Evidence for Implementation and Testing Unit.

Luis Farid Tejero Aoun

E - 21

I.T 1- Demonstrate one example of encapsulation that you have written in a program.

Evidence for unit

I.T 2 - Example the use of inheritance in a program.

Evidence for unit

I.T 3 - Example of searching

```
66
67  def self.film_by_id(id)
68    sql = "SELECT films.* FROM films WHERE films.id = $1"
69    values = [id]
70    films_hashes = SqlRunner.run(sql, values)
71    film = films_hashes.map {|film| Film.new(film)}
72    return film
73  end
```

Result of the search:

```
[➜  weeknd_nw git:(master) × ruby db/console.rb

From: /Users/user/codeclan_work/week_03/day_5/weeknd_hw/db/c
onsole.rb @ line 36 :

    31: ticket3.save()
    32:
    33:
    34:
    35: binding.pry
 => 36: nil

[[1] pry(main)> Ticket.film_by_id(2)                          ]
=> [#<Film:0x007fafa2cc20c8 @id=2, @price=15, @title="Drive"
>]
[2] pry(main)> ▊
```

I.T 4 – Example of sorting

```ruby
def self.most_sold()
  sql = "SELECT screening_id FROM tickets GROUP BY screening_id ORDER BY COUNT(*) DESC LIMIT
  1"
  tickets = SqlRunner.run(sql)[0]['screening_id'].to_i
  return self.film_by_id(tickets)[0].title + " is the most popular film at: " +
  self.screening_by_id(tickets)[0].function_time.to_s
end
```

Result of sorting:

```
[2] pry(main)> Ticket.most_sold()
=> "Infinity War is the most popular film at: 10"
[3] pry(main)>
```

I.T 5 - Example of an array, a function that uses an array and the result

```ruby
def find_a_pet(dog_name)
  @dogs_array.include?(dog_name)
end
```

```ruby
require("minitest/autorun")
require("minitest/rg")
require_relative("../array")


class MyArraysTest < MiniTest::Test
  def setup
    @dogs_array = ["Julieta", "Bongo", "Matheu"]
  end

  def test_find_pet_name__true
    @dogs_array.push("Paco")
    assert_equal(true, find_a_pet("Paco"))
  end

  def test_find_pet_name__false
    assert_equal(false, find_a_pet("Paco"))
  end
end
```

```
Run options: --seed 5500

# Running:

..

Finished in 0.001070s, 1869.1589 runs/s, 1869.1589 assertions/s.

2 runs, 2 assertions, 0 failures, 0 errors, 0 skips
→ i_t
```

specs/array_spec.rb   8:49   LF   UTF-8   Ruby   0 files

I.T 6 - Example of a hash, a function that uses a hash and the result

```ruby
# hashes.rb
def pets_by_type(shopname, type)
  count = []
  for pets in shopname[:pets]
    if (pets[:pet_type] == type)
      count.push(pets)
    end
  end
  return count
end
```

```ruby
# hashes_spec.rb
class MyHashTest < MiniTest::Test
  def setup
    @le_shop = {
      pets: [
        {
          name: "Lupe",
          pet_type: :cat,
          breed: "Lupidiam",
          price: 50
        },
        {
          name: "Basilius",
          pet_type: :cat,
          breed: "Bastium",
          price: 500
        },
        {
          name: "Le manche",
          pet_type: :doggo,
          breed: "Good boy",
          price: 1000,
        }
      ]
    }
  end

  def test_all_pets_by_type
    pets = pets_by_type(@le_shop, :doggo)
    assert_equal(1, pets.count)
  end
end
```

RESULT OF TEST:

```ruby
# hashes.rb
def pets_by_type(shopname, type)
  count = []
  for pets in shopname[:pets]
    if (pets[:pet_type] == type)
      count.push(pets)
    end
  end
  return count
end
```

```ruby
# hashes_spec.rb
class MyHashTest < MiniTest::Test
  def setup
    @le_shop = {
      pets: [
        {
          name: "Lupe",
          pet_type: :cat,
          breed: "Lupidiam",
          price: 50
        },
        {
          name: "Basilius",
          pet_type: :cat,
          breed: "Bastium",
          price: 500
        },
        {
          name: "Le manche",
          pet_type: :doggo,
          breed: "Good boy",
          price: 1000,
```

```
# Running:

.

Finished in 0.001169s, 855.4320 runs/s, 855.4320 assertions/s.

1 runs, 1 assertions, 0 failures, 0 errors, 0 skips
→ i_t
```

I.T 7 - Example of polymorphism in a program

Evidence for unit