

## 01 (Drohne und Stadtgebiet | Design Patterns)

Die Ambulanz-Drohne mit Künstlicher Intelligenz wird mit einem **Builder** erstellt. Die Drohne hat vier Hauptausleger. An jedem Hauptausleger existieren zwei Ausleger. Hauptausleger und Ausleger sind über eine Kompositionsstruktur (**Composite**) miteinander verbunden. An jedem Ausleger befinden sich an der Ober- und Unterseite je ein Elektromotor. Mit dem Elektromotor sind zwei Rotorblätter verbunden. Eine zentrale Steuereinheit (**Mediator**) steuert die Elektromotoren synchron über die Kommandos TakeOff, Forward, Left, Right, Land. Des Weiteren sind ein LiDAR, eine Kamera, ein GPS, zwei Scheinwerfer, eine Box mit der zentralen Steuereinheit integriert. Die Parameter allowedAtNight [true | false], antiTheftProtection [true | false], enableHooter [true | false] werden über ein Menü [i] configure parameter, [ii] undo, [iii] save, [iv] check und [v] exit konfiguriert. Um in das Menü zu gelangen, wird die **Drohne mit dem (Konsolen-)Parameter -configure** gestartet. *Undo* macht die letzte Änderung eines Parameters rückgängig (**Memento**), *save* speichert die Konfiguration in eine JSON-Datei und *check* prüft dediziert die mit der zentralen Steuereinheit assoziierten Bauteile (**Visitor**).

Der Drone Port verfügt über einen Sender/Empfänger zwecks Kommunikation mit der zentralen Steuereinheit der Drohne. Das Event bei einem Notfall bzw. Auftrag wird an die zentrale Steuereinheit der Drohne kommuniziert und dort durch AES entschlüsselt. Die Steuereinheit der Drohne wandelt die Route in eine Liste von Kommandos um. Die Liste der Kommandos wird sukzessive abgearbeitet und die Drohne landet neben dem Menschen, welcher mit seinem Smartphone den Notruf abgesetzt hat. Einer Box der Drohne werden zwei Elektroden entnommen und der Mensch mit dem Herzinfarkt erfolgreich reanimiert. Nach erfolgreicher Reanimation werden die Elektroden in die Box zurückgelegt und die Zentraleinheit über einen Sensor (**Observer**) automatisch informiert. Die Zentraleinheit erstellt die Liste mit den Kommandos für den Rückflug und die Drohne kehrt zum Drone Port zurück.

Jeder Mensch ist durch sein biometrisches Merkmal Gesicht 25x25 mit Zeichen aus dem Pool [.,+,\*,#] eindeutig identifiziert. Zu jedem Gesicht existiert eine eindeutige zehnstellige alphanumerische ID mit Zeichen aus dem Pool [a-z,0-9] und das zufällige Alter im Bereich [10-100].

Das **Stadtgebiet Mosbach** wird dargestellt als eine Matrix mit 1000x1000 Feldern. Ein Feld kann die Zeichen Leerzeichen, D (drone port), E (emergency call centre), R (restricted), S (skyscraper) und H (human) annehmen. Das Stadtgebiet wird mit Leerzeichen initialisiert und sukzessive die Zeichen D, E, R, S und H mit dem nachfolgend definierten Schema gesetzt. Für die Stationierung der vier Drone Ports wird das Stadtgebiet in vier gleich große Sektionen unterteilt. Ein Zeichen D wird zentriert in jeweils einer Sektion positioniert. Das Emergency Center (EC) ist zentriert im Stadtgebiet zu positionieren. Nach dem Zufallsprinzip sind 32500 R und 2500 S als Hindernisse in dem Stadtgebiet zu verteilen. Das EC kann von der Drohne überflogen werden und ist kein Hindernis. Nach der Positionierung von der Zeichen D, E, R und S werden 1000 2er-Paare H an freien Stellen im Stadtgebiet platziert. Um die beiden H herum, muss mindestens ein Leerzeichen existieren, damit die Drohne landen kann. Jeder Mensch verfügt über ein Smartphone mit KI-Unterstützung. Es werden 5000 Notfälle simuliert. Für den Notfall wird ein Paar im Stadtgebiet zufällig ausgewählt. Ein zufällig ausgewählter Mensch des Paares erleidet einen Herzinfarkt. Der gesunde Mensch des korrespondierenden Paares aktiviert die Kamera von seinem Smartphone und hält diese auf den anderen Menschen. Die Kamera klassifiziert die Situation als Notfall und kontaktiert automatisch den Bot des EC. Das Smart Phone kommuniziert an den Bot die Position des anrufenden Menschen und das Gesicht des anderen Menschen.

## 02 (EC, Pfadplanung und EventBus mit Google Guava)

---

Das EC verfügt über einen Bot, der den Notruf annimmt und die notwendigen Maßnahmen einleitet. Bei dem Notruf wird automatisch die Position des Anrufers und das Gesicht des anderen Menschen übermittelt. Das EC nutzt eine leistungsfähige **Datenbasis**, welche **auf Gesichtserkennung** zwecks Ermittlung der ID **optimiert** ist. Mit dem **A\*-Algorithmus oder D\*-Algorithmus** und der **Topologie** des **Stadtgebietes** ermittelt das EC den Drone Port, dessen Position die geringste Entfernung zu der Position des Anrufers hat. Das **EC** ist **über** einen **EventBus** mit dem **Drone Port verbunden**. Die Informationen zu ID des Menschen mit einem Herzinfarkt sowie die Route werden mit **AES verschlüsselt** und über ein Event an den Drone Port übermittelt. Für die **Pfadplanung** stehen der **A\*-Algorithmus und D\*-Algorithmus in dynamisch austauschbaren und digital signierten Komponenten** dem EC zur Verfügung. Für den Algorithmen A\* und D\* können existierende Implementierungen aus dem Internet als Basis genutzt werden. In einem Logfile ist die Evaluierung der Pfade nachvollz. zu protokollieren.

## 03 (Qualitäts- und Testmanagement)

---

1. Die Drohne wird vollständig durch den Builder erstellt.
2. Das Stadtgebiet Mosbach wird vollständig aufgebaut.
3. Erleidet ein Mensch eines Paares einen Herzinfarkt wird ein Notruf an das EC abgesetzt.
4. Der Bot des EC empfängt den Notruf und die Positionsangabe des Anrufers.
5. Für die Pfadberechnung akzeptiert das EC nur digital signierte Komponenten.
6. Die Drohne mit der geringsten Entfernung zu der Position des Anrufers wird selektiert.
7. Die eindeutige ID zu einem Gesicht wird korrekt ermittelt.
8. Die Zentraleinheit der Drohne empfängt ein Event mit vollständigen Informationen (ID, Route).
9. Die Zentraleinheit übersetzt die Informationen zur Route in korrekte Kommandos der Drohne.
10. Die Drohne landet neben dem Menschen, welcher den Notruf abgesetzt hat.
11. Nach Landung der Drohne wird der Mensch mit dem Herzinfarkt erfolgreich **reanimiert**.
12. Nach Reanimation bzw. dem Zurücklegen der Elektroden kehrt die Drohne zum Port zurück.

## 04 (Data Analytics | Lambda/Streams)

---

Die Drohne ist mit einem Flight Recorder – realisiert als **HSQLDB** – ausgestattet. Das EC kann Remote die Daten auslesen. Das EC sendet hierzu ein Event an die Drone Ports. Jeder Drone Port sendet eine Anfrage an die Zentraleinheit (ZE) der korrespondierenden Drohne. Alle Daten des Flight Recorder werden von der ZE ausgelesen, mit AES verschlüsselt und an den Drone Port kommuniziert. Der Drone Port erstellt ein Event und sendet die Informationen an das EC. Nach der Simulation von 5000 Notfällen soll durch das EC nachfolgend spezifizierte Analysen über alle Drone Ports durchgeführt werden.

1. Gesamtanzahl der Einsätze.
2. Top 2 Drone Ports mit der höchsten Anzahl von Einsätzen.
3. Sektor im Stadtgebiet mit den häufigsten Einsätzen.
4. Anzahl der Einsätze gruppiert nach Sektoren im Stadtgebiet.
5. Anzahl der Einsätze je Altersgruppe im Bereich 60-70, absteigend sortiert nach Anzahl.
6. Durchschnittliche Flugstrecke der Einsätze.
7. Top 3 Einsätze mit maximaler Flugstrecke und Information zu der Route.
8. Top 5 Einsätze mit minimaler Flugstrecke und Information zu der ID der reanimierten Person.

- Die **Aufgaben** sind innerhalb des Teams **eigenverantwortlich** zu **verteilen**.
- Nutzung geeigneter **englischer Begriffe** und **Bezeichnungen**.
- Als Entwicklungsumgebung wird [i] **Java SE Development Kit 16.0.1**, [ii] **IntelliJ IDEA Community oder Ultimate 2021.1** und [iii] **gradle** genutzt.
- **Zugelassene Bibliotheken:** JSON, Google Guava, HSQLDB.
- **Implementierung** und **Test** einer technisch einwandfrei lauffähigen Applikation durch Bearbeitung und Integration der spezifizierten Individualaufgaben je Studierenden.
- Das **Qualitäts- und Testmanagement** ist durch den Einsatz leistungsfähiger JUnit-Konstrukte **effektiv und effizient** zu **gestalten**.
- **Clean-Up** und **Formatierung** des **Source Code**.
- Je Team wird eine unverschlüsselte 7-Zip-Datei se\_II\_2021\_kpl\_drohne\_[matrikelnummern].7z (Kompressionsstärke Ultra) mit dem vollständigen IntelliJ-Projekt und readme.txt hochgeladen.
- Der Abgabe ist eine **readme.txt** mit der **Zuordnung Matrikelnummer** zu **Aufgabe** beizufügen.
- **Abgabetermin:** Upload in Moodle bis spätestens Sonntag, **13.06.2021**.
- **Bewertung:** 50 Punkte je Aufgabe