Evidence for Implementation and Testing Unit.

Hamish Hoad

Cohort E-21

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

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

I.T 3 - Example of searching

Example of a function searching data and the output result.


```
console.rb
18  location1 = Location.new({ 'category' =>
    'Attractions', 'name' => 'Mordor'})
19  location1.save()
20  location2 = Location.new({ 'category' => 'Tavern',
    'name' => 'The Prancing Pony'})
21  location2.save()
22
23  visit1 = Visit.new({ 'user_id' => user1.id,
    'location_id' => location1.id, 'review' => '0
    stars, far too hot'})
24  visit1.save()
25
26  visit2 = Visit.new({ 'user_id' => user3.id,
    'location_id' => location1.id, 'review' => '5
    stars, would visit again if I could'})
27  visit2.save()
28
29  visit3 = Visit.new({ 'user_id' => user1.id,
    'location_id' => location2.id, 'review' => '4
    stars, plenty of beer available'})
30  visit3.save()
```

```
visit.rb
41  def location()
42    sql = "SELECT * FROM locations WHERE id = $1"
43    values = [@location_id]
44    location_hash = SqlRunner.run(sql, values).first()
45    return User.new(location_hash)
46  end
47
```

```
ruby console.rb — ruby — ruby
db ruby console.rb

From: /Users/hamishhoad/codeclan_work/week_03/day_4/quest_advisor_start/db/console.rb @ line 37 :

    32:
    33: visit4 = Visit.new({ 'user_id' => user2.id, 'location_id' => location2.id, 'review' => '3 stars,
  too crowded, could not find my wizard friend'})
    34: visit4.save()
    35:
    36: binding.pry
 => 37: nil

[1] pry(main)> Location.all
=> [#<Location:0x007fc6431ae8a0 @category="Attractions", @id=33, @name="Mordor">,
 #<Location:0x007fc6431ae7b0 @category="Tavern", @id=34, @name="The Prancing Pony">]
[2] pry(main)>
```

I.T 4 – Example of sorting

Example of a function sorting data and outputting a table of results.

```ruby
10
11   film1 = Film.new({
12     "title" => "Doom",
13     "ticket_price" => 500
14   })
15   film1.save()
16
17   film2 = Film.new({
18     "title" => "Star Wars, The Empire Strikes Back",
19     "ticket_price" => 500
20   })
21   film2.save()
22
23   film3 = Film.new({
24     "title" => "Tron",
25     "ticket_price" => 500
26   })
27   film3.save()
28
29   customer1 = Customer.new({
30     "name" => "William Goldman",
31     "funds" => 1000
32   })
33   customer1.save()
34
```

```ruby
37
38   # Show which films a customer has booked to see
39   def find_customer_films
40     sql = "SELECT films.*
41     FROM customers
42     INNER JOIN tickets ON customers.id = tickets.customer_id
43     INNER JOIN films ON tickets.film_id = films.id
44     WHERE customers.id = $1"
45     values = [@id]
46     results = SqlRunner.run(sql, values)
47     result = results.map{|hash| Film.new(hash)}
48     return result
49   end
50
```

```
psql -d cc_cinema — psql — psql
→ db git:(master) × psql -d cc_cinema
psql (10.3)
Type "help" for help.

cc_cinema=# SELECT films.* FROM customers INNER JOIN tickets ON customers.id = tickets.cust
omer_id INNER JOIN films ON tickets.film_id = films.id WHERE customers.id = 1;
 id | title | ticket_price
----+-------+--------------
  1 | Doom  |          500
(1 row)

cc_cinema=#
```

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

Example of a test array in a program, the function and the test results.

```ruby
16   def test_sum_array
17     result = sum_array( [ 1,2,3,4,5 ] )
18     assert_equal( 15, result )
19   end
20
21
22
23
```

```ruby
8    def sum_array(numbers)
9      result = 0
10     for number in numbers
11       result += number
12     end
13     return result
14   end
```

```
hamishhoad@Hamishs-MBP — ..t_point/specs — -zsh
→ specs ruby my_functions_spec.rb
Run options: --seed 28893

# Running:

.....

Finished in 0.000802s, 6234.4140 runs/s, 6234.4140 assertions/s.

5 runs, 5 assertions, 0 failures, 0 errors, 0 skips
→ specs
```

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

Example of a hash in a program, a function that calls the hash and the output result.

```ruby
console.rb
18  location1 = Location.new({ 'category' =>
    'Attractions', 'name' => 'Mordor'})
19  location1.save()
20  location2 = Location.new({ 'category' => 'Tavern',
    'name' => 'The Prancing Pony'})
21  location2.save()
22
23  visit1 = Visit.new({ 'user_id' => user1.id,
    'location_id' => location1.id, 'review' => '0
    stars, far too hot'})
24  visit1.save()
25
26  visit2 = Visit.new({ 'user_id' => user3.id,
    'location_id' => location1.id, 'review' => '5
    stars, would visit again if I could'})
27  visit2.save()
28
29  visit3 = Visit.new({ 'user_id' => user1.id,
    'location_id' => location2.id, 'review' => '4
    stars, plenty of beer available'})
30  visit3.save()
```

```ruby
visit.rb
41  def location()
42    sql = "SELECT * FROM locations WHERE id = $1"
43    values = [@location_id]
44    location_hash = SqlRunner.run(sql, values).first()
45    return User.new(location_hash)
46  end
47
```

```
ruby console.rb — ruby — ruby
db ruby console.rb

From: /Users/hamishhoad/codeclan_work/week_03/day_4/quest_advisor_start/db/console.rb @ line 37 :

    32:
    33: visit4 = Visit.new({ 'user_id' => user2.id, 'location_id' => location2.id, 'review' => '3 stars,
 too crowded, could not find my wizard friend'})
    34: visit4.save()
    35:
    36: binding.pry
 => 37: nil

[1] pry(main)> Location.all
=> [#<Location:0x007fc6431ae8a0 @category="Attractions", @id=33, @name="Mordor">,
 #<Location:0x007fc6431ae7b0 @category="Tavern", @id=34, @name="The Prancing Pony">]
[2] pry(main)>
```

I.T 7 - Example of polymorphism in a program