

Tree: 8ca21e6d2a ▾

airport_challenge / lib / airport.rb

Find file

Copy path

Fetching contributors...

 Cannot retrieve contributors at this time

36 lines (28 sloc) | 483 Bytes

```
1  class Airport
2
3      def initialize
4          @planes = []
5          @max_capacity = 3
6      end
7
8      def add_plane(plane)
9          if full?
10             'Airport is full'
11         else
12             @planes << plane unless full?
13             'Plane is landed'
14         end
15     end
16
17     def remove_plane
18         @planes.shift
19         'Plane has left airport'
20     end
21
22     def empty?
23         @planes.empty?
24     end
25
26     def change_max_capacity(max_capacity)
27         @max_capacity = max_capacity
28     end
29
30     private
31
32     def full?
33         @planes.length == @max_capacity
34     end
35 end
```

Tree: 8ca21e6d2a ▾

airport_challenge / lib / plane.rb

Find file

Copy path

Fetching contributors...

 Cannot retrieve contributors at this time

30 lines (22 sloc) | 372 Bytes

```
1 class Plane
2   attr_accessor :status
3   attr_accessor :current_airport
4
5   def initialize
6     @status = "landed"
7   end
8
9   def fly
10    self.status = "flying"
11  end
12
13  def land
14    self.status = "landed"
15  end
16
17  def flying?
18    status == "flying"
19  end
20
21  def landed?
22    status == "landed"
23  end
24
25  def assigned_airport(airport)
26    self.current_airport = airport
27  end
28
29 end
```

Tree: 8ca21e6d2a ▼

airport_challenge / [lib](#) / **startup.rb**

Find file

Copy path

Fetching contributors...



Cannot retrieve contributors at this time

5 lines (4 sloc) | 81 Bytes

```
1  require 'airport'
2  require 'plane'
3  require 'traffic_controller'
4  require 'weather'
```

Fetching contributors...

 Cannot retrieve contributors at this time

36 lines (29 sloc) | 791 Bytes

```
1  require "weather"
2  require "airport"
3
4  class TrafficController
5      def land_to(plane, airport)
6          if weather.stormy?
7              "It is too windy and stormy out there"
8          elsif plane.landed?
9              "This plane is already landed here"
10         else
11             plane.land
12             airport.add_plane(plane)
13         end
14     end
15
16     def takeoff_from(plane, airport)
17         return "It is too windy and stormy out there" if weather.stormy?
18         return "The plane has already left the airport" if plane.flying?
19         return "The airport is empty" if airport.empty?
20         return "The plane is not in this airport" if airport != plane.current_airport
21
22         plane.fly
23         airport.remove_plane
24     end
25
26     def assign_plane_to_airport(plane, airport)
27         plane.assigned_airport(airport)
28     end
29
30     private
31
32     def weather
33         Weather.new
34     end
35 end
```

Tree: 8ca21e6d2a ▼

airport_challenge / lib / weather.rb

Find file

Copy path

Fetching contributors...



Cannot retrieve contributors at this time

12 lines (9 sloc) | 129 Bytes

```
1  class Weather
2    attr_reader :state
3
4    def initialize
5      @state = [*1..100].sample
6    end
7
8    def stormy?
9      state > 90
10   end
11 end
```

Tree8ca21e6d2a ▾

airport_challenge / spec / airport_spec.rb

Find file

Copy path

Fetching contributors...

 Cannot retrieve contributors at this time

35 lines (27 sloc) | 707 Bytes

```
1  require 'airport'
2  require 'plane'
3
4  describe Airport do
5    it "is not full" do
6      airport = Airport.new
7      plane = Plane.new
8
9      expect(airport.add_plane(plane)).to eq('Plane is landed')
10   end
11
12   it "is full" do
13     airport = Airport.new
14     plane = Plane.new
15     3.times { airport.add_plane(plane) }
16
17     expect(airport.add_plane(plane)).to eq('Airport is full')
18   end
19
20   it "send a plane away" do
21     airport = Airport.new
22     expect(airport.remove_plane).to eq('Plane has left airport')
23   end
24
25   it "is empty" do
26     airport = Airport.new
27     expect(airport.empty?).to eq(true)
28   end
29
30   it "change max capacity" do
31     airport = Airport.new
32     expect(airport.change_max_capacity(5)).to eq(5)
33   end
34 end
```

Tree: 8ca21e6d2a ▾

airport_challenge / spec / plane_spec.rb

Find file

Copy path


Fetching contributors...

 Cannot retrieve contributors at this time

34 lines (24 sloc) | 571 Bytes

```
1  require 'plane'
2
3  describe Plane do
4    let(:plane) { Plane.new }
5
6    it "report left airport" do
7      expect(plane.fly).to eq("flying")
8    end
9
10   it "report landed to airport" do
11     expect(plane.land).to eq("landed")
12   end
13
14   it "default state of a plane" do
15     expect(plane.status).to eq("landed")
16   end
17
18   it "is flying" do
19     plane.fly
20
21     expect(plane.flying?).to eq(true)
22   end
23
24   it "is landed" do
25     expect(plane.landed?).to eq(true)
26   end
27
28   it "assigned airport" do
29     airport = Airport.new
30
31     expect(plane.assigned_airport(airport)).to eq(airport)
32   end
33 end
```

Fetching contributors...

 Cannot retrieve contributors at this time

77 lines (54 sloc) | 2.24 KB

```
1  require 'traffic_controller'
2  require 'plane'
3  require 'airport'
4
5  describe TrafficController do
6    let(:plane) { Plane.new }
7    let(:airport) { Airport.new }
8
9    it "land plane to an airport" do
10      plane.fly
11      weather = double(:sunny_weather, stormy?: false )
12
13      allow(subject).to receive(:weather).and_return(weather)
14
15      expect(subject.land_to(plane, airport)).to eq("Plane is landed")
16    end
17
18    it "prevent landing when weather is stormy" do
19      weather = double(:stormy_weather, stormy?: true)
20
21      allow(subject).to receive(:weather).and_return(weather)
22
23      expect(subject.land_to(plane, airport)).to eq("It is too windy and stormy out there")
24    end
25
26    it "take off plane from an airport and confirm plane left airport" do
27      airport.add_plane(plane)
28      plane.assigned_airport(airport)
29      weather = double(:sunny_weather, stormy?: false)
30
31      allow(subject).to receive(:weather).and_return(weather)
32
33      expect(subject.takeoff_from(plane,airport)).to eq('Plane has left airport')
34    end
35
36    it "prevent take off when weather stormy" do
37      airport.add_plane(plane)
38      plane.assigned_airport(airport)
39      weather = double(:stormy_weather, stormy?: true)
40
41      allow(subject).to receive(:weather).and_return(weather)
42
43      expect(subject.takeoff_from(plane, airport)).to eq('It is too windy and stormy out there')
44    end
45
46    it "prevent landing when airport is full" do
47      plane.fly
48      airport = double(:full_airport, add_plane: 'Airport is full')
49      weather = double(:sunny_weather, stormy?: false)
50
51      allow(subject).to receive(:full?).and_return(airport)
52      allow(subject).to receive(:weather).and_return(weather)
53
54      expect(subject.land_to(plane, airport)).to eq('Airport is full')
55    end
56
57    it "prevent to take off if airport is empty" do
58      plane.assigned_airport(airport)
59      weather = double(:sunny_weather, stormy?: false)
60
61      allow(subject).to receive(:weather).and_return(weather)
62
63      expect(subject.takeoff_from(plane, airport)).to eq('The airport is empty')
64    end
65
66    it "assign plane to airport" do
67      plane.fly
68      expect(subject.assign_plane_to_airport(plane, airport)).to eq(airport)
69    end
70
71    it "take off from different airport" do
72      airport.add_plane(plane)
73
74      expect(subject.takeoff_from(plane, airport)).to eq("The plane is not in this airport")
75    end
76  end
```


Tree: 8ca21e6d2a ▾

airport_challenge / spec / weather_spec.rb

Find file

Copy path

Fetching contributors...

 Cannot retrieve contributors at this time

18 lines (13 sloc) | 336 Bytes

1 require 'weather'

2

3 describe Weather do

4 it "is stormy" do

5 weather = Weather.new

6 allow(weather).to receive(:state).and_return(94)

7

8 expect(weather.stormy?).to eq(true)

9 end

10

11 it "is sunny" do

12 weather = Weather.new

13 allow(weather).to receive(:state).and_return(44)

14

15 expect(weather.stormy?).to eq(false)

16 end

17 end