

ドメイン駆動設計と イミュータブルな設計

2021年11月24日

有限会社システム設計 増田



『ドメイン駆動設計』のエッセンス

第3章 モデル駆動設計（モデルと実装を一致させる）

第10章 しなやかな設計（変更を楽で安全にする設計パターン）

第15章 蒸留（コアドメインに集中する）

クラス設計の方針の違い



できれば
イミュータブルにする

できるだけ
イミュータブルにする

かならず
イミュータブルにする

クラス設計の方針の違い



できれば
イミュータブルにする

値オブジェクト
副作用のない関数
閉じた操作
宣言的な設計スタイル

できるだけ
イミュータブルにする

ドメイン・プリミティブ
部分的不変エンティティ
エンティティスナップショット
エンティティリレー

かならず
イミュータブルにする

値オブジェクト
コレクションオブジェクト
区分オブジェクト
口座・履歴パターン

イミュータブルなクラス的设计パターン

閉じた操作

withメソッド (setterの代替)

イベントリポジトリ・集約ファクトリ

閉じた操作

操作(メソッド)の引数の型と返す値の型が、そのクラスの型に閉じる

String#concat(String other) : String

BigDecimal#add(BigDecimal other) : BigDecimal

Money#add(Money other) : Money

Quantity#add(Quantity other) : Quantity

Set#union(Set other) : Set

Map#merge(Map other) : Map

イミュータブルな値オブジェクトの基本パターン

withメソッドパターン

イミュータブルなクラスを設計する時のsetterの代替パターン
不変なオブジェクトを元に別インスタンスを作成する

`LocalDate#withMonth(int month) : LocalDate`

`LocalDate#withYear(int year) : LocalDate`

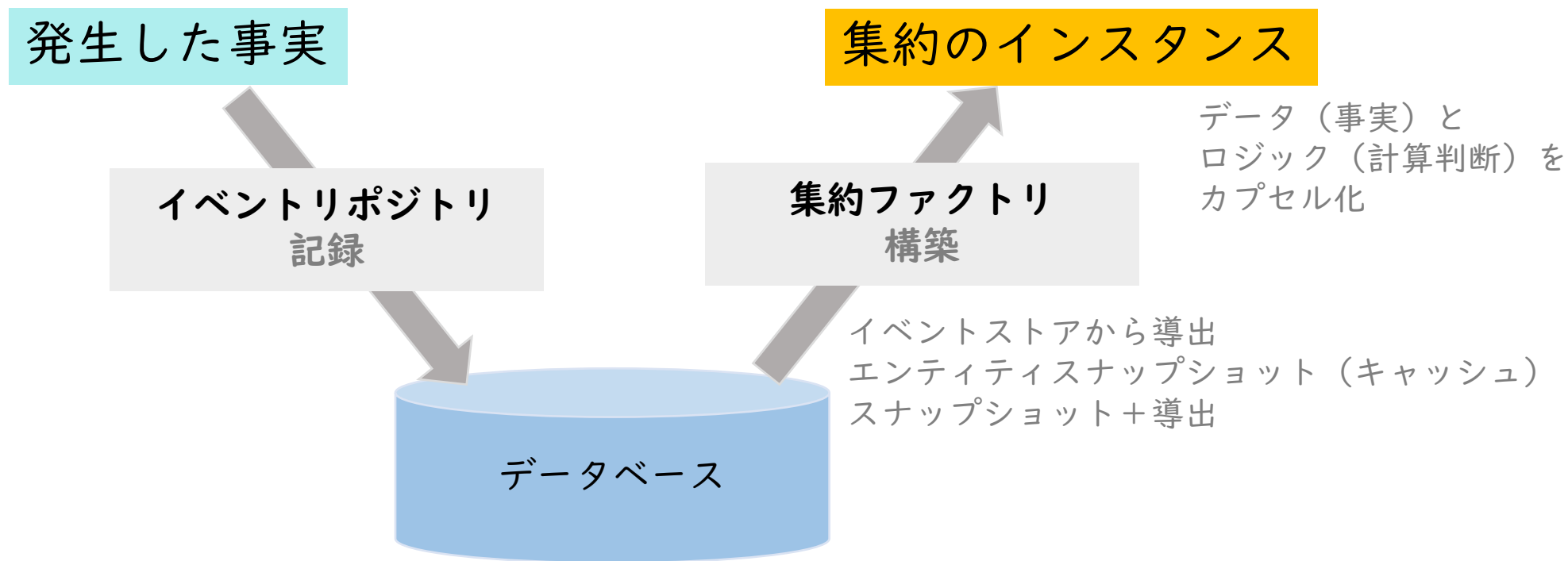
`SalesOrder#add(ItemLine itemLine) : SalesOrder` //別インスタンス

イミュータブルな集約への応用例（『ドメイン駆動設計』10章）

`SharePie#比例配分with(long 配分する総額) : SharePie` //新たな配分構成

イベントリポジトリ・集約ファクトリ

目的ごとに異なる集約



事実の記録と集約の構築を非対称にする（集約を永続化しない）

集約を変更する時は、まず事実を記録してから集約インスタンスを生成する