

Tree: 25cc23fb11 ▾

airport_challenge / lib / airport.rb

Find file

Copy path

Fetching contributors...

 Cannot retrieve contributors at this time

39 lines (31 sloc) | 828 Bytes

```
1  require_relative 'weather'
2  require_relative 'plane'
3
4  class Airport
5    DEFAULT_CAPACITY = 100
6    attr_reader :planes, :capacity, :weather
7
8    def initialize(capacity = DEFAULT_CAPACITY, weather)
9      @planes = []
10     @capacity = capacity
11     @weather = weather
12   end
13
14   def land(plane)
15     raise "Airport full" if full?
16     raise "Plane has already landed in the airport" if exist?(plane)
17     raise "It is too stormy to land" if @weather.stormy?
18     plane.land
19     @planes << plane
20   end
21
22   def take_off(plane)
23     raise "Plane can't take off if it has not landed at the airport" unless exist?(plane)
24     raise "It is too stormy to take off" if @weather.stormy?
25     plane.take_off
26     @planes.delete(plane)
27   end
28
29   private
30   def full?
31     @planes.length >= capacity
32   end
33
34   def exist?(plane)
35     @planes.include?(plane)
36   end
37
38   end
```

Tree: 25cc23fb11 ▾

airport_challenge / lib / plane.rb

Find file

Copy path

Fetching contributors...

 Cannot retrieve contributors at this time

19 lines (13 sloc) | 189 Bytes

1 require_relative 'airport'

2

3 class Plane

4 attr_reader :flying

5

6 def initialize

7 @flying = true

8 end

9

10 def land

11 @flying = false

12 end

13

14 def take_off

15 @flying = true

16 end

17

18 end

Tree: 25cc23fb11 ▼

airport_challenge / lib / weather.rb

Find file

Copy path

Fetching contributors...




Cannot retrieve contributors at this time

15 lines (10 sloc) | 169 Bytes

```
1  class Weather
2
3      def stormy?
4          outlook == :stormy
5      end
6
7      private
8
9      def outlook
10         random_number = rand(101)
11         random_number == 100 ? :stormy : :sunny
12     end
13
14 end
```

Fetching contributors...

 Cannot retrieve contributors at this time

81 lines (67 sloc) | 2.42 KB

```
1  require 'airport'
2  require 'plane'
3  require 'weather'
4
5  describe Airport do
6    let(:plane) { double(:plane, take_off: nil, land: nil) }
7    let(:weather) { double(:weather, stormy?: false) }
8    subject(:airport) { described_class.new weather }
9
10   it 'has a default capacity' do
11     expect(subject.capacity).to eq(Airport::DEFAULT_CAPACITY)
12   end
13
14   describe "#land" do
15     before do
16       allow(plane).to receive(:land)
17     end
18
19     it 'lands a plane in the airport' do
20       airport.land(plane)
21     end
22
23     it 'has the plane after it has landed' do
24       airport.land(plane)
25       expect(airport.planes).to include plane
26     end
27
28     it 'does not allow plane already landed to land again' do
29       airport.land(plane)
30       expect { airport.land(plane) }.to raise_error "Plane has already landed in the airport"
31     end
32
33     it 'raise an error when the airport is full' do
34       allow(weather).to receive(:stormy?).and_return(false)
35       airport.capacity.times do
36         plane = double(:plane)
37         allow(plane).to receive(:land).and_return(false)
38         airport.land(plane)
39       end
40       expect { airport.land(plane) }.to raise_error "Airport full"
41     end
42
43     it 'will prevent landing when the weather is stormy' do
44       allow(weather).to receive(:stormy?).and_return(true)
45       expect { airport.land(plane) }.to raise_error "It is too stormy to land"
46     end
47   end
48
49   describe "#take_off" do
50     before do
51       allow(plane).to receive(:land).and_return(false)
52       allow(plane).to receive(:take_off).and_return(true)
53     end
54
55     it 'instructs the plane to take off' do
56       allow(weather).to receive(:stormy?).and_return(false)
57       airport.land(plane)
58       airport.take_off(plane)
59     end
60
61     it 'confirms that the plane taken off is no longer at the airport' do
62       allow(weather).to receive(:stormy?).and_return(false)
63       airport.land(plane)
64       airport.take_off(plane)
65       expect(airport.planes).to eq []
66     end
67
68     it 'can only allow planes already landed to take off' do
69       allow(weather).to receive(:stormy?).and_return(false)
70       expect { airport.take_off(plane) }.to raise_error "Plane can't take off if it has not landed at the airport"
71     end
72
73     it 'will prevent taking off when the weather is stormy' do
74       allow(plane).to receive(:land)
75       airport.land(plane)
76       allow(weather).to receive(:stormy?).and_return(true)
77       expect { airport.take_off(plane) }.to raise_error "It is too stormy to take off"
78     end
79   end
80 end
```

Tree: 25cc23fb11 ▾

airport_challenge / spec / plane_spec.rb

Find file

Copy path

Fetching contributors...

 Cannot retrieve contributors at this time

27 lines (22 sloc) | 708 Bytes

```
1  require 'airport'
2  require 'plane'
3
4  describe Plane do
5    let(:airport) { double :airport }
6    subject(:plane) { described_class.new }
7
8    describe "#landed" do
9      it 'has landed at the airport and should not be flying' do
10        allow(airport).to receive(:land).and_return(plane.land)
11        airport.land(plane)
12        expect(plane.flying).to eq false
13      end
14    end
15
16    describe "#taken off" do
17      it 'has taken off and confirms that it is no longer in the airport' do
18        allow(airport).to receive(:land).and_return(plane.land)
19        allow(airport).to receive(:take_off).and_return(plane.take_off)
20        airport.land(plane)
21        airport.take_off(plane)
22        expect(plane.flying).to eq true
23      end
24    end
25
26  end
```

Tree: 25cc23fb11 ▾

airport_challenge / spec / weather_spec.rb

Find file

Copy path

Fetching contributors...



Cannot retrieve contributors at this time

12 lines (9 sloc) | 220 Bytes

```
1  require 'weather'
2
3  describe Weather do
4    subject(:weather) { described_class.new }
5
6    describe '#initialization' do
7      it 'generate a weather' do
8        expect(weather.stormy?).to eq(true).or eq(false)
9      end
10   end
11 end
```