و کاربردی سر و کار داریم. اما اینکه کدام یک از این مسائل برای ما مسئله اصلی است و کدام یک فرع بسیار مهم است. با دانستن این موضوع می‌توانیم به عنوان یک مدیر و پیشرو سرمایه‌های مالی و انسانی خود را به بهترین شکل ممکن برای رسیدن به هدف هدایت کنیم.

- چرا باید مسئله بزرگ را به زیر مسئله‌ها خورد کنیم؟
- چگونه مسئله اصلی را از بین مجموعه‌ای از زیر مسئله‌ها تعیین کنیم؟
- چگونه منابع خود را به مسئله‌های مختلف تخصیص دهیم؟
- دامنه اصلی برنامه و دامنه پشتیبان چیست؟
- آیا باید همه قسمت‌های برنامه را به بهترین شکل ممکن طراحی و توسعه دهیم؟

۴- الگوهای پیاده‌سازی دامنه

هدف اصلی در DDD مدیریت پیچیدگی‌ها است. یکی از پیچیدگی‌هایی که در هنگام توسعه نرم‌افزار با آن سر و کار داریم مربوط به نحوه پیاده‌سازی دامنه برنامه است. باید به روشی دامنه برنامه را توسعه دهیم که برای نیازهای آتی قابل توسعه باشد و به سادگی بتوان آن را در آینده توسعه داد. الگوهای مختلفی برای پیاده‌سازی دامنه برنامه وجود دارد که در این قسمت به بررسی این الگوها می‌پردازیم.

- Rich Domain Model
- Anemic Domain Model
- Transaction Script و مزایا و معایب آن
- Table Module چیست و چه زمانی کاربرد دارد؟
- Active Record

۵- Bonded Context چیست؟

در یک نرم‌افزار بسیار بزرگ این احتمال وجود دارد که نیاز باشد نرم‌افزار را به قسمت‌های مختلفی تقسیم کنیم و هر قسمت را به طور جداگانه طراحی و پیاده‌سازی کنیم. جدا بودن این قسمت‌ها می‌تواند در لایه کد یا دیتابیس و حتی تیم و زبان برنامه‌نویسی توسعه باشد. با این حال در انتها تمام این قسمت‌ها باید بتوانند با یکدیگر ارائه خدمات داشته باشند تا به یک نرم‌افزار جامع برسیم. چگونگی انجام این کار و چالش‌های پیش رو مسئله‌ای است که در این قسمت آن را بررسی می‌کنیم.

- یک مدل بسیار بزرگ پیچیدگی زیادی دارد
- چگونگی اشتراک یک مدل بزرگ بین تیم‌ها و مشکلات آن
- تفاوت واژه‌ها در بین تیم‌های مختلف و ایرادات آن
- چگونگی کار با کدهای قدیمی در یک مدل بسیار بزرگ
- چگونه برنامه خود را به قسمت‌های کوچک و منسجم تقسیم کنیم؟
- تعیین محدوده قسمت‌ها برنامه و جلوگیری از تاثیر متقابل روی هم
- Subdomain چیست؟
- Bonded Context چیست؟
- تفاوت SD و BC
- چگونگی برقراری ارتباط بین قسمت‌های مختلف
- Shared kernel چیست؟

- آشنایی با Anticorruption Layer

۶- آشنایی با معماری و لایه بندی

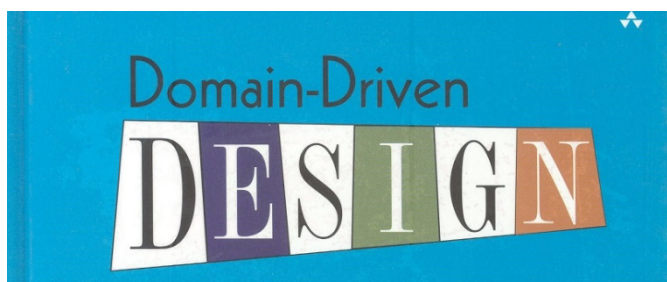
Application هنگام توسعه به روش DDD راهکارهایی به ما داده می شود که پیچیدگی های تکنولوژی را از پیچیدگی های واقعی پروژه تشخیص دهیم و به درستی پیاده سازی کنیم اما برای هر پروژه ای باید معماری و لایه بندی مناسبی داشته باشیم تا بتوانیم در آینده به خوبی پروژه را توسعه دهیم. در این قسمت با معماری پروژه و چگونگی انجام آن آشنا می شویم.

- آشنایی کلی با معماری پروژه
- چگونه یک معماری خوب منجر به جداسازی پیچیدگی های تکنولوژی از دامنه برنامه می شود؟
- طراحی و پیاده سازی یک پروژه چند لایه
- معکوس سازی وابستگی و نقش آن در لایه بندی برنامه
- لایه دامنه و نحوه پیاده سازی آن
- Application Layer Service چیست و چه کاربردی دارد؟
- Infrastructure چیست و چه قسمت هایی در آن پیاده سازی می شوند؟
- نحوه برقراری ارتباط بین لایه های مختلف

۷- آشنایی با Value Object ها:

هنگام توسعه نرم افزار بعضا اشیایی داریم که مقدار آن ها معرف ماهیت آن ها است و با تغییر هر یک از خواص آن دیگر نمی توانیم بگوییم با همان شی قبلی سر و کار داریم. به زبان دیگر تنها راه شناسایی این اشیا مقدار آن ها است. به این اشیا اصلاحات Value Object می گوئیم و در این قسمت به بررسی این دسته از اشیا می پردازیم.

- Value object چیست؟
- نقش Value Object ها در DDD
- آشنایی با ویژگی های Value Object ها
- نحوه پیاده سازی Value Object ها
- ذخیره و بازیابی مقادیر به کمک NoSql ها



۸- Entity ها و نحوه پیاده سازی آنها:

هنگامی که در حال تجزیه و تحلیل برنامه های خود هستیم بعضا با کلاس ها و اشیایی مواجه می شویم که در برنامه با یک مشخصه خاص قابل شناسایی هستند. مثل یک مشتری خاص در یک برنامه فروشگاه. اینجا با مفهوم Entity سر و کار داریم. و در ادامه به بررسی باید ها و نبایدهای آن و چگونگی پیاده سازی آن می پردازیم.

- Entity چیست؟
- چگونه Entity ها را در برنامه خود تشخیص دهیم؟
- تفاوت Entity ها با Value Object ها
- الگوی State و کاربرد آن در Entity ها
- الگوی Memento و کاربرد آن در توسعه Entity ها

۹- آشنایی با Domain Service ها:

زمانی که در حال توسعه یک سیستم بزرگ هستیم بعضا با مفاهیمی سر و کار داریم که مربوط به یک Entity یا Aggregate خاص نیست و نمی‌توان با هیچ یک از این موارد به تنهایی به پیاده‌سازی راه حل پرداخت و مسئله را حل کرد. در این جا به سراغ Domain Service ها می‌رویم. اما نکته‌ای که وجود دارد کاربرد فراوان کلمه Service در توسعه نرم‌افزار است و مشکلاتی که کاربرد این کلمه در مفاهیم مختلف ایجاد می‌کند. در ادامه با هم بررسی خواهیم کرد که:

- Domain Service چیست؟
- بررسی انواع سرویس و تشخیص تفاوت‌ها آن‌ها
- چه زمانی از Domain Service استفاده می‌کنیم؟
- استفاده از Domain Event ها
- نحوه ارتباط Entity ها و Domain Service ها
- الگوهای پرکاربرد برای توسعه سرویس‌ها

۱۰- Domain Event چیست؟

هنگامی که با DDD سر و کار داریم به مرور متوجه می‌شویم در صورتی که با رخدادهای سیستم به خوبی آشنا شویم و راه حل بهینه برای مدیریت این رخدادها داشته باشیم نرم‌افزار بهتری خواهیم داشت. رخدادهایی که در دامنه برنامه ما اتفاق می‌افتند را اصلاحات Domain Event می‌نامیم. یکی از دستاوردهایی که در تعامل با Domain Expert ها خواهیم داشت به دست آوردن اطلاعات مناسب از همین رخدادها خواهد بود.

- آشنایی با Domain Event
- Domain Event ها چه وظیفه ای به عهده دارند
- نقش Domain Event ها در Domain
- نقش Domain Event در Application
- پیاده سازی کمک رخدادها در دات نت
- In memory Bus
- استفاده از loc Container برای پیاده سازی Domain Event

۱۱- Aggregate و کاربردهای آن:

هنگامی که در حال توسعه نرم افزار هستیم متوجه خواهیم شد که فقط Entity ها و Value Object ها نیستند که برای ما اهمیت دارند و می توانند مسئله ما را حل کنند. بلکه بعضا روابط بین اشیا و پیچیدگی هایی که این روابط ایجاد می کنند هم برای ما اهمیت پیدا می کند. این روابط برای ما به کمک Aggregate ها طراحی و پیاده سازی می شود.

- چگونگی برخورد با پیچیدگی هنگام کار با روابط بین اشیا
- چرا Aggregate ها برای ما اهمیت دارند؟
- Aggregate root چیست؟
- آشنایی با وظایف Aggregate root
- بررسی انواع روابط در Aggregate ها
- نحوه ذخیره و بازیابی Aggregate ها

۱۲- Factory ها و Repository ها و کاربرد آن ها در DDD

هنگامی که در حال توسعه نرم افزار هستیم با Entity ها Value Object ها و Aggregate ها سر و کار داریم و به مرور هر چه برنامه ما بزرگتر شود تعداد و در هم تنیدگی این اجزا نیز بیشتر می شود. و مسلما به مرور یکی از صورت مسئله هایی که با آن مواجه می شویم ارائه راه حل مناسب جهت نمونه سازی از این اجزا می باشد.

- چگونه نمونه سازی از استفاده جدا می شود؟
- استفاده از الگوی Factory Method برای ایجاد نمونه جدید
- استفاده از Factory Method جهت باز سازی نمونه
- چه زمانی از Factory استفاده می کنیم؟
- Repository چیست و نقش آن در برنامه ما چیست؟
- بررسی اشتباهات رایج در مورد Repository
- Domain Model چیست و چه تفاوتی با Data Model دارد؟
- آشنایی با فریم ورک های ذخیره سازی اطلاعات

۱۳- Event Sourcing

نگهداری تاریخچه تغییرات از نیازهایی است که همیشه در نرم افزارها مسئله اصلی بوده. نیازهای سخت افزاری و پیچیدگی های نرم افزار برای پیاده سازی یک تاریخچه خوب می تواند موجب شکست یا موفقیت یک پروژه بزرگ شود. یکی از روش هایی که جهت نگهداری این تاریخچه مورد استفاده قرار می گیرد Event Sourcing است که در ادامه به بررسی جزئیات آن می پردازیم.

- آشنایی با Event Sourcing
- چه مسائلی را برای ما حل می کند؟
- چگونه یک Event Store داشته باشیم؟
- استفاده از SQL و NoSQL ها برای نگهداری تاریخچه
- بررسی رابطه Event Sourcing با CQRS

مدرس دوره علیرضا ارومند کیست

مدرس، معمار و مشاور ارشد نرم افزار

علیرضا ارومند به عنوان Technical Manager شرکت داتین (وابسته به فناپ) در حوزه پروژههای بانکی



فعال است. او همچنین مدرس و Technical Manager پروژههای نیک آموز می باشد از دیگر تخصص های او میتوان به: تولید فریمورک برنامه نویسی فوق العاده حرفه ای با مدیریت بیش از ۱ میلیون تراکنش در ثانیه، همکاری با تیم توسعه شرکت ارتباط فردا (بانک آینده)، مشاور فنی شرکت توسعه رفاه پردیس (بانک رفاه)، مدیر فنی خبرگزاری نسیم، سخنران تنها همایش مورد تایید مایکروسافت در خاورمیانه در حوزه ASP.NET Core ، مدیر فنی خبرگزاری بین المللی پیام کوتاه نسیم (برنده جشنواره وب ایران)، مدرس دوره های ASP.NET ، Dot Net در نیک آموز، همکاری با تیم توسعه شرکت ارتباط فردا



دوره در یک نگاه کلی:

- عنوان دوره:** دوره آموزش غیرحضوری Domain Driven Design
- مخاطبین:** برنامه نویسانی که حداقل ۲ سال سابقه برنامه نویسی دارند.
- پیش نیاز دوره:** دوره آموزش کوئری نویسی در SQL Server
- طول دوره:** ۵ جلسه در مجموع ۲۰ ساعت
- پشتیبانی:** دارای گروه تلگرامی اختصاصی است
- نحوه ارائه:** غیرحضوری
- نحوه ارسال:** ارسال با پیک یا پست
- مدیریت دوره:** فرید طاهری مدیریت نیک آموز
- مدرس:** علیرضا ارومند

نحوه دریافت دوره‌های ارسالی چگونه است؟



ارسال به تهران: طی ۲۴ ساعت کاری پس از ثبت سفارش، هماهنگی لازم جهت ارسال بسته با پیک توسط تیم پشتیبانی با شما انجام خواهد شد.

ارسال به خارج از تهران: طی ۲۴ ساعت کاری پس از ثبت سفارش، بسته ارسالی با آدرس ثبت شده در پنل کاربری، به اداره پست تحویل داده می‌شود. زمان ارسال و تحویل بسته به شما، منوط به زمان‌بندی اداره پست بوده و از ۲ الی ۵ روز کاری متغیر خواهد بود.

صدور فاکتور رسمی چگونه است؟



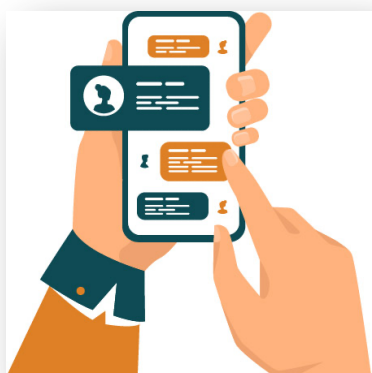
در صورت تمایل به دریافت فاکتور رسمی، پیش از خرید خود با واحد فروش مجموعه (۰۲۱ - ۹۱ ۰۷ ۰۰ ۱۷) تماس حاصل نمایید. شایان ذکر است، امکان صدور فاکتور رسمی پس از خرید آنلاین از سایت مجموعه به هیچ عنوان وجود نخواهد داشت.

پشتیبانی بی نظیر

آموزش بدون پشتیبانی کاملاً بی‌معنی است، الان تمام دوره‌های نیک آموز دارای پشتیبانی از طریق سایت و تلگرام است.

پس ثبت و نهایی شدن سفارش شما در سایت نیک‌آموز، تیم پشتیبانی طی ۲۴ تا ۷۲ ساعت کاری با شما تماس خواهند گرفت تا فرایند عضو شدن شما در گروه پشتیبانی تلگرامی هر دوره انجام شود.

در صورت وجود هر گونه سوال و ابهامی می‌توانید با شماره‌های شرکت تماس حاصل فرمایید و یا از طریق بخش چت پشتیبانی سایت، سوالات خود را مطرح نمایید.



آدرس: تهران، یوسف آباد، میدان فرهنگ، خیابان ۳۳، پلاک ۲۹، زنگ ۲، دفتر نیک آموز

شماره تماس: ۰۲۱ - ۹۱ ۰۷ ۰۰ ۱۷ | **موبایل فروش:** ۰۹۱ ۰۴۰۰۶ ۲۰۶