

Lufthansa | Fast Bag Drop

Spezifikation

Für den Flug LH 2121 um 22:00 Uhr mit einem Airbus A350-900 von Frankfurt (FRA) nach Hong Kong (HKG) werden alle Passenger über „**Fast Bag Drop**“ (**Automat**) eingescannt. Der Automat ist charakterisiert durch serialNumber (uuid), manufacturer (SMITH, WATSON), state (ON, OFF, LOCKED) und hat eine linke und rechte Sektion. Jede Sektion besteht aus einem Scanner für Passport, einen Drucker für Boarding Pass, Baggage Tag und ggf. Voucher sowie ein Conveyor Belt, einem Sensor, einem Scanner für Baggage, ein Lesegerät für die IDCard eines Employee und ein Display. Ein **Human** ist charakterisiert durch name (z.B. Nancy Sherman). **Passenger**, **Service Agent** und **Federal Police Officer** sind Human. Einem Passenger ist genau ein **Passport** zugeordnet. Ein Passport ist charakterisiert durch id (z.B. 7OYEP2K8Z). Ein **Employee** ist ein Human. Service Agent und Federal Police Officer sind Employee. Einem Employee ist eine **IDCard** zugeordnet. Die IDCard kennt nicht den Employee. Der Automat bietet die **Leistungen Startup, Shutdown, Unlock, Import, Check-In, Baggage Drop, Determine Weight, Scan Baggage, Scan Baggage Tag, Explosives Investigation, Data Analytics** und **Export** an. Check-In ist unvollständig und wird durch Baggage Drop vervollständigt. Baggage Drop ist unvollständig wird durch Determine Weight, Scan Baggage und Scan Baggage Tag vervollständigt. Explosives Investigation wird von einem Federal Police Officer durchgeführt und ist ein Sonderfall zu Baggage Drop. Mit **Import** wird eine CSV-Datei mit den Informationen zu flight, source, destination, gate, boarding time, ticket id, booking class, name und passport id von einem Service Agent in eine hash-basierte Datenbasis des Automaten importiert. Vor dem Automaten existieren zwei Warteschlangen [**i**] Business und [**ii**] Premium Economy und Economy. Die Kunden der Klasse Premium Economy werden priorisiert eingescannt. Die Gepäckstücke werden gemäß der **Spezifikation baggage_content.txt** erstellt. Den Passenger werden sukzessive die Gepäckstücke gemäß **Spezifikation assignment.txt** zugeordnet und die Einreihung der Passagiere in die korrespondierende Warteschlange vorgenommen. Einem Passagier ist genau ein Ticket zugeordnet. Ein Ticket besteht aus einem linken und rechten Abschnitt, realisiert als physische Inklusion. Der linke und rechte Abschnitt des Tickets ist je charakterisiert durch name (z.B. Nancy Sherman), flight (LH 2121), bookingClass (B für Business, P für Premium Economy und E für Economy), source (FRA), destination (HKG), date (aktuelles Datum im Format dd.MMM), id (z.B. ccjwf639o4twzc0) und sequence (fortlaufende Nummer beginnend bei 1). Der rechte Abschnitt des Tickets erhält zusätzlich die Information zum gate A05 und die boardingTime 22:00. Ein Service Agent ist für Startup und Shutdown des Automaten verantwortlich. Die **IDCard** ist charakterisiert durch uuid, status (ACTIVE, LOCKED) und **RFID-Chip**. Die PIN wird verschlüsselt auf dem RFID-Chip gespeichert. Für den Automaten und die IDCard ist in einer zentralen Konfiguration der Algorithmus (AES oder DES¹) für die Verschlüsselung festgelegt. Der RFID-Chip ist mit der IDCard über eine Kompositionssstruktur physisch inkludiert. **Startup/Shutdown** | Für Startup und Shutdown hält der Service Agent seine IDCard an das Lesegerät des Automaten und gibt über das Display die PIN ein. **Unlock** | Für das Entsperren eines Automaten im Status LOCKED hält der Federal Police Officer seine IDCard an das Lesegerät des Automaten und gibt über das Display die PIN ein. Nach dreimaliger Falscheingabe der PIN wird der Status der IDCard auf LOCKED gesetzt. Gesperrte IDCard werden vom Lesegerät über eine Fehlermeldung auf dem Display abgewiesen. Nach korrekter Eingabe der PIN wird in Abhängigkeit von dem Status ein/zwei Button angezeigt. Bei dem Status OFF wird der Button Startup angezeigt. Bei dem Status ON werden die Button Export und Shutdown angezeigt. Bei Drücken des Button Startup wechselt der Automat in den Status ON. Bei dem Status LOCKED wird der Button Unlock angezeigt. Bei Drücken des Button Unlock wechselt der Automat in den Status UNLOCKED.

1 Bitte nutzen Sie die Standard-Bibliotheken von Java.

Check-In | Die Simulation beginnt mit der linken Warteschlange, den Passenger der Business Class. Der Passenger legt seinen Passport auf den Scanner und es wird zu der id des Passport ein ggf. registriertes Ticket gesucht. Wurde kein Ticket gefunden, wird am Display die Fehlermeldung „Sorry. No registered ticket found for [name] and flight LH2121“ angezeigt. Wurde ein Ticket gefunden, wird ein freier Sitzplatz der korrespondierenden Buchungsklasse ermittelt und die vollständigen Informationen zu dem Ticket auf dem Display mit der Fragestellung „Proceed with check-in for flight LH2121?“ und zwei Button Yes und No angezeigt. Bei Drücken von No wird am Display die Meldung „Check-In cancelled by user“ angezeigt. Bei Drücken von Yes wird am Display die Meldung „Please enter number of checked-in baggage“ angezeigt. Die dem Passenger zugeordnete Anzahl der Baggage wird ermittelt und erfasst. Der Passenger stellt das Baggage auf das Conveyor Belt. Der Sensor² registriert dies und das Baggage wird gewogen. Das Gewicht des Baggage wird im Display oberhalb zum Eingang des korrespondierenden Conveyor Belt angezeigt. Ist das Gewicht des Baggage im zulässigen Bereich wird durch einen dedizierten Scanner die Suche mit Brute Force nach dem Schlüsselwort explosives in content durchgeführt. Wird kein Sprengstoff gefunden, wird der Baggage Tag gedruckt. Einem Boarding Pass sind beliebig viele Baggage Tag zugeordnet. Ein Baggage Tag gehört zu genau einem Boarding Pass. Der Passenger bringt den Tag am Baggage an, ein Scanner registriert dies und das Baggage wird vom Conveyor Belt entfernt (null). Aus Aspekten der Auditierung wird für jedes eingecheckte Baggage in einem **Record** die Informationen zu Zeitstempel in Nanosekunden des abgeschlossenen Scan auf Explosives, Ticket, Baggage Tag und result (OK, NOK) in einer dedizierten hash-basierten Datenbasis gespeichert. Bei Überschreiten des zulässigen Gewichts von 23 kg erfolgt ein Signalton (beep) und die Anzeige der Fehlermeldung „Baggage exceeds weight limit of 23 kg“. In diesem Fall nimmt der Kunde das Gepäckstück vom Conveyor Belt. Bei einem Fund wird der Automat gesperrt (LOCKED) und die Federal Police zwecks Explosives Investigation benachrichtigt. Der Federal Police Officer entfernt das Baggage vom Conveyer Belt und entsperrt durch seine IDCard mit Eingabe der PIN auf dem Display den Automaten. Nachdem alle Baggage des Passenger eingecheckt und/oder abgewiesen wurden, wird der Boarding Pass und ggf. Voucher gedruckt. Ein Passenger der Business Class enthält einen Voucher für die Lounge. Ein Passenger in der Buchungsklasse Premium Economy erhält einen Voucher für ein AC/DC-Konzert. Der Passenger nimmt den Boarding Pass und ggf. Voucher an sich und verlässt (null) den Automaten. Der nächste Passenger beginnt den Check-In. Die Simulation endet, wenn beide Warteschlangen leer sind. **Data Analytics** | Für die statistische Auswertung sind zwei Abfragen **[i]** Gesamtgewicht der Gepäckstücke gruppiert nach Buchungsklasse und **[ii]** Passagierliste (name | booking class | seat id | count baggage) je Buchungsklasse und absteigend sortiert nach Nachnamen, bei Aufruf der Leistung Data Analytics am Display auszugeben. **Export** | Generierung einer CSV-Datei fast_bag_drop.csv mit den Informationen Zeitstempel in Nanosekunden des abgeschlossenen Scan auf Explosives, name, booking class, passport id, ticket id, baggage tag id. **Test Management** | **[01]** Warteschlange mit Passagieren und Gepäck wird korrekt initialisiert, **[02]** Leistungen Startup, Shutdown, Data Analytics und Export können nur von einem Service Agent ausgeführt werden, **[03]** Leistungen Startup und Shutdown werden korrekt ausgeführt, **[04]** Boarding Pass wird korrekt erstellt, **[05]** Baggage Tag wird korrekt erstellt, **[06]** Leistung Check-In mit Baggage Drop wird im Normalfall korrekt durchgeführt und **[07]** Leistung Check-In mit Baggage Drop wird im Sonderfall Explosives Investigation korrekt ausgeführt. **Zukünftige Anforderungen** | **[01]** Wahlweise Identifizierung über Passport, Fingerprint oder Gesichtserkennung, **[02]** Suche in Baggage (content) wahlweise mit den Algorithmen Brute Force, Boyer-Moore und Knuth-Morris-Pratt, **[03]** Baggage Tag wird um QR-Code³ mit den Informationen zu source, destination und flight erweitert und **[04]** Wahlweise Nutzung einer App für elektronischen Boarding Pass und korrespondierende Baggage Tag(s).

2 Realisierung über Design Pattern „Observer“ | Bibliothek wird vom Dozenten bereitgestellt.

3 ZXing | <https://simplesolution.dev/java-generate-qr-code-zxing/>



Wichtige Hinweise für die Bearbeitung

- Die **Bearbeitung** dieser Aufgabenstellung im **Team mit sieben Studierenden S01 - S07**.
S01 : Anforderungsmanagement (SOPHIST Master-Methode), Anwendungsfalldiagramm(e), Aktivitätsdiagramme für die Anwendungsfälle Startup, Shutdown und Unlock.
S02 : Aktivitätsdiagramm(e) für Check-In mit assoziierten Unterprozessen und Klassendiagramme (auszugsweise) für zukünftige Anforderungen.
S03 : Klassendiagramm und Paketdiagramm für Basis.
S04 : Aktualisierung Klassendiagramm auf Basis **S03** unter konsequenter Berücksichtigung der SOLID-Design Prinzipien und zukünftigen Anforderungen.
S05 : Sequenzdiagramm(e) und Zustandsdiagramm(e).
S06 : Implementierung auf Basis des aktualisierten Klassendiagramms **S04**.
S07 : Testmanagement | JUnit und/oder Mockito für Implementierung **S06**.
- Zielsetzungen sind **[i] Training der Abstraktion und Modellierung⁴** sowie **[ii] Implementierung** und **[iii] Testmanagement** mit JUnit und/oder Mockito.
- Verwenden Sie den **Style Guide** und geeignete **englische Begriffe**.
- Bitte achten Sie bei der **Modellierung** auf ein **geordnetes Gesamtbild** (Look & Feel) sowie die **Konsistenz zwischen** den **Modellen** und der **Implementierung**.
- **Implementieren** Sie einen technisch einwandfrei funktionierenden Automat „Fast Bag Drop“ mit den spezifizierten Anwendungsfällen.
- Erstellen Sie eine **zielführende Teststrategie** für die Anwendungsfälle. **Implementieren** Sie ein leistungsfähiges **Testmanagement**.
- Für die **Modellierung** wird **Visual Paradigm Community 16** und das **Template** genutzt.
- Die **Hinweise** der **Code Inspection** in IntelliJ sind **weitestgehend zu berücksichtigen**.
- Als **Entwicklungsumgebung** wird **[i] Java SE Development Kit 17.0.4.1 (LTS)**, **[ii] IntelliJ IDEA Community oder Ultimate 2022.2.1** und **[iii] gradle 7.5.1** genutzt.

Je Team wird **eine unverschlüsselte 7-Zip-Datei** (Kompressionsstärke: Ultra) mit der Bezeichnung **fast_bag_drop_[team_id].7z** in **Moodle hochgeladen**.

⁴ Nur Modellierung von Klassen für originäre Funktionalität des Automaten. Utility-Klassen und Annotations für die Simulation sind nicht zu modellieren.