

Evidence for Implementation and Testing Unit.

Joseph Manuel

E20

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

I.T 4 – Example of sorting

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

```
room.rb      guest.rb      song.rb      room_spec.rb      guest_spec.rb      song_spec.rb
1 class Room
2
3   attr_reader :room_number, :occupants, :playlist
4
5
6   def initialize(new_room_number, new_occupants,
7     new_playlist)
8     @room_number = new_room_number
9     @occupants = new_occupants
10    @playlist = new_playlist
11  end
12
13  def add_song_to_playlist()
14    | @playlist.playlist
15  end
16
17  def add_guest_to_occupants(guest_to_add)
18    | @occupants << guest_to_add
19  end
20
21  def remove_guest_from_occupants(guest_to_remove)
22    | @occupants.delete(guest_to_remove)
23  end
24
25
26
27 end
28

1 require('minitest/autorun')
2 require_relative('../room.rb')
3 require_relative('../song.rb')
4 require_relative('../guest.rb')
5
6 class TestRoom < MiniTest::Test
7
8   def setup()
9
10
11
12   @song1 = Song.new("song1","artist1")
13   @song2 = Song.new("song2","artist2")
14
15   @array_of_songs = [@song1,@song2]
16
17   @guest1 = Guest.new("guest1")
18   @guest2 = Guest.new("guest2")
19   @guest3 = Guest.new("guest3")
20   @guest4 = Guest.new("guest4")
21
22   @occupants = [@guest1,@guest2,@guest3,@guest4]
23
24   @room_1 = Room.new(101, @occupants, @array_of_songs)
25
26
27 end
28
29 # testing the getters
30 def test_room_getters()
31   assert_equal(101, @room_1.room_number)
32   assert_equal(@occupants, @room_1.occupants)
33   assert_equal(@array_of_songs, @room_1.playlist)
34 end
35
36

Ruby - room_spec.rb:57 ✓

Run options: --seed 36711

# Running:

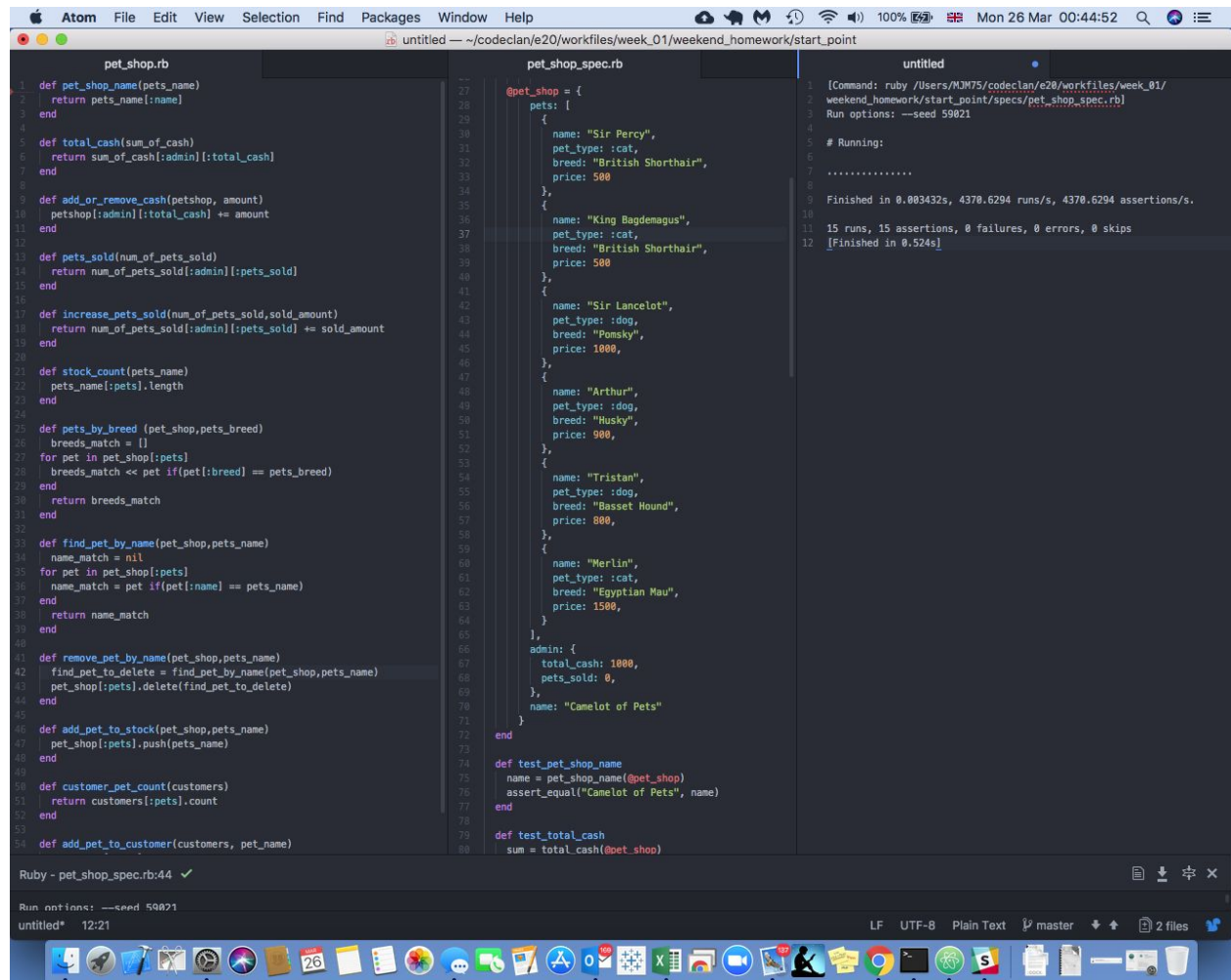
...

Finished in 0.001492s, 2010.7238 runs/s, 3351.2864 assertions/s.

3 runs, 5 assertions, 0 failures, 0 errors, 0 skips
[Finished in 0.75s]

room.rb* 25:1  LF UTF-8 Ruby  master  4 files
```

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



```
pet_shop.rb
1 def pet_shop_name(pets_name)
2   return pets_name[:name]
3 end
4
5 def total_cash(sum_of_cash)
6   return sum_of_cash[:admin][:total_cash]
7 end
8
9 def add_or_remove_cash(petshop, amount)
10  petshop[:admin][:total_cash] += amount
11 end
12
13 def pets_sold(num_of_pets_sold)
14  return num_of_pets_sold[:admin][:pets_sold]
15 end
16
17 def increase_pets_sold(num_of_pets_sold,sold_amount)
18  return num_of_pets_sold[:admin][:pets_sold] += sold_amount
19 end
20
21 def stock_count(pets_name)
22  pets_name[:pets].length
23 end
24
25 def pets_by_breed (petshop,pets_breed)
26  breeds_match = []
27  for pet in petshop[:pets]
28    breeds_match << pet if(pet[:breed] == pets_breed)
29  end
30  return breeds_match
31 end
32
33 def find_pet_by_name(pet_shop,pets_name)
34  name_match = nil
35  for pet in pet_shop[:pets]
36    name_match = pet if(pet[:name] == pets_name)
37  end
38  return name_match
39 end
40
41 def remove_pet_by_name(pet_shop,pets_name)
42  find_pet_to_delete = find_pet_by_name(pet_shop,pets_name)
43  pet_shop[:pets].delete(find_pet_to_delete)
44 end
45
46 def add_pet_to_stock(pet_shop,pets_name)
47  pet_shop[:pets].push(pets_name)
48 end
49
50 def customer_pet_count(customers)
51  return customers[:pets].count
52 end
53
54 def add_pet_to_customer(customers, pet_name)
```

```
pet_shop_spec.rb
27 @@pet_shop = {
28   pets: [
29     {
30       name: "Sir Percy",
31       pet_type: :cat,
32       breed: "British Shorthair",
33       price: 500
34     },
35     {
36       name: "King Bagdemagus",
37       pet_type: :cat,
38       breed: "British Shorthair",
39       price: 500
40     },
41     {
42       name: "Sir Lancelot",
43       pet_type: :dog,
44       breed: "Pomsky",
45       price: 1000,
46     },
47     {
48       name: "Arthur",
49       pet_type: :dog,
50       breed: "Husky",
51       price: 900,
52     },
53     {
54       name: "Tristan",
55       pet_type: :dog,
56       breed: "Basset Hound",
57       price: 800,
58     },
59     {
60       name: "Merlin",
61       pet_type: :cat,
62       breed: "Egyptian Mau",
63       price: 1500,
64     }
65   ],
66   admin: {
67     total_cash: 1000,
68     pets_sold: 0,
69   },
70   name: "Camelot of Pets"
71 }
72
73 end
74
75 def test_pet_shop_name
76   name = pet_shop_name(@pet_shop)
77   assert_equal("Camelot of Pets", name)
78 end
79
80 def test_total_cash
81   sum = total_cash(@pet_shop)
```

```
untitled
1 [Command: ruby /Users/MJM75/codeclan/e20/workfiles/week_01/
2 weekend_homework/start_point/specs/pet_shop_spec.rb]
3 Run options: --seed 59021
4
5 # Running:
6
7 .....
8
9 Finished in 0.003432s, 4370.6294 runs/s, 4370.6294 assertions/s.
10
11 15 runs, 15 assertions, 0 failures, 0 errors, 0 skips
12 [Finished in 0.524s]
```

## I.T 7 - Example of polymorphism in a program