

Training

Service-orientierte Architekturen

Ein **Flughafen** (Airport: name) hat 10 **Fluggastbrücken** (Gate: isOpen, Aircraft aircraft) **A01-A05** und **B01-B05**. An jedem **Gate** ist ein **Flugzeug** stationiert (**F01** ► A01, **F02** ► A02, **F03** ► A03, **F04** ► A04, **F05** ► A05, **F06** ► B01, **F07** ► B02, **F08** ► B03, **F09** ► B04, **F10** ► B05).

Details zur **Topologie** des **Flughafen** sind in der Datei „airport_apron_runway.pdf“ aufgeführt.

Folgende **Kreuzungen** auf dem Vorfeld (Apron) sind zusätzlich realisiert (**01**) O2 ► M1: X1, (**02**) N2 ► M1 : X2, (**03**) O2 ► L1: X3, (**04**) N2 ► L1: X4, (**05**) L2 ► N1: X5, (**06**) L2 ► O1: X6, (**07**) M2 ► O1: X7, (**08**) M2 ► N1: X8, (**09**) O5 ► M6: X9, (**10**) N5 ► M6: X10, (**11**) O5 ► L6: X11, (**12**) N5 ► L6: X12, (**13**) L5 ► O6: X13, (**14**) L5 ► N6: X14, (**15**) M5 ► O6: X15, (**16**) M5 ► N6. Aus Aspekten der Lesbarkeit sind diese textuell beschrieben und nicht in der Topologie aufgeführt.

Ein **Flugzeug** (Aircraft: id, manufacturer, numberOfSeatFirstClass, numberOfSeat-BusinessClass, numberOfSeatEconomyClass) **ist** ein **Service**, der mit Verhalten aus Events reagiert.

Ein Flugzeug hat 568 **Sitzplätze** (Seat: bookingClass, id). Zulässige Werte für bookingClass sind First, Business, Economy. Auf einem Sitzplatz kann maximal ein Passagier sitzen.

Ein Flugzeug hat zwei **Tragflächen** (Wing: wingPosition). Das Attribut wingPosition kann den Wert Left oder Right annehmen. Unter einer Tragfläche befinden sich zwei Triebwerke (Fan: id, rpm).

Ein **Triebwerk** hat 96 Schaufeln (Blade: id, material). Das Triebwerk implementiert die Methoden start, increaseRPM, decreaseRPM und shutdown. Die Methoden increaseRPM und decreaseRPM erwarten als Parameter einen Wert (value), um den der aktuelle Wert für rpm erhöht/reduziert wird. Das Attribut material bei Blade kann die Werte Titan und Carbon annehmen.

Zwischen Fan und Blade besteht eine physische Inklusion.

Eine Tragfläche hat zwei **Flap** (id, degree). Flap implementiert die Methoden neutral, levelOne, levelTwo und levelThree. Zwischen Tragfläche und Flap besteht eine Aggregation.

Ein Flugzeug hat fünf **Fahrwerke** (Gear: type, isDown, isBrakeSet). Zulässige Attribute für Typ bei Fahrwerk sind Front und Rear. Ein Fahrwerk hat 2-6 Räder (Wheel: id, diameter). Das Fahrwerk implementiert die Methoden up, down, setBrake und releaseBrake.

Zu einem Flugzeug gehört eine **Crew**.

Die Crew besteht aus zwei **Piloten** (Pilot: flightExperienceInMiles) und drei Flugbegleitern (FlightAttendant: isPurser). Piloten und Flugbegleiter sind Mitarbeiter.

Die zentralen **Methoden** taxi, holdShort, takeOff und land des Flugzeugs sind **als Facade** realisiert und werden durch Events von ApronControl bzw. dem Tower aktiviert.

Es existieren zwei **Frequenzbereiche** (i) ApronControl: 121.5, (ii) Tower: 128.5.

Über die Frequenzen werden **Events** von der Vorfeldkontrolle (ApronControl) bzw. dem Tower an das Flugzeug kommuniziert. Die **Kontrolle** der **Start-/Landebahn** obliegt dem **Tower**.

Die Frequenz wird an dem Kontrollpunkt Sx gewechselt.

Die Events sind (i) Taxi, (ii) HoldShort, (iii) RunwayClearedForTakeOff, (vi) RunwayClearedToLand.

Ein landendes Flugzeug empfängt das Event HoldShort vom Tower.

Ein startendes Flugzeug empfängt das Event HoldShort von der Vorfeldkontrolle (ApronControl).

Zu Beginn der **Simulation** wird die **Windrichtung** (Osten ► Westen, Start/Landung 08, Westen ► Osten, Start/Landung 26) zufällig bestimmt. Flugzeuge starten gegen den Wind landen mit dem Wind.

Zu Simulationszwecken befinden sich 10 Flugzeuge (F11-F20) im Landeanflug.

Landung (F11-F20) und Start (F01-F10) wechseln sich sukzessive einander ab.

Nachfolgend ist ein **Beispiel** für **Koordination/Kommunikation** aufgeführt.

(01) RunwayCleared-ToLand(F11,O8L), HoldShort(F11,S1), Taxi(F01,S04,O2 | O3 | O4 | O5 | X9 | M6), HoldShort(F01,S4), Taxi(F11,A01,O1), RunwayClearedForTakeOff(F01,O8R),

(02) RunwayClearedToLand(F12,O8R), HoldShort(F12,S3), Taxi(F02,S2,N2 | X4 | L1), HoldShort(F02,S2), Taxi(F12,A02,N6 | N5 | N4 | N3,A02), RunwayClearedForTakeOff(F02,O8L).

Die Vorfeldkontrolle und der Tower nutzen eine gemeinsame AODB (Airport Operations Database) für die Aufzeichnung (#, aircraft_id, unix_timestamp, event_message). Jedes Flugzeug zeichnet in einem FlightRecorder – realisiert als Logfile – die empfangenen Events mit Zeitstempel auf.

Zielsetzungen

- a) **Realisierung einer leistungsfähigen Service-orientierten Architektur.**
- b) Events sind mit den Informationen für einen reibungslosen Betrieb zu parametrisieren
- c) Wiederholung und Vertiefung des Wissens zu Service-orientierten Architekturen.
- d) Training kollaborative Software Entwicklung anhand einer komplexen Aufgabenstellung.

Wichtige Hinweise:

- **Implementierung** einer technisch einwandfrei lauffähigen Applikation in Java 11.
- **Test** der Implementierung mit **JUnit** und Gewährleistung der Funktionsweise.
- **Pro Team** wird **diese Aufgabe** bearbeitet.
- Nutzung der **camelCase-Notation**, um die Lesbarkeit zu vereinfachen.
- Zulässige externe Bibliotheken: guava*.jar, junit*.jar, opentest4j.jar und hsqldb.jar.
- Verwendung geeigneter **englischer** Begriffe für **Namen** und **Bezeichnungen**.
- **Erstellung einer vollständigen und verschlüsselten 7-Zip-Datei unter Beachtung des Prozedere für die Abgabe von Prüfungsleistungen und der Namenskonvention.**
- **Zeitansatz:** 15 Stunden
- **Abgabetermin:** Sonntag, 03.02.2019
- **Bewertung:** Testat