Claire Connachan
CodeClan Cohort - E20

Evidence for Implementation and Testing Unit

I.T 1 - Take a screenshot of an example of encapsulation in a program.

!.T 2 - Take a screenshot of the use of inheritance in a program.

I.T 3 Demonstrate searching data in a program.
Screenshot of a function that searches data:

```ruby
def is_song_in_playlist?(song_to_check)
  song_titles = @playlist.map { |song| song.title }
  song_titles.include?(song_to_check.title)
end
```

Screenshot of the result of the function running:

```
[→  homework git:(master) ✗ ruby runner.rb
true
```

I.T 4 Demonstrate sorting data in a program.
Screenshot of a function that sorts data:

```ruby
def most_popular_screening()
  sql = "
    SELECT COUNT(screenings.time), screenings.time
      FROM tickets
      INNER JOIN films ON tickets.film_id = films.id
      INNER JOIN screenings ON tickets.screening_id =
      screenings.id
      WHERE films.id = $1
      GROUP BY screenings.time
      ORDER BY COUNT(screenings.time) DESC
      LIMIT 1;
    "
  values = [@id]
  array = SqlRunner.run(sql, values)
  return array[0]
end
```

Screenshot of the result of the function running:

```
→  homework git:(master) ✗ ruby runner.rb
{"count"=>"4", "time"=>"9am"}
```

I.T 5 Demonstrate the use of an array in a program.

Screenshot of an array in a program and function that uses the array:

```ruby
claire = Guest.new("Claire")
ewa = Guest.new("Ewa")
mike = Guest.new("Mike")
aileen = Guest.new("Aileen")

@guest = Guest.new("Lewis", 20)

@occupants = [claire, ewa, mike, aileen]

@room = Room.new("Karaoke Room",
@occupants, @playlist)
```

```ruby
def add_guest(guest_to_add)
    if @occupants.count < @capacity &&
    guest_to_add.wallet >= @fee
      @occupants << guest_to_add
      @till += @fee
    end
end
```

Screenshot of the result of the function running:

```ruby
p @room.occupants
@room.add_guest(@guest)
p @room.occupants
```

```
→  homework git:(master) ✗ ruby runner.rb
[#<Guest:0x007fcacda98cc0 @name="Claire", @wallet=0, @fave_song=nil>, #<Guest:0x
007fcacda98c70 @name="Ewa", @wallet=0, @fave_song=nil>, #<Guest:0x007fcacda98c20
 @name="Mike", @wallet=0, @fave_song=nil>, #<Guest:0x007fcacda98bd0 @name="Ailee
n", @wallet=0, @fave_song=nil>]
[#<Guest:0x007fcacda98cc0 @name="Claire", @wallet=0, @fave_song=nil>, #<Guest:0x
007fcacda98c70 @name="Ewa", @wallet=0, @fave_song=nil>, #<Guest:0x007fcacda98c20
 @name="Mike", @wallet=0, @fave_song=nil>, #<Guest:0x007fcacda98bd0 @name="Ailee
n", @wallet=0, @fave_song=nil>, #<Guest:0x007fcacda98b80 @name="Lewis", @wallet=
20, @fave_song=nil>]
→  homework git:(master) ✗
```

I.T 6 Demonstrate the use of a hash in a program.

Screenshot of a hash in a program and a function that uses the hash:

```ruby
@customers = [
  { name: "Craig", pets: [], cash: 1000 },
  { name: "Zsolt", pets: [], cash: 50 }
]

@new_pet = { name: "Bors the Younger", pet_type: :cat, breed: "Cornish Rex", price: 100 }

@pet_shop = { pets: [
      { name: "Sir Percy", pet_type: :cat, breed: "British Shorthair", price: 500 },
      { name: "King Bagdemagus", pet_type: :cat, breed: "British Shorthair", price: 500 },
      { name: "Sir Lancelot", pet_type: :dog, breed: "Pomsky", price: 1000 },
      { name: "Arthur", pet_type: :dog, breed: "Husky", price: 900 },
      { name: "Tristan", pet_type: :dog, breed: "Basset Hound", price: 800 },
      { name: "Merlin", pet_type: :cat, breed: "Egyptian Mau", price: 1500 }
    ],
    admin: { total_cash: 1000, pets_sold: 0 }, name: "Camelot of Pets" }
```

```ruby
def find_pet_by_name(shop, expected_name)
  result = nil
  for pet in shop[:pets]
    if pet[:name] == expected_name
      result = pet
    end
  end
  return result
end
```

Screenshot of the result of the function running:

```
[→  start_point git:(master) ✗ ruby runner.rb
{:name=>"Arthur", :pet_type=>:dog, :breed=>"Husky", :price=>900}
→  start_point git:(master) ✗
```