JAVACREAM

Training
Consulting
Projectmanagement

Spring Grundlagen

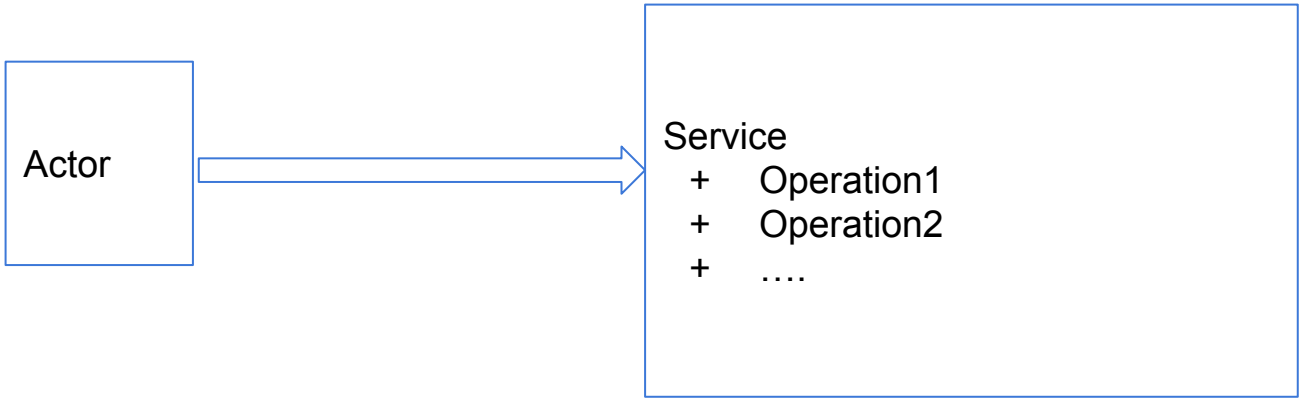
Kurze Vorstellung

JAVACREAM
Training
Consulting
Projectmanagement

- Name
- Rolle im Unternehmen
- Themenbezogene Vorkenntnisse
- Aktuelle Problemstellung
- Konkrete individuelle Zielsetzung

Ausgangssituation

Fachanwendung



Bei Modellierung einer Fachanwendung keinerlei Bezug zu Spring vorhanden

Technisches Modell

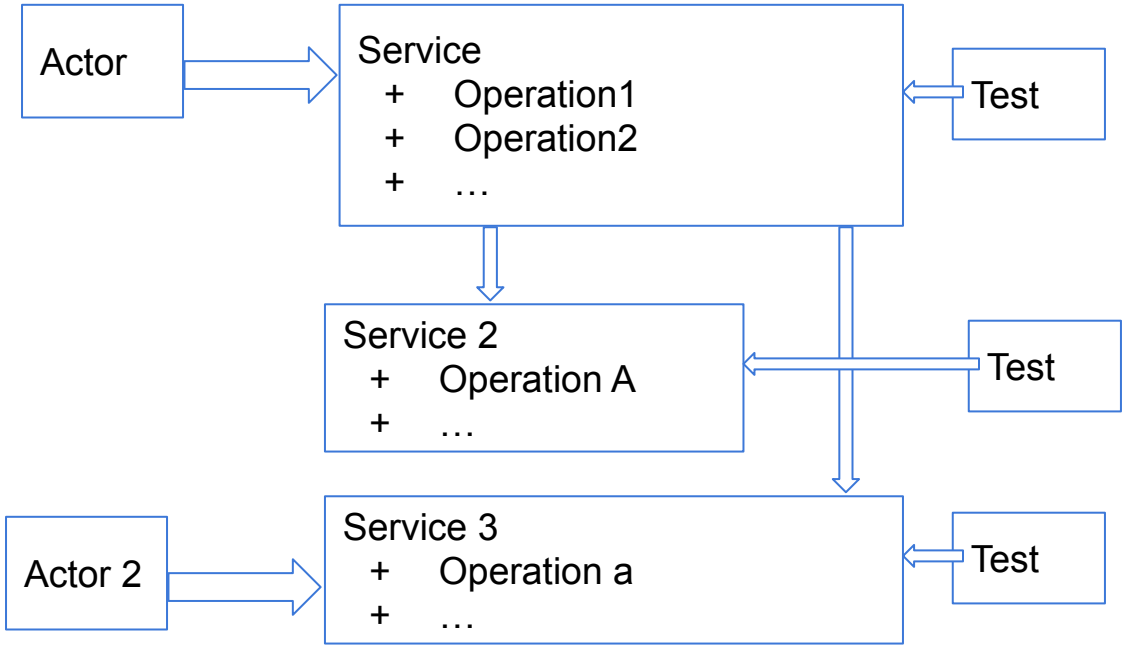
Anforderungen an das Modell

- \+ Wartbarkeit
- \+ Wiederverwendung
- \+ Testbarkeit

Umsetzung durch Modularisierung statt einer monolithischen Applikation

Bezug zu Spring ist indirekt

- \+ Bei Verwendung von Spring ist die Modularisierung eines technisches Modells sehr gut möglich

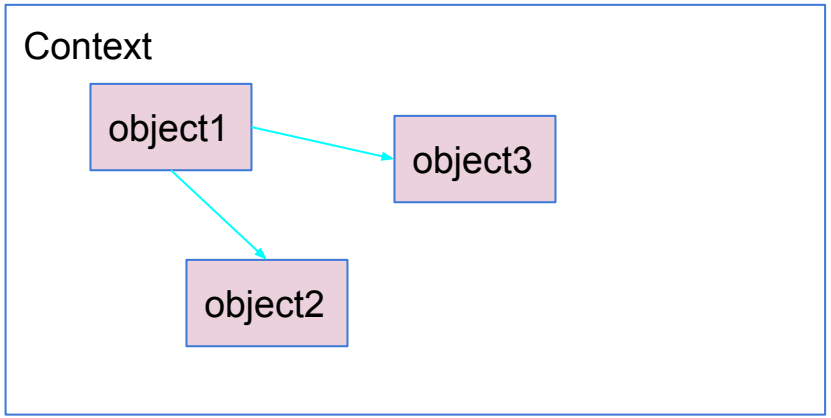


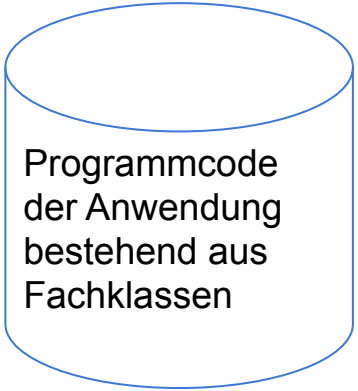
Context & Dependency Injection (CDI)

CDI baut aus den einzelnen Modulen das Objektgeflecht der Anwendung auf

Spring ist eine Umsetzung des Design Patterns Context & Dependency Injection

“Spring ist ein CDI-Framework”



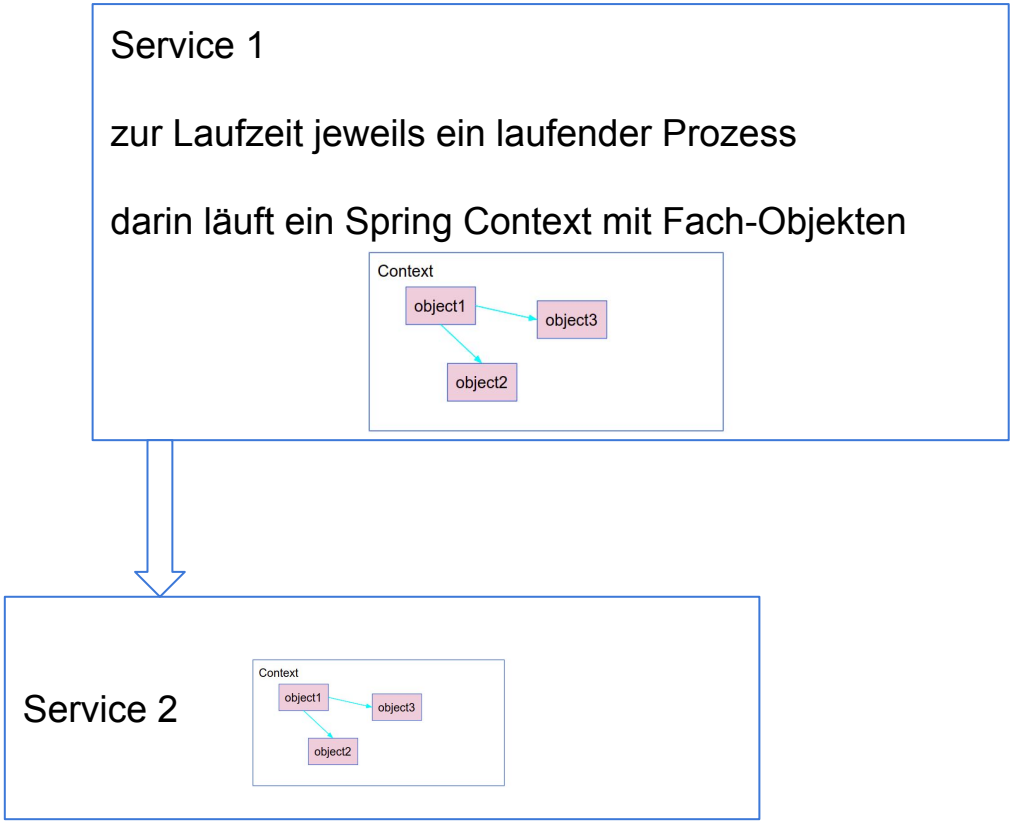


Aufgabe des Contexts

+ Identifikation der relevanten Fachklassen und Instanzierung von Fach-Objekten
+ Identifikation der Abhängigkeiten der Objekte und das Setzen der Abhängigkeit

Exkurs: Module versus Fachklassen

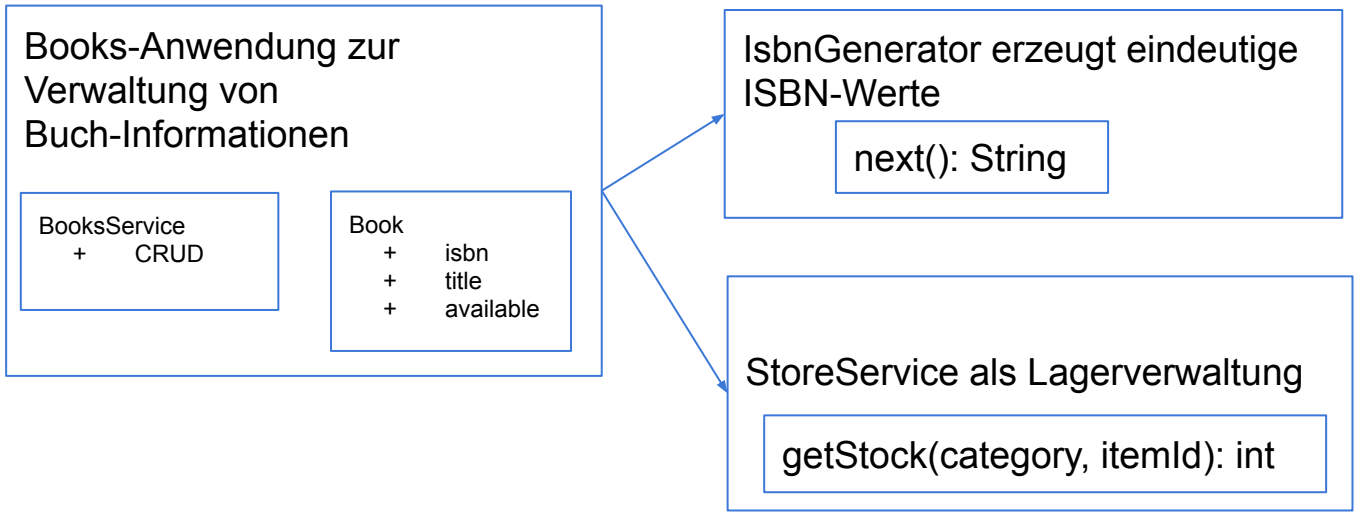
Service-oriented bzw. Microservices



Die Fachanwendung des Trainings

Modell

- Vollständig vorgegeben
- Fachlich einfach



Umsetzung

- Ist auch bereits vorhanden
  - Bisher
    - Die gesamte Datenhaltung In Memory
    - Zugriff ist nur für Actors im selben Prozess möglich
      - Actors = Test-Fälle
- Programmierung ist typisch für eine statisch typisierte Programmiersprache wie Java
  - Operationen sind in Schnittstellen definiert
  - Datenstrukturen sind simple Daten-Container (eigentlich structs oder records)
  - Operationen und Datencontainer definieren das API einer Fachanwendung
  - Zugehörige Implementierung ist eine Klasse, die die Schnittstelle implementiert

Umsetzung

- Bisher
  - Die Anwendung hat keinerlei Bezug zu Spring!
  - Die Anwendung selber ist jedoch CDI-konform!
    - Verifizierung: Relevante Fachklassen (MapBooksService, SimpleStoreService, RandomIsbnGenerator, CounterIsbnGenerator) werden im Rahmen der Anwendung NIEMALS mit new instanziert
    - Dependency ist ein Attribut vom Typ einer API-Schnittstelle + setter-Methode
      - Das ist das GoF-Pattern “Strategy”, das CDI-Pattern ist eine Meta-Pattern aus Strategy und Factory
  - Der Testfall übernimmt die Aufgaben des Contextes
    - new-Operatore
    - Aufruf der setter-Methoden

- Exkurs: “Spring” oder “Spring Boot”?
  - Spring = Spring Core ist das CDI-Framework
  - Spring Boot
    - Vereinfachter Build-Prozess
      - Dependency Management mit parent-pom und startern
    - Autoconfigure
      - “Convention over Configuration”
        - Welche Pakete sollen nach Spring-Informationen durchforstet werden?
        - Es wird automatisch eine Konfigurationsdatei namens application.properties eingeladen
  - Spring Core ist prinzipiell unabhängig von Spring Boot, aber es ist fast sinnlos, kein Spring Boot zu benutzen

Programmiermodell von Spring

- So nicht:
  - keine Namenskonventionen
  - benutzt keine Spring-Schnittstellen
    - relevant = “implements ContextAware”
- sondern ausgerichtet auf die Bereitstellung von Meta-Informationen
  - Externe XML-Konfiguration
  - **Java Annotations** (C#: Attributes)
    - @Component
      - oder @Service oder @Repository -> später
    - @Autowired
      - Referenzen auf Spring-relevante Objekte
    - @Value
      - Konfiguration auf einen Key, der in der application.xml eingetragen ist