41 lines (31 sloc) | 789 Bytes

```ruby
require_relative "plane"
require_relative "weather"

class Airport

  attr_reader :planes, :capacity, :weather
  DEFAULT_CAPACITY = 10

  def initialize(capacity = DEFAULT_CAPACITY, weather = Weather.new)
    @planes = []
    @capacity = capacity
    @weather = weather
  end

  def land(plane)
    fail "Airport full" if full?
    fail "Plane is already landed" if plane.landed == true
    fail "Bad weather – unable to land" if stormy?
    @planes << plane
    plane.land
  end

  def take_off(plane)
    fail "Plane not in airport" if !@planes.include?(plane)
    fail "Bad weather – unable to take off" if stormy?
    @planes = @planes - [plane]
    plane.take_off
  end

  private

  def full?
    @planes.length >= capacity
  end

  def stormy?
    @weather.stormy? == true
  end

end
```

Tree: **ea01fb49c7** ▾   **airport_challenge** / lib / **plane.rb**

Find file   Copy path

Fetching contributors…

Cannot retrieve contributors at this time

14 lines (9 sloc) | 119 Bytes

```ruby
class Plane

  attr_reader :landed

  def land
    @landed = true
  end

  def take_off
    @landed = false
  end

  end
```

Tree: **ea01fb49c7** ▾   **airport_challenge** / lib / **weather.rb**

Find file   Copy path

Fetching contributors…

Cannot retrieve contributors at this time

16 lines (12 sloc) | 146 Bytes

```ruby
class Weather

  def stormy?
    @stormy = random
  end

  def random
    if rand(4).to_i == 1
      true
    else
      false
    end
  end

end
```

Tree: ea01fb49c7 ▾   **airport_challenge** / spec / **airport_spec.rb**   Find file   Copy path

Fetching contributors…

Cannot retrieve contributors at this time

64 lines (51 sloc) | 1.86 KB

```ruby
require "airport"

describe Airport do

  subject(:airport) { Airport.new }
  let(:plane) { double :plane, landed: nil, land: nil, take_off: nil }

  before do
    allow(airport).to receive(:stormy?) {false}
  end
  describe "#initialize" do
    it "should ensure that the default capacity is set" do
      expect(airport.capacity).to eq Airport::DEFAULT_CAPACITY
    end

    it "should allow the user to set the capacity" do
      expect(Airport.new(20).capacity).to eq 20
    end
  end

  describe "#land" do
    it "should raise an error if the plane is already landed" do
      allow(plane).to receive(:landed).and_return(true)
      expect { airport.land(plane) }.to raise_error("Plane is already landed")
    end

    before do
      airport.land(plane)
    end
    it "should allow the plane to land and stay landed in the airport" do
      expect(airport.planes).to include(plane)
    end

    it "should raise an error if the aiport is full" do
      airport = Airport.new(0)
      expect { airport.land(plane) }.to raise_error("Airport full")
    end

    it "should raise an error if the weather is stormy" do
      allow(airport).to receive(:stormy?).and_return(true)
      expect { airport.land(plane) }.to raise_error("Bad weather — unable to land")
    end
  end

  describe "#take_off" do
    it "should raise an error if the plane is not in the aiport" do
      expect { airport.take_off(plane) }.to raise_error("Plane not in airport")
    end

    it "should allow a plane to take off and leave the airport" do
      airport.land(plane)
      airport.take_off(plane)
      expect(airport.planes.include?(plane)).to eq false
    end

    it "should raise an error if the weather is stormy" do
      airport.land(plane)
      allow(airport).to receive(:stormy?).and_return(true)
      expect { airport.take_off(plane) }.to raise_error("Bad weather — unable to take off")
    end
  end

end
```

airport_challenge / spec / **plane_spec.rb**   Find file   Copy path

24 lines (19 sloc) | 388 Bytes

```ruby
require "plane"

describe Plane do

  describe "#land" do
    before do
      subject.land
    end
    it "should set the status to landed = true" do
      expect(subject.landed).to eq true
    end
  end

  describe "#take_off" do
    before do
      subject.take_off
    end
    it "should set the status to landed = false" do
      expect(subject.landed).to eq false
    end
  end

end
```

Tree: ea01fb49c7 ▾   **airport_challenge** / spec / **weather_spec.rb**

Find file   Copy path

Fetching contributors…

◯ Cannot retrieve contributors at this time

12 lines (8 sloc) | 213 Bytes

```ruby
1   require "weather"
2
3   describe Weather do
4
5     describe "#stormy?" do
6       it "should set the weather stormy status to either true or false" do
7         expect(subject.stormy?).to eq(true).or(eq(false))
8       end
9     end
10
11  end
```