Airport - Practica 06 LDH

Generated by Doxygen 1.9.2

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Namespace Documentation

4.1 Package es.ull.flights

Classes

- class Flight

    This class defines the attributes of a flight.
- class **FlightTest**

    This class defines different tests for the class Flight.

4.2 Package es.ull.passengers

Classes

- class Passenger

    This class defines the attributes of a flight.
- class **PassengerTest**

    This class defines different tests for the class Passenger.

Chapter 5

Class Documentation

5.1 es.ull.flights.Flight Class Reference

This class defines the attributes of a flight.

Public Member Functions

- Flight (String flightNumber, int seats)

  Flight information.
- String getFlightNumber ()

  Get the number of the flight.
- int getNumberOfPassengers ()

  Get number of passengers on board of the flight.
- boolean addPassenger (Passenger passenger)

  Adding passengers to a flight.
- boolean removePassenger (Passenger passenger)

  Remove a passenger from a flight.

5.1.1 Detailed Description

This class defines the attributes of a flight.

Definition at line 34 of file Flight.java.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 Flight()

```
es.ull.flights.Flight.Flight (
             String flightNumber,
             int seats )
```

Flight information.

Parameters

flightNumber	flight number
seats	seat number

Definition at line 56 of file Flight.java.

5.1.3 Member Function Documentation

5.1.3.1 addPassenger()

```
boolean es.ull.flights.Flight.addPassenger (
             Passenger passenger )
```

Adding passengers to a flight.

Parameters

passenger	passenger booking a flight

Returns

verify if passenger successfully booked a flight

Definition at line 86 of file Flight.java.

5.1.3.2 getFlightNumber()

```
String es.ull.flights.Flight.getFlightNumber ( )
```

Get the number of the flight.

Returns

flight number

Definition at line 69 of file Flight.java.

5.1.3.3 getNumberOfPassengers()

```
int es.ull.flights.Flight.getNumberOfPassengers ( )
```

Get number of passengers on board of the flight.

Returns

number of passengers

Definition at line 77 of file Flight.java.

5.1.3.4 removePassenger()

```
boolean es.ull.flights.Flight.removePassenger (
             Passenger passenger )
```

Remove a passenger from a flight.

Parameters

passenger	passenger who has booked the flight

Returns

verify if passenger was successfully removed from the flight

Definition at line 99 of file Flight.java.

The documentation for this class was generated from the following file:

- src/main/java/es/ull/flights/Flight.java

5.2 es.ull.passengers.Passenger Class Reference

This class defines the attributes of a flight.

Public Member Functions

- Passenger (String identifier, String name, String countryCode)

  Passenger attends a flight.
- String getIdentifier ()

  Get passenger identifier.
- String getName ()

  Get passenger name.

- String getCountryCode ()

  Get passenger country code.
- Flight getFlight ()

  Get passenger flight's details.
- void joinFlight (Flight flight)

  Passenger books a flight.
- void setFlight (Flight flight)

  Defines a flight.
- String toString ()

  Defines as 'string' the parameters.

5.2.1 Detailed Description

This class defines the attributes of a flight.

Definition at line 34 of file Passenger.java.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 Passenger()

```
es.ull.passengers.Passenger.Passenger (
            String identifier,
            String name,
            String countryCode )
```

Passenger attends a flight.

Parameters

identifier	passenger identifier
name	passenger name
countryCode	passenger country code

Definition at line 55 of file Passenger.java.

5.2.3 Member Function Documentation

5.2.3.1 getCountryCode()

```
String es.ull.passengers.Passenger.getCountryCode ( )
```

Get passenger country code.

Returns

passenger's country code

Definition at line 85 of file Passenger.java.

5.2.3.2 getFlight()

```
Flight es.ull.passengers.Passenger.getFlight ( )
```

Get passenger flight's details.

Returns

passenger's flight

Definition at line 93 of file Passenger.java.

5.2.3.3 getIdentifier()

```
String es.ull.passengers.Passenger.getIdentifier ( )
```

Get passenger identifier.

Returns

passenger's identifier

Definition at line 69 of file Passenger.java.

5.2.3.4 getName()

```
String es.ull.passengers.Passenger.getName ( )
```

Get passenger name.

Returns

passenger's name

Definition at line 77 of file Passenger.java.

5.2.3.5 joinFlight()

```
void es.ull.passengers.Passenger.joinFlight (
             Flight flight )
```

Passenger books a flight.

Parameters

flight	flight's details

Definition at line 101 of file Passenger.java.

5.2.3.6 setFlight()

```
void es.ull.passengers.Passenger.setFlight (
            Flight flight )
```

Defines a flight.

Parameters

flight	flight's details

Definition at line 120 of file Passenger.java.

5.2.3.7 toString()

```
String es.ull.passengers.Passenger.toString ( )
```

Defines as 'string' the parameters.

Returns

passenger's details as 'string' information

Definition at line 129 of file Passenger.java.

The documentation for this class was generated from the following file:

- src/main/java/es/ull/passengers/Passenger.java

Chapter 6

File Documentation

6.1 src/main/java/es/ull/flights/Flight.java File Reference

Classes

- class es.ull.flights.Flight

  This class defines the attributes of a flight.

Packages

- package es.ull.flights

6.2 Flight.java

Go to the documentation of this file.

```java
/*
 * ========================================================================
 *
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements.  See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License.  You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 *
 * ========================================================================
 */
package es.ull.flights;

import java.util.HashSet;
import java.util.Set;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import es.ull.passengers.Passenger;

public class Flight {

    private String flightNumber;
    private int seats;
    private Set<Passenger> passengers = new HashSet<>();

    private static String flightNumberRegex = "^[A-Z]{2}\\d{3,4}$";
    private static Pattern pattern = Pattern.compile(flightNumberRegex);

    public Flight(String flightNumber, int seats) {
        Matcher matcher = pattern.matcher(flightNumber);
        if (!matcher.matches()) {
            throw new RuntimeException("Invalid flight number");
        }
        this.flightNumber = flightNumber;
        this.seats = seats;
    }

    public String getFlightNumber() {
        return flightNumber;
    }

    public int getNumberOfPassengers() {
        return passengers.size();
    }

    public boolean addPassenger(Passenger passenger) {
        if (getNumberOfPassengers() >= seats) {
            throw new RuntimeException("Not enough seats for flight " + getFlightNumber());
        }
        passenger.setFlight(this);
        return passengers.add(passenger);
    }

    public boolean removePassenger(Passenger passenger) {
        passenger.setFlight(null);
        return passengers.remove(passenger);
    }
}
```

6.3 src/main/java/es/ull/passengers/Passenger.java File Reference

Classes

- class es.ull.passengers.Passenger

  This class defines the attributes of a flight.

Packages

- package es.ull.passengers

6.4 Passenger.java

Go to the documentation of this file.

```
/*
 * ========================================================================
 *
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements.  See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License.  You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 *
 * ==========================================================================
 */
package es.ull.passengers;

import java.util.Arrays;
import java.util.Locale;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import es.ull.flights.Flight;

public class Passenger {
    private String identifier;
    private String name;
    private String countryCode;
    private Flight flight;

    public Passenger(String identifier, String name, String countryCode) {
        if (!Arrays.asList(Locale.getISOCountries()).contains(countryCode)) {
            throw new RuntimeException("Invalid country code");
        }

        this.identifier = identifier;
        this.name = name;
        this.countryCode = countryCode;
    }

    public String getIdentifier() {
        return identifier;
    }

    public String getName() {
        return name;
    }

    public String getCountryCode() {
        return countryCode;
    }

    public Flight getFlight() {
        return flight;
    }

    public void joinFlight(Flight flight) {
        Flight previousFlight = this.flight;
        if (null != previousFlight) {
            if (!previousFlight.removePassenger(this)) {
                throw new RuntimeException("Cannot remove passenger");
            }
        }
        setFlight(flight);
        if (null != flight) {
            if (!flight.addPassenger(this)) {
                throw new RuntimeException("Cannot add passenger");
            }
        }
    }

    public void setFlight(Flight flight) {
        this.flight = flight;
    }

    @Override
    public String toString() {
        return "Passenger " + getName() + " with identifier: " + getIdentifier() + " from " + getCountryCode();
    }
}
```

6.5 src/test/java/es/ull/flights/FlightTest.java File Reference

Classes

- class **es.ull.flights.FlightTest**

  This class defines different tests for the class Flight.

Packages

- package es.ull.flights

6.6 FlightTest.java

Go to the documentation of this file.

```java
package es.ull.flights;

import es.ull.passengers.Passenger;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.DisplayName;

import static org.junit.jupiter.api.Assertions.*;
import static org.junit.jupiter.api.Assertions.assertEquals;

@DisplayName("Flight Test")
class FlightTest {
    private Flight flight;
    private Flight flight2;
    private Passenger passenger;
    private Passenger passenger2;

    @BeforeEach
    void setUp(){
        flight = new Flight("FR4849", 20);
        flight2 = new Flight ("BC4624", 1);
        passenger = new Passenger ("id1", "Francesco", "IT");
        passenger2 = new Passenger ("id2", "Anna", "ES");
    }

    @Test
    @DisplayName("Add Passenger Test")
    void testAddPassenger() {
        assertAll("Verify that the passenger is correctly assigned to a flight",
                () -> assertTrue(flight.addPassenger(passenger), "Passenger is assigned to the flight"),
                () -> assertEquals(1, flight.getNumberOfPassengers(), "Number of passengers"),
                () -> assertTrue(flight2.addPassenger(passenger), "Adding a passenger"),
                () -> assertThrows(RuntimeException.class, ()-> flight2.addPassenger(passenger2), "Not possible to add more passengers")
        );
    }

    @Test
    @DisplayName("Get Flight Details")
    void testGetFlightDetails() {
        flight.addPassenger(passenger);
        assertAll("Verify the flight’s details",
                () -> assertEquals("FR4849", flight.getFlightNumber(), "Flight’s number"),
                () -> assertEquals(1, flight.getNumberOfPassengers(), "Number of passengers")
        );
    }

    @Test
    @DisplayName("Remove Passenger Test")
    void testRemovePassenger() {
        assertAll("Verify if it is possible to remove passenger",
                () -> assertTrue(flight2.addPassenger(passenger), "Added passenger to flight"),
                () -> assertTrue(flight2.removePassenger(passenger), "Removed passenger from the flight"),
                () -> assertFalse(flight2.removePassenger(passenger2), "The passenger is not on the flight and cannot be removed")
        );
    }
}
```

6.7 src/test/java/es/ull/passengers/PassengerTest.java File Reference

Classes

- class **es.ull.passengers.PassengerTest**

  This class defines different tests for the class Passenger.

Packages

- package es.ull.passengers

6.8 PassengerTest.java

Go to the documentation of this file.

```
package es.ull.passengers;

import es.ull.flights.Flight;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.DisplayName;
import static org.junit.jupiter.api.Assertions.*;

class PassengerTest {
    private Passenger passenger;
    private Flight flight;
    private Flight emptyFlight;

    @BeforeEach
    void setUp() {
        passenger = new Passenger("id1", "Francesco", "IT");
        flight = new Flight("FR4849", 12);
        emptyFlight = new Flight("LB2457", 0);
    }

    @Test
    @DisplayName("Get Passenger’s Country Code")
    void testGetCountryCode() {
        assertEquals("IT", passenger.getCountryCode(), "Passenger’s country code");
    }

    @Test
    @DisplayName("Get Passenger’s Identifier ")
    void testGetIdentifier() {
        assertEquals("id1", passenger.getIdentifier(), "Passenger’s identifier");
    }

    @Test
    @DisplayName("Get Passenger’s Name")
    void testGetName() {
        assertEquals("Francesco", passenger.getName(), "Passenger’s name");
    }

    @Test
    @DisplayName("Passenger’s toString")
    void testToString() {
        assertEquals("Passenger Francesco with identifier: id1 from IT" , passenger.toString());
    }

    @Test
    @DisplayName("Passenger joining a flight")
    void testJoinFlight() {
        assertAll("Verify is passenger is assigned correctly to a flight",
                () -> assertDoesNotThrow(()-> passenger.joinFlight(flight), "We can join a flight"),
                () -> assertEquals(flight, passenger.getFlight(), "The flight is set correctly"),
                () -> assertThrows(RuntimeException.class, () -> passenger.joinFlight(emptyFlight), "We cannot join a flight")
        );
    }
}
```

Index