Найти в открытых источниках что такое доменно-ориентированный подход в разработке ПО (как он называется в оригинале, в чём его суть, про структуру какой части решения идёт речь?)

Domain-Driven Design

Осуществляется с помощью инструментов и шаблонов стратегического моделирования, таких как: единый язык, предметная область, предметная подобласть, смысловое ядро, ограниченный контекст, карта контекстов.

Проектирование с помощью тактических шаблонов, таких как, например, сущность, объект, значение, репозиторий, событие, агрегат.

Основной целью применения DDD является получение высококачественной модели программного обеспечения, которая будет максимально точно отражать поставленные бизнес-цели. Для реализации этого требуется объединение усилий как разработчиков, так и экспертов в предметной области.

**Свойства:**

Единый язык (Ubiquitous Language)

Ограниченный контекст (Bounded context)

Предметная область (Domain), предметная подобласть (Subdomain), смысловое ядро (Core domain)

Пространство задач и пространство решений

Карта контекстов

DDD – способ построение больших систем на основе супер-простых элементарных компонент.

DDD различает ограниченные контексты, домены и поддомены. Домены — это пространства проблем, которые вы хотите устранить. Это области, в которых объединяются знания, поведение, законы и действия. Семантическая связь между компонентами или службами отображается в доменах, зависимостях поведения. Другой аспект доменов — обмен данными. Участники команды должны использовать язык, который понимают все члены команды, чтобы каждый мог работать эффективно. Этот общий язык называется вездесущим языком или языком домена.

Домены разложены в поддомены, чтобы лучше управлять сложностью. Типичным примером этого является разложение домена на поддомены, которые соответствуют одной конкретной бизнес-проблеме.

**Структура:**

Сущность (Entity)

Если какое-то понятие предметной области является уникальным и отличным от всех других объектов в системе, то для его моделирования используется сущность. Такие объекты-сущности могут сильно отличаться своей формой за весь цикл существования, тем не менее их всегда можно однозначно идентифицировать и найти по запросу. Для этого используются уникальные идентификаторы, создание которых необходимо продумать в первую очередь при проектировании сущности.

Объект-Значение (Value Object)

Если для объекта не важна индивидуальность, если он полностью определяется своими атрибутами, его следует считать объектом-значением. Чтобы выяснить, является ли какое-то понятие значением, необходимо выяснить, обладает ли оно большинством из следующих характеристик:

* Оно измеряет, оценивает или описывает объект предметной области;
* Его можно считать неизменяемым;
* Оно моделирует нечто концептуально целостное, объединяя связанные атрибуты в одно целое;
* При изменении способа измерения или описания его можно полностью заменить;
* Его можно сравнивать с другими объектами с помощью отношения равенства значений;
* Оно предоставляет связанным с ним объектам функцию без побочных эффектов.

Служба Предметной Области (Domain Service)

Используя единый язык, существительные этого языка отражаются в объекты, а глаголы отражаются в поведения этих объектов. Очень часто существуют глаголы или какие-то действия, которые нельзя отнести к какой-то сущности или к какому-то объекту-значению. Если существует такого рода операция в предметной области, ее объявляют как служба предметной области (она отличается от прикладной службы, которая является клиентом). Есть три характеристики служб:

* Операция, выполняемая службой, относится к концепции предметной области, которая не принадлежит ни одной из существующих сущностей;
* Операция выполняется над различными объектами модели предметной области;
* Операция не имеет состояния.

Агрегат (Aggregate)

Агрегат является самым сложным из всех тактических инструментов DDD. Агрегатом называется кластер из объектов сущностей или значений. То есть эти объекты рассматриваются как единое целое с точки зрения изменения данных.

Хранилища (Repository)

Хранилищем называется область памяти, которая предназначена для безопасного хранения помещенных в нее элементов. Именно этим является предметно-ориентированное хранилище. Хранилище используется для агрегатов. Помещая агрегат в соответствующее хранилище, а затем извлекая его оттуда, вы получаете целостный объект. Если агрегат будет изменен, то изменения будут сохранены. Если агрегат будет удален, то его уже нельзя будет извлечь.