

# Aufgabe 1.1

Mittwoch, 13. Oktober 2021 19:43

Pattern = Mustertlösungen für wiederkehrende Probleme

1. Layered Architecture + Modules + Anti-Corruption-Layer
2. Services
3. Entities
4. Object-Values
5. Repositories
6. Aggregates
7. Factories

Bounded Context:

- Einsatzbereich eines Domänenmodells
- Begrenzter Bereich (Grenze der Gültigkeit eines Modells)

- sollten in Domain Modell sein

4. Identifizieren sich über ihre Werte - kein eigener lifecycle
3. Repräsentieren die Objekte eines Bounded context's model
  - eigenen Lebenszyklus
  - konstante Identität
6. Gruppierung an Constraints. Root Entity ist die Stelle um Zugang zum Objektgraphen zu bekommen und lifecycle - Objekte, die Aggregate nicht miteinander koppeln. Logisch zusammengehören

Frei von Spring

1. LA: Unterteilung eines Programms in Schichten
  - jede Schicht eigenes Design
  - von darunterliegenden Schichten abhängig
  - gesamte Code auf eine Schicht beziehen

Module:

- fachliche Schichtung
- strukturieren Bounded Context da Sachverhalte getrennt werden

Anti-Corruption Layer:

- dient als Wrapper für die Logik

- isoliert Bounded Context voneinander
  - dient zur Übersetzung der Anforderungen von Subsystem an Subsystem
5. übernimmt das Lesen, Schreiben, ... von Objekten in einer Datenquelle und das Mapping

7. Wenn Aggregate-Objekt kompliziert wird oder von interner Struktur zu viel preisgibt, kann diese Funktionalität in einer Factory-Klasse gekapselt werden.

2. Wird benötigt wenn Funktionalität kein Entity, V-O oder Aggregate zugeordnet werden kann

Domain Events:

- für Änderungshistorie

- für das Informieren verteilter Bausteine