

Tree: 3a36c0fd08 ▼

airport_challenge / .rspec

Find file

Copy path

Fetching contributors...

 Cannot retrieve contributors at this time

3 lines (2 sloc) | 53 Bytes

```
1 --color --format documentation
2 --require spec_helper
```

Tree: 3a36c0fd08 ▾

airport_challenge / lib / airport.rb

Find file

Copy path

Fetching contributors...

 Cannot retrieve contributors at this time

41 lines (34 sloc) | 970 Bytes

```
1  require_relative 'weather.rb'
2
3  class Airport
4    DEFAULT_CAPACITY = 20
5    attr_reader :planes, :weather, :max_capacity
6    # Initializing with hash removes argument order dependency
7    def initialize args
8      @planes = []
9      @weather = args[:weather]
10     @max_capacity = (args[:max_capacity] || DEFAULT_CAPACITY)
11 end
12
13 def depart(plane)
14   raise "Plane not found at this Airport" unless find_plane(plane)
15   raise "Unsuitable conditions for takeoff" if weather.stormy?
16   plane.takeoff
17   remove_plane(plane)
18 end
19
20 def receive(plane)
21   raise "Unsuitable conditions for landing" if weather.stormy?
22   raise "Airport full, unable to receive plane" if planes.length >= max_capacity
23   plane.land(self)
24   add_plane(plane)
25 end
26
27 def find_plane(plane)
28   planes.include? plane
29 end
30
31 private
32 def add_plane(plane)
33   planes << plane
34 end
35
36 def remove_plane(plane)
37   planes.delete(plane)
38   raise "Plane has not departed" if find_plane(plane)
39 end
40 end
```

Tree: 3a36c0fd08 ▾

airport_challenge / lib / plane.rb

Find file

Copy path

Fetching contributors...

 Cannot retrieve contributors at this time

18 lines (15 sloc) | 306 Bytes

1 class Plane
2 attr_accessor :grounded, :airport
3 def initialize
4 @grounded = true
5 end
6
7 def land(airport)
8 raise "Already grounded" if grounded
9 self.grounded = true
10 self.airport = airport
11 end
12
13 def takeoff
14 raise "Already flying" unless grounded
15 self.grounded = false
16 end
17 end

Tree: 3a36c0fd08 ▼

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

Find file

Copy path

Fetching contributors...



Cannot retrieve contributors at this time

11 lines (9 sloc) | 165 Bytes

```
1  class Weather
2    def stormy?
3      record_weather == "stormy"
4    end
5
6    private
7    def record_weather
8      ["stormy", "clear", "clear", "clear", "clear"].sample
9    end
10 end
```

Tree: 3a36c0fd08 ▾

airport_challenge / spec / airport_spec.rb

Find file

Copy path

Fetching contributors...

 Cannot retrieve contributors at this time

47 lines (44 sloc) | 1.68 KB

```
1  require 'airport'
2
3  describe Airport do
4    let(:plane) { double(:plane, takeoff: nil, land: nil) }
5    let(:weather) { double(:weather, stormy?: false) }
6    subject(:airport) { described_class.new weather: weather }
7
8    it 'has a default capacity' do
9      expect(subject.max_capacity).to eq(Airport::DEFAULT_CAPACITY)
10   end
11   it 'has a variable capacity' do
12     expect(Airport.new(weather: weather, max_capacity: 30).max_capacity).to eq(30)
13   end
14   describe '#depart' do
15     it 'can depart a plane' do
16       expect(subject).to respond_to(:depart).with(1).argument
17     end
18     it 'confirms plane has departed' do
19       subject.receive(plane)
20       subject.depart(plane)
21       expect(subject.find_plane(plane)).to be false
22     end
23     it 'does not depart planes not found' do
24       expect { subject.depart(plane) }.to raise_error(RuntimeError, "Plane not found at this Airport")
25     end
26     it 'does not depart planes in stormy weather' do
27       subject.receive(plane)
28       allow(weather).to receive(:stormy?).and_return(true)
29       expect { subject.depart(plane) }.to raise_error(RuntimeError, "Unsuitable conditions for takeoff")
30     end
31     it 'does not receive planes when full' do
32       20.times { subject.receive(plane) }
33       expect { subject.receive(plane) }.to raise_error(RuntimeError, "Airport full, unable to receive plane")
34     end
35   end
36   describe '#receive' do
37     it 'does not allow plane to land in stormy weather' do
38       allow(weather).to receive(:stormy?).and_return(true)
39       expect { subject.receive(plane) }.to raise_error(RuntimeError, "Unsuitable conditions for landing")
40     end
41     it 'can receive a plane' do
42       subject.receive(plane)
43       expect(subject.planes.length).to eq(1)
44     end
45   end
46 end
```

Fetching contributors...

 Cannot retrieve contributors at this time

29 lines (27 sloc) | 763 Bytes

```
1  require 'plane'
2
3  describe Plane do
4    let(:airport) { double(:airport, receive: nil) }
5    describe '#land' do
6      it 'Can land' do
7        expect(subject).to respond_to(:land).with(1).argument
8      end
9      it 'can not land when grounded' do
10       expect { subject.land(airport) }.to raise_error(RuntimeError, "Already grounded")
11     end
12     it 'lands at the correct airport' do
13       subject.takeoff
14       subject.land(airport)
15       expect(subject.airport).to eq airport
16     end
17   end
18   describe '#takeoff' do
19     it 'Can takeoff' do
20       subject.takeoff
21       expect(subject.grounded).to be false
22     end
23     it 'can not takeoff when flying' do
24       subject.takeoff
25       expect { subject.takeoff }.to raise_error(RuntimeError, "Already flying")
26     end
27   end
28 end
```

Tree: 3a36c0fd08 ▼

airport_challenge / spec / weather_spec.rb

Find file

Copy path

Fetching contributors...



Cannot retrieve contributors at this time

8 lines (6 sloc) | 155 Bytes

```
1  require 'weather'
2
3  describe Weather do
4    it 'returns true or false when asked if stormy' do
5      expect(subject.stormy?).to be(true).or be(false)
6    end
7  end
```