

---

## FMS

Implementieren Sie ausgewählte Teile eines Flug Management Systems (FMS).

### Einleitung

Betrachten Sie das Klassendiagramm (1) um einen Überblick zu erlangen. Beachten Sie, dass im package `FMS.provided` einige Klassen bereits gegeben sind.

### Aufgabe

Implementieren Sie die Klassen

Aufgabe	Punkte
<b>Flight</b>	<b>29</b>
- getters	2
- setters & helper	4
- constructors	4
- natural ordering	2
- add(Staff)	4
- add (Passenger)	4
- readyToBoard	2
- board(Passenger)	4
- boardingCompleted	2
- getBonusMiles	1
<b>ScheduledFlight</b>	<b>5</b>
- constructors	3
- getBonusMiles	2
<b>Charter</b>	<b>5</b>
- constructor	3
- getBonusMiles	2
<b>DepartureComparator</b>	<b>4</b>
<b>OriginMatcher</b>	<b>4</b>
<b>Main</b>	<b>13</b>
- main	3
- export	5
- filter	5

laut Spezifikation in den javadocs.

### Hinweise

Implementieren Sie zunächst `Flight`, `Charter` und `ScheduledFlight`.

### Anhang

Verwenden sie folgende Code Snippets:

- `Flight.toString`:

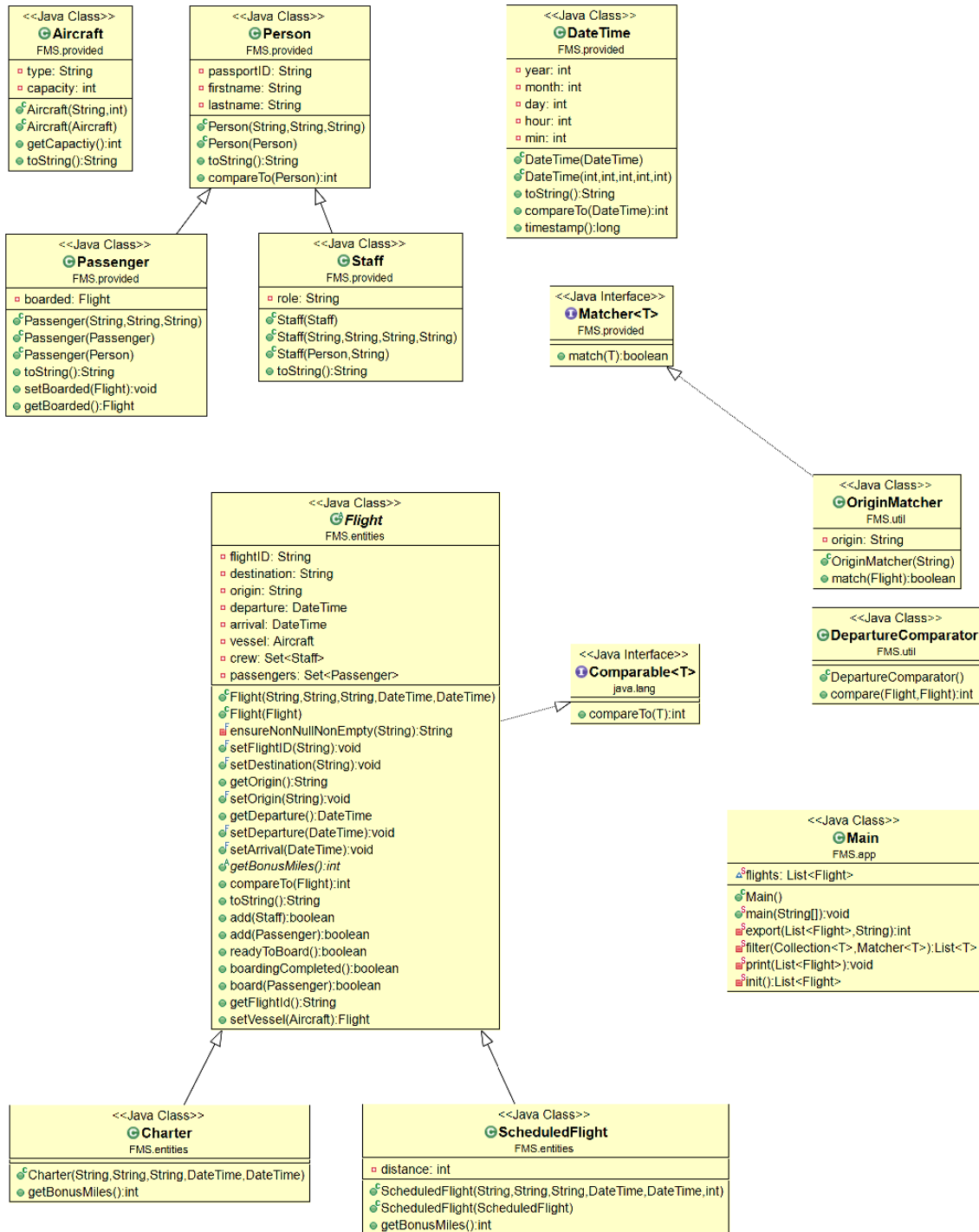


Figure 1: FMS Klassendiagramm

---

```
@Override
public String toString() {
    return String.format(
        "%5s from %3.3S (%s) to %3.3S (%s)" + " with a crew of %d and %d passengers "
        + "<%s> boarding%scompleted \textbackslash{}%s\textbackslash{}%s",
        flightID,
        origin,
        departure,
        destination,
        arrival,
        crew == null ? 0 : crew.size(),
        passengers == null ? 0 : passengers.size(),
        vessel == null ? "no vessel" : vessel.toString(),
        boardingCompleted() ? " " : " not ", crew, passengers);
}
```

- Main.init

```
private static List<Flight> init() {

    // ----- aircrafts
    List<Aircraft> crafts = new LinkedList<>();
    crafts.add(new Aircraft("A3", 3));
    crafts.add(new Aircraft("C4", 4));
    crafts.add(new Aircraft("Boeing 737", 175));

    // ----- passengers
    List<Passenger> passengers = new LinkedList<>();
    passengers.add(new Passenger("John", "Doe", "AT00004711"));
    passengers.add(new Passenger("Jane", "Doe", "AT00000815"));
    passengers.add(new Passenger("John", "Jackson", "US00004711"));
    passengers.add(new Passenger("Jack", "Doe", "UK00000007"));
    passengers.add(new Passenger("Jack", "Johnson", "DE00004711"));

    // ----- staff
    List<Staff> staff = new LinkedList<>();
    staff.add(new Staff("Rip", "Riley", "ISIS666", "Pilot"));
    staff.add(new Staff("Cheryl", "Tunt", "ISIS456", "Chef de Cabin"));
    staff.add(new Staff("Lana", "Kane", "ISIS001", "Stewardess"));

    // ----- flights
    List<Flight> flights = new LinkedList<>();
    flights.add(new ScheduledFlight("OS006", "FRA", "VIE", new DateTime(2018, 06, 25, 6, 30),
        new DateTime(2018, 06, 25, 8, 55), 800));
    flights.add(new Charter("OS001", "VIE", "VIE", new DateTime(2018, 06, 23, 6, 30),
        new DateTime(2018, 06, 23, 8, 55)));
    flights.add(new Charter("OS381", "ARN", "VIE", new DateTime(2018, 05, 25, 6, 30),
        new DateTime(2018, 05, 25, 9, 13)));
    flights.add(new Charter("OS001", "VIE", "JFK", new DateTime(2018, 07, 25, 4, 30),
        new DateTime(2018, 07, 25, 7, 20)));
    flights.add(new Charter("OS502", "JFK", "VIE", new DateTime(2018, 06, 28, 18, 30),
        new DateTime(2018, 06, 28, 19, 55)));
    flights.add(new ScheduledFlight("OS007", "VIE", "CDG", new DateTime(2018, 06, 28, 21, 00),
        new DateTime(2018, 06, 29, 0, 05), 1000));
    flights.add(new ScheduledFlight("OS008", "VIE", "CDG", new DateTime(2018, 06, 1, 6, 30),
        new DateTime(2018, 06, 1, 8, 05), 1000));
}
```

---

```

Flight f;

// ----- set up flight
f = flights.get(0).setVessel(crafts.get(0));
f.add(staff.get(0));
f.add(staff.get(1));
f.add(staff.get(2));
// list passengers
f.add(passengers.get(0));
f.add(passengers.get(1));
f.add(passengers.get(2));
f.add(passengers.get(3)); //exceeds vessel capacity
// board passengers
f.board(passengers.get(0)); // board listed passengers
f.board(passengers.get(1));
f.board(passengers.get(2));
f.board(passengers.get(3)); // board a passenger not listed


// ----- set up flight
f = flights.get(1).setVessel(crafts.get(1));
f.add(passengers.get(1));
f.add(passengers.get(2));
f.add(passengers.get(3));
f.add(new Staff(passengers.get(0), "Pilot")); //a passenger becomes a pilot
f.board(passengers.get(1)); // already on board another flight
f.board(passengers.get(3)); // board a passenger


// ----- set up flight
f = flights.get(2).setVessel(crafts.get(2));
Passenger p = new Passenger(staff.get(0)); //staff becomes a passenger
f.add(passengers.get(2));
f.add(passengers.get(1));
f.add(passengers.get(3));
f.add(p);
f.board(p); //no boarding without crew


// ----- set up flight
f = flights.get(3);
f.add(passengers.get(2));
f.add(passengers.get(1));
f.add(passengers.get(3));


// ----- set up flight
f = flights.get(4);
f.add(passengers.get(2));
f.add(passengers.get(1));
f.add(passengers.get(3));


// ----- set up flight

```

---

---

```
    f = flights.get(5);
    f.add(passengers.get(2));
    f.add(passengers.get(1));
    f.add(passengers.get(3));

    return flights;
}
```