

Airport - Practica 06 LDH

Generated by Doxygen 1.9.2



<b>1 Namespace Index</b>	<b>1</b>
1.1 Packages	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 File Index</b>	<b>5</b>
3.1 File List	5
<b>4 Namespace Documentation</b>	<b>7</b>
4.1 Package es.ull.flights	7
4.2 Package es.ull.passengers	7
<b>5 Class Documentation</b>	<b>9</b>
5.1 es.ull.flights.Flight Class Reference	9
5.1.1 Detailed Description	9
5.1.2 Constructor & Destructor Documentation	9
5.1.2.1 Flight()	9
5.1.3 Member Function Documentation	10
5.1.3.1 addPassenger()	10
5.1.3.2 getFlightNumber()	10
5.1.3.3 getNumberOfPassengers()	11
5.1.3.4 removePassenger()	11
5.2 es.ull.passengers.Passenger Class Reference	11
5.2.1 Detailed Description	12
5.2.2 Constructor & Destructor Documentation	12
5.2.2.1 Passenger()	12
5.2.3 Member Function Documentation	12
5.2.3.1 getCountryCode()	12
5.2.3.2 getFlight()	13
5.2.3.3 getIdentifier()	13
5.2.3.4 getName()	13
5.2.3.5 joinFlight()	13
5.2.3.6 setFlight()	14
5.2.3.7 toString()	14
<b>6 File Documentation</b>	<b>15</b>
6.1 src/main/java/es/ull/flights/Flight.java File Reference	15
6.2 Flight.java	15
6.3 src/main/java/es/ull/passengers/Passenger.java File Reference	16
6.4 Passenger.java	16
6.5 src/test/java/es/ull/flights/FlightTest.java File Reference	17
6.6 FlightTest.java	18
6.7 src/test/java/es/ull/passengers/PassengerTest.java File Reference	18

6.8 PassengerTest.java . . . . .	19
<b>Index</b>	<b>21</b>

# Chapter 1

## Namespace Index

### 1.1 Packages

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

<a href="#">es.ull.flights</a>	7
<a href="#">es.ull.passengers</a>	7



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">es.ull.flights.Flight</a>	
This class defines the attributes of a flight	9
<a href="#">es.ull.passengers.Passenger</a>	
This class defines the attributes of a flight	11





## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

src/main/java/es/ull/flights/ <a href="#">Flight.java</a> . . . . .	15
src/main/java/es/ull/passengers/ <a href="#">Passenger.java</a> . . . . .	16
src/test/java/es/ull/flights/ <a href="#">FlightTest.java</a> . . . . .	17
src/test/java/es/ull/passengers/ <a href="#">PassengerTest.java</a> . . . . .	18



## 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**

<i>flightNumber</i>	flight number
<i>seats</i>	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**

<i>passenger</i>	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

<i>passenger</i>	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

<i>identifier</i>	passenger identifier
<i>name</i>	passenger name
<i>countryCode</i>	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**

<i>flight</i>	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**

<i>flight</i>	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.](#)

```
00001 /*
00002  * =====
00003  *
00004  * Licensed to the Apache Software Foundation (ASF) under one or more
00005  * contributor license agreements. See the NOTICE file distributed with
00006  * this work for additional information regarding copyright ownership.
00007  * The ASF licenses this file to You under the Apache License, Version 2.0
00008  * (the "License"); you may not use this file except in compliance with
00009  * the License. You may obtain a copy of the License at
00010  *
00011  * http://www.apache.org/licenses/LICENSE-2.0
00012  *
00013  * Unless required by applicable law or agreed to in writing, software
00014  * distributed under the License is distributed on an "AS IS" BASIS,
00015  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00016  * See the License for the specific language governing permissions and
00017  * limitations under the License.
00018  *
00019  * =====
00020  */
00021 package es.ull.flights;
00022
00023 import java.util.HashSet;
00024 import java.util.Set;
00025 import java.util.regex.Matcher;
00026 import java.util.regex.Pattern;
00027
00028 import es.ull.passengers.Passenger;
00029
00034 public class Flight {
00035
00044     private String flightNumber;
```

```

00045     private int seats;
00046     private Set<Passenger> passengers = new HashSet<>();
00047
00048     private static String flightNumberRegex = "[A-Z]{2}\\d{3,4}$";
00049     private static Pattern pattern = Pattern.compile(flightNumberRegex);
00050
00056     public Flight(String flightNumber, int seats) {
00057         Matcher matcher = pattern.matcher(flightNumber);
00058         if (!matcher.matches()) {
00059             throw new RuntimeException("Invalid flight number");
00060         }
00061         this.flightNumber = flightNumber;
00062         this.seats = seats;
00063     }
00064
00069     public String getFlightNumber() {
00070         return flightNumber;
00071     }
00072
00077     public int getNumberOfPassengers() {
00078         return passengers.size();
00079     }
00080
00086     public boolean addPassenger(Passenger passenger) {
00087         if (getNumberOfPassengers() >= seats) {
00088             throw new RuntimeException("Not enough seats for flight " + getFlightNumber());
00089         }
00090         passenger.setFlight(this);
00091         return passengers.add(passenger);
00092     }
00093
00099     public boolean removePassenger(Passenger passenger) {
00100         passenger.setFlight(null);
00101         return passengers.remove(passenger);
00102     }
00103 }

```

## 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.](#)

```

00001  /*
00002   * =====
00003   *
00004   * Licensed to the Apache Software Foundation (ASF) under one or more
00005   * contributor license agreements. See the NOTICE file distributed with
00006   * this work for additional information regarding copyright ownership.
00007   * The ASF licenses this file to You under the Apache License, Version 2.0
00008   * (the "License"); you may not use this file except in compliance with
00009   * the License. You may obtain a copy of the License at
00010   *
00011   * http://www.apache.org/licenses/LICENSE-2.0
00012   *
00013   * Unless required by applicable law or agreed to in writing, software
00014   * distributed under the License is distributed on an "AS IS" BASIS,
00015   * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00016   * See the license for the specific language governing permissions and
00017   * limitations under the License.
00018   *

```

```

00019  * =====
00020  */
00021  package es.ull.passengers;
00022
00023  import java.util.Arrays;
00024  import java.util.Locale;
00025  import java.util.regex.Matcher;
00026  import java.util.regex.Pattern;
00027
00028  import es.ull.flights.Flight;
00029
00034  public class Passenger {
00044      private String identifier;
00045      private String name;
00046      private String countryCode;
00047      private Flight flight;
00048
00055      public Passenger(String identifier, String name, String countryCode) {
00056          if (!Arrays.asList(Locale.getISOCountries()).contains(countryCode)) {
00057              throw new RuntimeException("Invalid country code");
00058          }
00059
00060          this.identifier = identifier;
00061          this.name = name;
00062          this.countryCode = countryCode;
00063      }
00064
00069      public String getIdentifier() {
00070          return identifier;
00071      }
00072
00077      public String getName() {
00078          return name;
00079      }
00080
00085      public String getCountryCode() {
00086          return countryCode;
00087      }
00088
00093      public Flight getFlight() {
00094          return flight;
00095      }
00096
00101      public void joinFlight(Flight flight) {
00102          Flight previousFlight = this.flight;
00103          if (null != previousFlight) {
00104              if (!previousFlight.removePassenger(this)) {
00105                  throw new RuntimeException("Cannot remove passenger");
00106              }
00107          }
00108          setFlight(flight);
00109          if (null != flight) {
00110              if (!flight.addPassenger(this)) {
00111                  throw new RuntimeException("Cannot add passenger");
00112              }
00113          }
00114      }
00115
00120      public void setFlight(Flight flight) {
00121          this.flight = flight;
00122      }
00123
00128      @Override
00129      public String toString() {
00130          return "Passenger " + getName() + " with identifier: " + getIdentifier() + " from " +
getCountryCode();
00131      }
00132  }

```

## 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.](#)

```

00001 package es.ull.flights;
00002
00003 import es.ull.passengers.Passenger;
00004 import org.junit.jupiter.api.Test;
00005 import org.junit.jupiter.api.BeforeEach;
00006 import org.junit.jupiter.api.DisplayName;
00007
00008 import static org.junit.jupiter.api.Assertions.*;
00009 import static org.junit.jupiter.api.Assertions.assertEquals;
00010
00014 @DisplayName("Flight Test")
00015 class FlightTest {
00023     private Flight flight;
00024     private Flight flight2;
00025     private Passenger passenger;
00026     private Passenger passenger2;
00027
00028     @BeforeEach
00029     void setUp(){
00030         flight = new Flight("FR4849", 20);
00031         flight2 = new Flight ("BC4624", 1);
00032         passenger = new Passenger ("idl", "Francesco", "IT");
00033         passenger2 = new Passenger ("id2", "Anna", "ES");
00034     }
00035
00039     @Test
00040     @DisplayName("Add Passenger Test")
00041     void testAddPassenger() {
00042         assertAll("Verify that the passenger is correctly assigned to a flight",
00043             () -> assertTrue(flight.addPassenger(passenger), "Passenger is assigned to the
flight"),
00044             () -> assertEquals(1, flight.getNumberOfPassengers(), "Number of passengers"),
00045             () -> assertTrue(flight2.addPassenger(passenger), "Adding a passenger"),
00046             () -> assertThrows(RuntimeException.class, ()-> flight2.addPassenger(passenger2), "Not
possible to add more passengers")
00047         );
00048     }
00049
00053     @Test
00054     @DisplayName("Get Flight Details")
00055     void testGetFlightDetails() {
00056         flight.addPassenger(passenger);
00057         assertAll("Verify the flight's details",
00058             () -> assertEquals("FR4849", flight.getFlightNumber(), "Flight's number"),
00059             () -> assertEquals(1, flight.getNumberOfPassengers(), "Number of passengers")
00060         );
00061     }
00062
00066     @Test
00067     @DisplayName("Remove Passenger Test")
00068     void testRemovePassenger() {
00069         assertAll("Verify if it is possible to remove passenger",
00070             () -> assertTrue(flight2.addPassenger(passenger), "Added passenger to flight"),
00071             () -> assertTrue(flight2.removePassenger(passenger), "Removed passenger from the
flight"),
00072             () -> assertFalse(flight2.removePassenger(passenger2), "The passenger is not on the
flight and cannot be removed")
00073         );
00074     }
00075 }

```

## 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.](#)

```

00001 package es.ull.passengers;
00002
00003 import es.ull.flights.Flight;
00004 import org.junit.jupiter.api.Test;
00005 import org.junit.jupiter.api.BeforeEach;
00006 import org.junit.jupiter.api.DisplayName;
00007 import static org.junit.jupiter.api.Assertions.*;
00008
00013 class PassengerTest {
00021     private Passenger passenger;
00022     private Flight flight;
00023     private Flight emptyFlight;
00024
00025     @BeforeEach
00026     void setUp() {
00027         passenger = new Passenger("id1", "Francesco", "IT");
00028         flight = new Flight("FR4849", 12);
00029         emptyFlight = new Flight("LB2457", 0);
00030     }
00031
00035     @Test
00036     @DisplayName("Get Passenger's Country Code")
00037     void testGetCountryCode() {
00038         assertEquals("IT", passenger.getCountryCode(), "Passenger's country code");
00039     }
00040
00044     @Test
00045     @DisplayName("Get Passenger's Identifier ")
00046     void testGetIdentifier() {
00047         assertEquals("id1", passenger.getIdentifier(), "Passenger's identifier");
00048     }
00049
00053     @Test
00054     @DisplayName("Get Passenger's Name")
00055     void testGetName() {
00056         assertEquals("Francesco", passenger.getName(), "Passenger's name");
00057     }
00058
00062     @Test
00063     @DisplayName("Passenger's toString")
00064     void testToString() {
00065         assertEquals("Passenger Francesco with identifier: id1 from IT" , passenger.toString());
00066     }
00067
00071     @Test
00072     @DisplayName("Passenger joining a flight")
00073     void testJoinFlight() {
00074         assertAll("Verify is passenger is assigned correctly to a flight",
00075             () -> assertDoesNotThrow(()-> passenger.joinFlight(flight), "We can join a flight"),
00076             () -> assertEquals(flight, passenger.getFlight(), "The flight is set correctly"),
00077             () -> assertThrows(RuntimeException.class, () -> passenger.joinFlight(emptyFlight),
00078                 "We cannot join a flight")
00079         );
00080 }

```





# Index

- addPassenger
  - es.ull.flights.Flight, 10
- es.ull.flights, 7
- es.ull.flights.Flight, 9
  - addPassenger, 10
  - Flight, 9
  - getFlightNumber, 10
  - getNumberOfPassengers, 10
  - removePassenger, 11
- es.ull.passengers, 7
- es.ull.passengers.Passenger, 11
  - getCountryCode, 12
  - getFlight, 13
  - getIdentifier, 13
  - getName, 13
  - joinFlight, 13
  - Passenger, 12
  - setFlight, 14
  - toString, 14
- Flight
  - es.ull.flights.Flight, 9
- getCountryCode
  - es.ull.passengers.Passenger, 12
- getFlight
  - es.ull.passengers.Passenger, 13
- getFlightNumber
  - es.ull.flights.Flight, 10
- getIdentifier
  - es.ull.passengers.Passenger, 13
- getName
  - es.ull.passengers.Passenger, 13
- getNumberOfPassengers
  - es.ull.flights.Flight, 10
- joinFlight
  - es.ull.passengers.Passenger, 13
- Passenger
  - es.ull.passengers.Passenger, 12
- removePassenger
  - es.ull.flights.Flight, 11
- setFlight
  - es.ull.passengers.Passenger, 14
- src/main/java/es/ull/flights/Flight.java, 15
- src/main/java/es/ull/passengers/Passenger.java, 16
- src/test/java/es/ull/flights/FlightTest.java, 17, 18
- src/test/java/es/ull/passengers/PassengerTest.java, 18, 19
- toString
  - es.ull.passengers.Passenger, 14