

# FlugMonitor OpenSky

## Datenquelle

- ADS-B Technologie**
  - Frequenz 1090 MHz
  - Automatische Positionssendung
  - Reichweite ca. 400 km
- OpenSky Network**
  - Gemeinnütziges Forschungsprojekt
  - Freiwillige Empfänger weltweit
  - Kostenlose REST-API
  - Rate-Limit: ~10 Req/min anonym
  - Erhöhtes Limit nach Registrierung
- Java-API**
  - OpenSkyApi.getOpenSkyApiInstance()
  - Rückgabe: OpenSkyStates
- StateVector (pro Flugzeug)**
  - ICAO24- Transpondercode
  - Geografische Position (Lat/ Lon)
  - Geodätische Höhe in Metern
  - Geschwindigkeit & Kurs
  - onGround-Flag

## Benutzeroberfläche

- Fenster (JFrame)**
  - Größe 610 x 900 px
  - System Look & Feel
  - Zentriert beim Start
- Buttons**
  - Aktualisieren (Dunkelblau)**
    - Symbol #
    - Startet SwingWorker
    - Deaktiviert während Abruf
  - Kopieren (Dunkelgrün)**
    - Symbol #
    - Schreibt Tabelle in Clipboard
    - Bestätigung 3 Sekunden
- Tabelle**
  - 18 Höhenbereiche + Summe
  - Spalte 1: Flugbereich in Meter
  - Spalte 2: Anzahl Flugzeuge
  - ZebraRenderer (Zebrastrreifen)
  - Summenzeile grün + fett
  - JScrollPane (scrollbar)
- Statusleiste**
  - Zeitstempel nach Abruf
  - Fehlermeldung bei Netzwerkfehler
  - kleinhirn.eu rechts dezent

## Technische Architektur

- Klassen**
  - FlugMonitorSwing (JFrame)
    - initialisiereUI()
    - ladeStatistik()
    - berechneVerteilung()
    - bestimmeIndex()
    - aktualisiereTabelle()
    - kopiereTabelle()
  - DiagramPanel (JPanel)
    - setDaten(int[])
    - paintComponent(Graphics)
  - ZebraRenderer
  - Bereich (Java Record)
- Threading-Modell**
  - Event Dispatch Thread (EDT)
    - Alle UI-Operationen
    - Buttons aktivieren/deaktivieren
  - SwingWorker (Hintergrund)
    - doInBackground() # Netzwerk
    - done() # UI aktualisieren
    - Verhindert UI-Einfrieren
- Datenfluss**
  - Klick # SwingWorker.execute()
  - doInBackground: getStates()
  - berechneVerteilung() # int[18]
  - aktualisiereTabelle(int[])
  - diagramPanel.setDaten(int[])
- Java 21 Features**
  - Records (Bereich-Klasse)
  - Lambdas (ActionListener)
  - var (lokale Typinferenz)

## Höhenbereiche

- Boden**
  - Parken (onGround oder null)
  - Farbe: Graugrün
- Niedrigflug 1-5.000 m**
  - 1 - 500 m (Start / Landung)
  - 500 - 1.000 m
  - 1.000 - 2.000 m
  - 2.000 - 3.000 m
  - 3.000 - 4.000 m
  - 4.000 - 5.000 m
  - Farben: Grün # Türkis
- Mittelflug 5.000-9.000 m**
  - 5.000 - 6.000 m
  - 6.000 - 7.000 m (Minimum!)
  - 7.000 - 8.000 m
  - 8.000 - 9.000 m
  - Farben: Hellblau # Blau
- Reiseflug 9.000-13.000 m**
  - 9.000 - 10.000 m
  - 10.000 - 10.500 m
  - 10.500 - 11.000 m
  - 11.000 - 11.500 m (Maximum!)
  - 11.500 - 12.000 m
  - 12.000 - 13.000 m
  - Farben: Blau # Violett
- Stratosphäre**
  - über 13.000 m
  - Militär / Spezial-fts
  - Farbe: Grauviolett

## Diagramm Graphics2D

- Rendering**
  - Antialiasing aktiviert
  - LCD Text-Antialiasing
  - g.create() # dispose()
  - Dunkler Hintergrund #10141E
- Balken-Zeichnung**
  - Spurlinie (dunkel, volle Breite)
  - RoundRectangle2D (rx=4)
  - Glanzstreifen (Alpha 35/255)
  - Wert rechts (drawString)
- Farbverlauf**
  - Graugrün = Boden
  - Grün = Niedrigflug
  - Türkis = Übergang
  - Blau = Mittelflug
  - Violett = Reiseflug (Gipfel)
  - Grauviolett = Stratosphäre
- Achsen & Gitter**
  - 5 X-Achsen-Gitterschritte
  - Deutsche Tausend-Trennzeichen
  - Y-Labels: Monospaced 10pt
  - Diagramm-Rahmen: drawRect
- Koordinaten**
  - MarginLeft 152 px (Y-Labels)
  - MarginRight 52 px (Werte)
  - MarginTop 30 px (Titel)
  - MarginBottom 30 px (X-Achse)

## Build & Deployment

- Maven (pom.xml)**
  - GroupId: de.wenzlaff.tools
  - ArtifactId: flug-monitor-swing
  - Version: 2.0.0
  - Java: 21
  - Dependency: opensky-api
- Maven Shade Plugin**
  - Alle Deps gebündelt (Fat-JAR)
  - Manifest: MainClass gesetzt
  - Ausgabe: ...2.0.0.jar
- Starten**
  - mvn package
  - java -jar target/...jar
  - Oder direkt aus IDE
- Voraussetzungen**
  - Java 21 JDK
  - Maven 3.8+
  - Internetverbindung (OpenSky)

## Erweiterungen

- Daten**
  - Bounding Box (nur Deutschland)
  - Historische Daten (History-API)
  - Flugzeugtyp-Filter (ICAO-Kat.)
  - Airline-Filter (Callsign-Prefix)
- Benutzeroberfläche**
  - Auto-Aktualisierung (Timer)
  - Dark Mode (FlatLaf)
  - Weitkarte (JMapView)
  - Tagesverlauf-Liniendiagramm
- Export**
  - CSV-Export (java.nio.file)
  - PDF-Report
  - JSON-Ausgabe
- Netzwerk**
  - Authentifizierung (mehr Requests)
  - WebSocket (Push statt Poll)
  - Proxy-Unterstützung