Предметно – ориентированное программирование (DDD)

Часть первая, введение в DDD

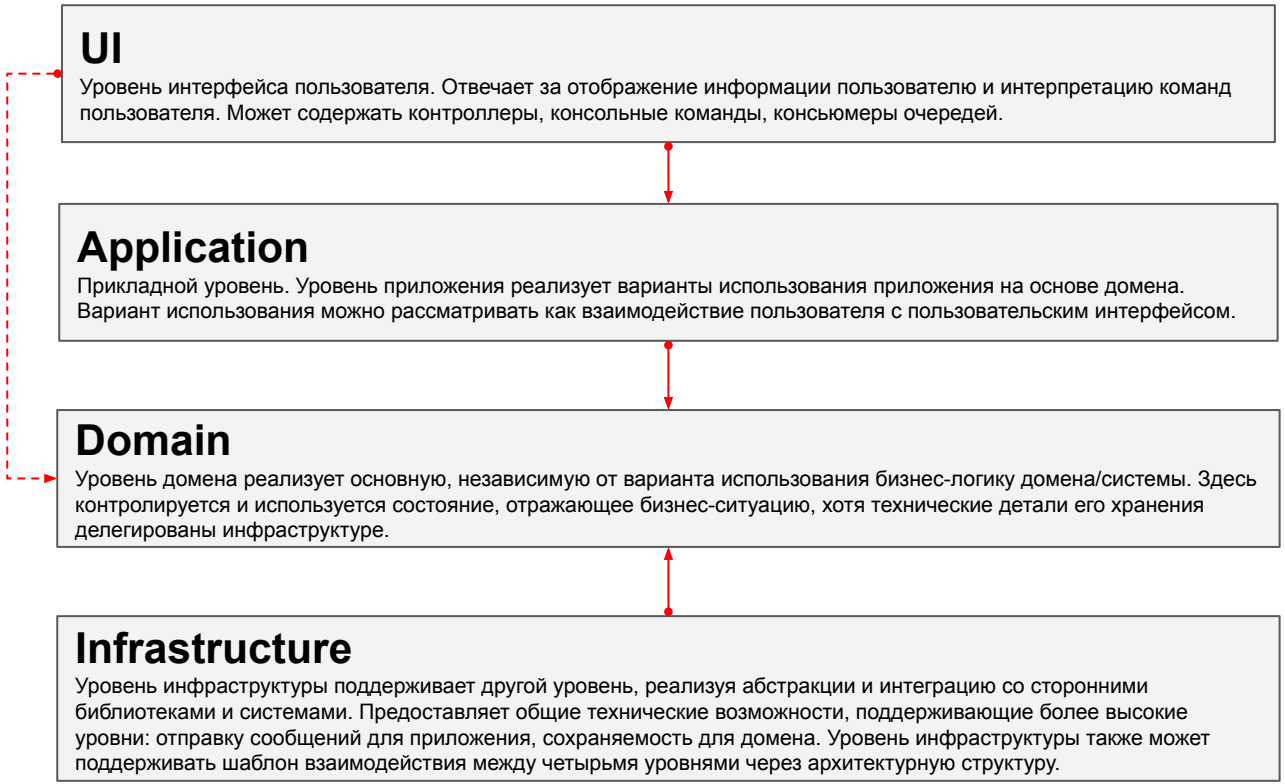
Концепция

Предметно ориентированное программирование - это набор принципов и схем, направленных на создание оптимальных систем объектов. Сводится к созданию программных абстракций, которые называются моделями предметных областей. В эти модели входит бизнес-логика, устанавливающая связь между реальными условиями области применения продукта и кодом. Данный подход позволяет значительно ускорить процесс проектирования программного обеспечения в незнакомой предметной области, повысить процент покрытия тестами, выработать системный подход в проектированию сервисов.

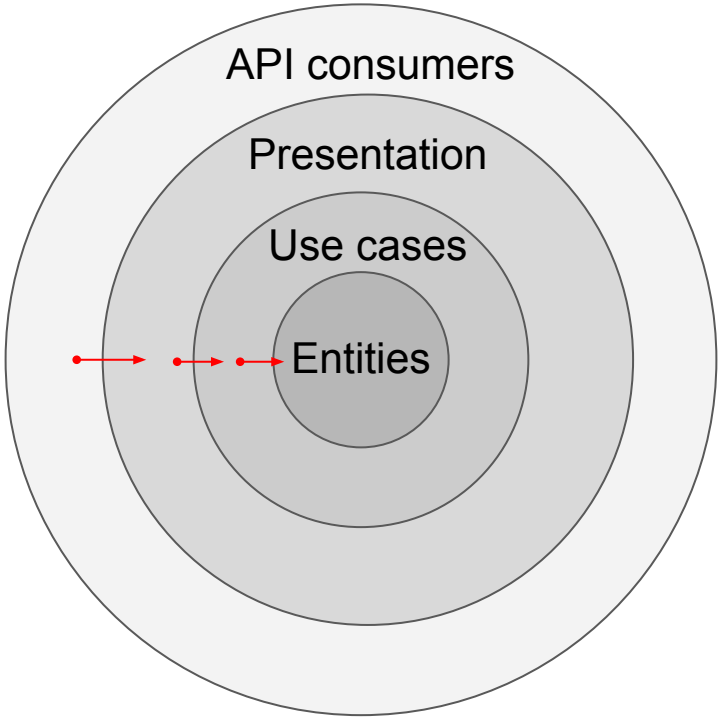
Задачи

1. Высокое качество ПО, точно реализующее бизнес задачу
2. Единый язык (Ubiquitous Language)
3. Ограниченный контекст (Bounded Context)
4. Высокий процент покрытия тестами (Code coverage)

Слои


UI
Уровень интерфейса пользователя. Отвечает за отображение информации пользователю и интерпретацию команд пользователя. Может содержать контроллеры, консольные команды, консьюмеры очередей.
Application
Прикладной уровень. Уровень приложения реализует варианты использования приложения на основе домена. Вариант использования можно рассматривать как взаимодействие пользователя с пользовательским интерфейсом.
Domain
Уровень домена реализует основную, независимую от варианта использования бизнес-логику домена/системы. Здесь контролируется и используется состояние, отражающее бизнес-ситуацию, хотя технические детали его хранения делегированы инфраструктуре.
Infrastructure
Уровень инфраструктуры поддерживает другой уровень, реализуя абстракции и интеграцию со сторонними библиотеками и системами. Предоставляет общие технические возможности, поддерживающие более высокие уровни: отправку сообщений для приложения, сохраняемость для домена. Уровень инфраструктуры также может поддерживать шаблон взаимодействия между четырьмя уровнями через архитектурную структуру.

Контекст



Понятия

1. **Единый язык.** Для того чтобы уравнять разработчиков и экспертов предметной области, чтобы было гораздо проще обмениваться полезными знаниями о предметной области, подход DDD предлагает применять общий набор терминов, понятий и фраз, который будет использоваться в общении между членами команды, и который позже отразится в исходном коде результирующей программы.
2. **Ограниченный контекст.** Это явная граница, внутри которой существует модель предметной области, которая отображает единый язык в модель программного обеспечения.
3. **Предметная область.** Это то, что делает организация, и среда, в которой она это делает. Разработчик программного обеспечения для организации обязательно работает в ее предметной области. Следует понимать, что при разработке модели предметной области необходимо сосредоточиться в определенной подобласти, так как практически невозможно создать единственную, всеобъемлющую модель бизнеса даже умеренно сложной организации. Очень важно разделять модели на логические разделенные предметные подобласти (Subdomain) всей предметной области, согласно их фактической функциональности. Подобласти позволяют быстрее определить разные части предметной области, необходимые для решения конкретной задачи.
4. **Entity.** Сущность являющаяся уникальной в предметной области, отличительной особенностью которой является уникальный идентификатор.
5. **Value Object.** В отличие от entity является не уникальным, сравнение производится по уникальному набору атрибутов.
6. **Domain Service.** Операции, выполняемые службой, относится к концепции предметной области, которые не принадлежат ни одной из существующих сущностей, выполняющиеся над различными объектами модели предметной области и не имеющие состояния.
7. **Module.** Модули внутри модели являются именованными контейнерами для некоторой группы объектов предметной области, тесно связанных друг с другом. Их цель – ослабление связей между классами, которые находятся в различных модулях. Так как модули в подходе DDD – это неформальные или обобщенные разделы, их следует правильно называть. Выбор их имен является функцией единого языка.
8. **Aggregate.** Это кластер из объектов сущностей или значений. То есть эти объекты рассматриваются как единое целое с точки зрения изменения данных. У каждого агрегата есть корень Aggregate Root и граница, внутри которой всегда должны быть удовлетворены инварианты.

Материалы

1. Книги и видео доклады Эрика Эванса
2. Доклад “Грамотное ООП” Дмитрий Елисеев
3. Интернет ресурсы Habr/Medium по теме DDD