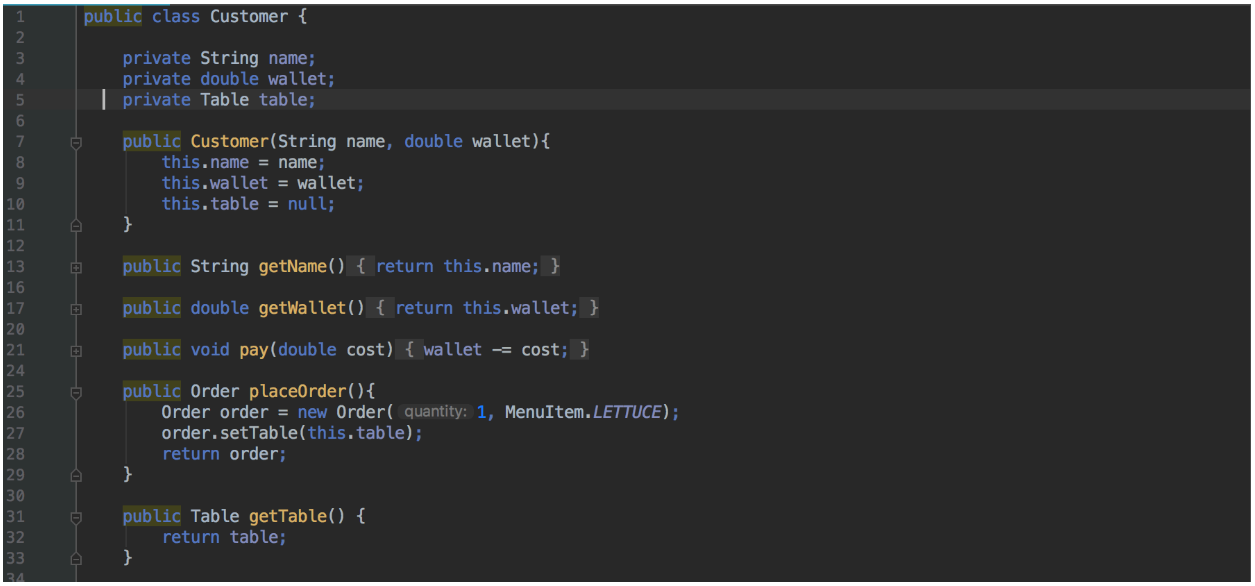
Evidence for Implementation and Testing Unit.

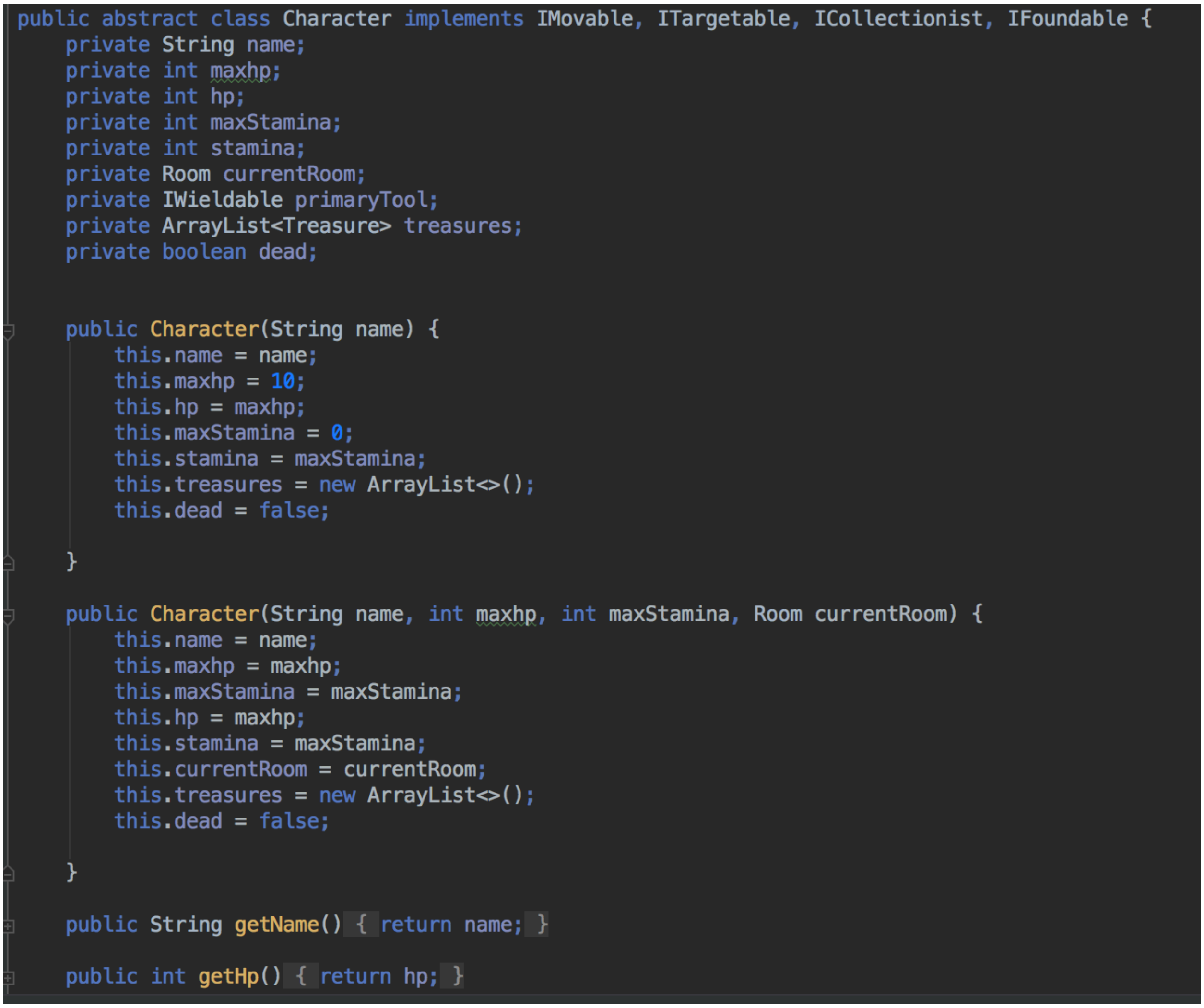
Jaime Lopez
Cohort E20

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

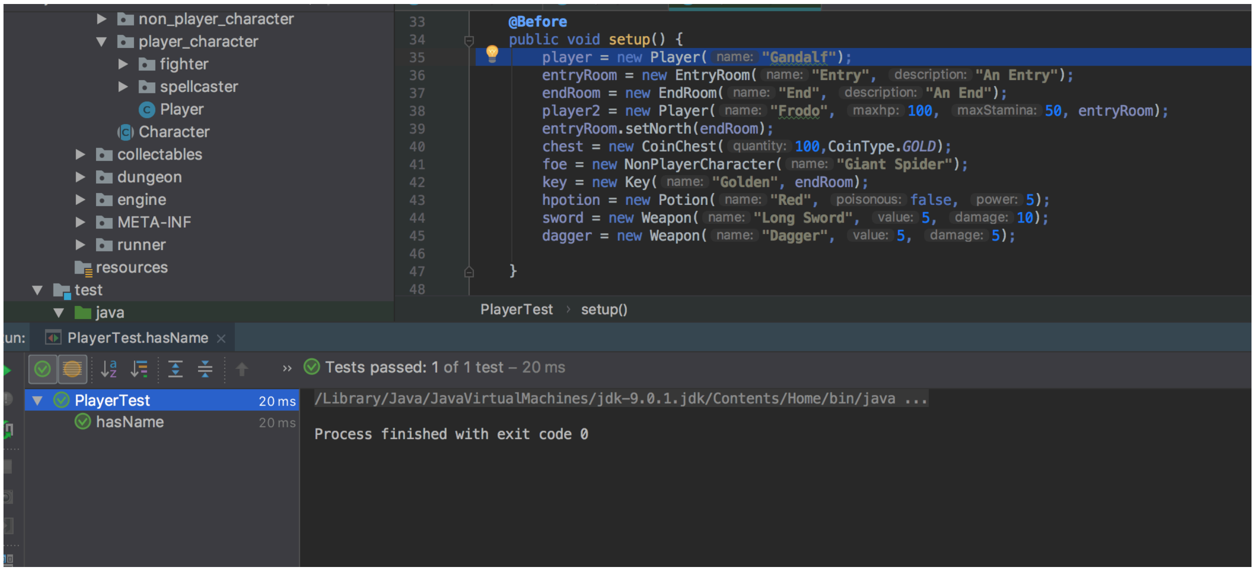


```
public class Customer {

    private String name;
    private double wallet;
    private Table table;

    public Customer(String name, double wallet){
        this.name = name;
        this.wallet = wallet;
        this.table = null;
    }

    public String getName() { return this.name; }

    public double getWallet() { return this.wallet; }

    public void pay(double cost) { wallet -= cost; }

    public Order placeOrder(){
        Order order = new Order( quantity: 1, MenuItem.LETTUCE);
        order.setTable(this.table);
        return order;
    }

    public Table getTable() {
        return table;
    }
```

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

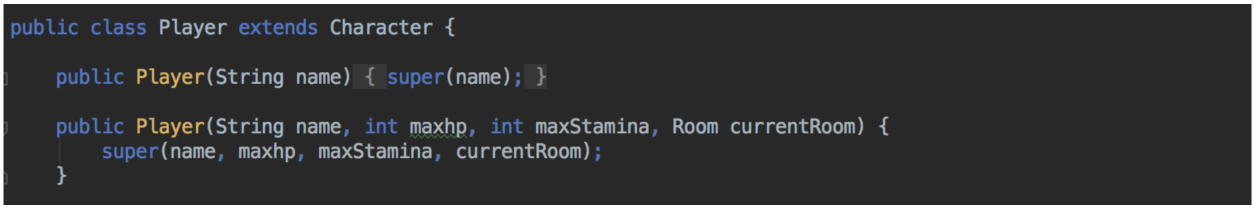


```
public abstract class Character implements IMovable, ITargetable, ICollectionist, IFoundable {
    private String name;
    private int maxhp;
    private int hp;
    private int maxStamina;
    private int stamina;
    private Room currentRoom;
    private IWieldable primaryTool;
    private ArrayList<Treasure> treasures;
    private boolean dead;


    public Character(String name) {
        this.name = name;
        this.maxhp = 10;
        this.hp = maxhp;
        this.maxStamina = 0;
        this.stamina = maxStamina;
        this.treasures = new ArrayList<>();
        this.dead = false;

    }

    public Character(String name, int maxhp, int maxStamina, Room currentRoom) {
        this.name = name;
        this.maxhp = maxhp;
        this.maxStamina = maxStamina;
        this.hp = maxhp;
        this.stamina = maxStamina;
        this.currentRoom = currentRoom;
        this.treasures = new ArrayList<>();
        this.dead = false;

    }

    public String getName() { return name; }

    public int getHp() { return hp; }
```



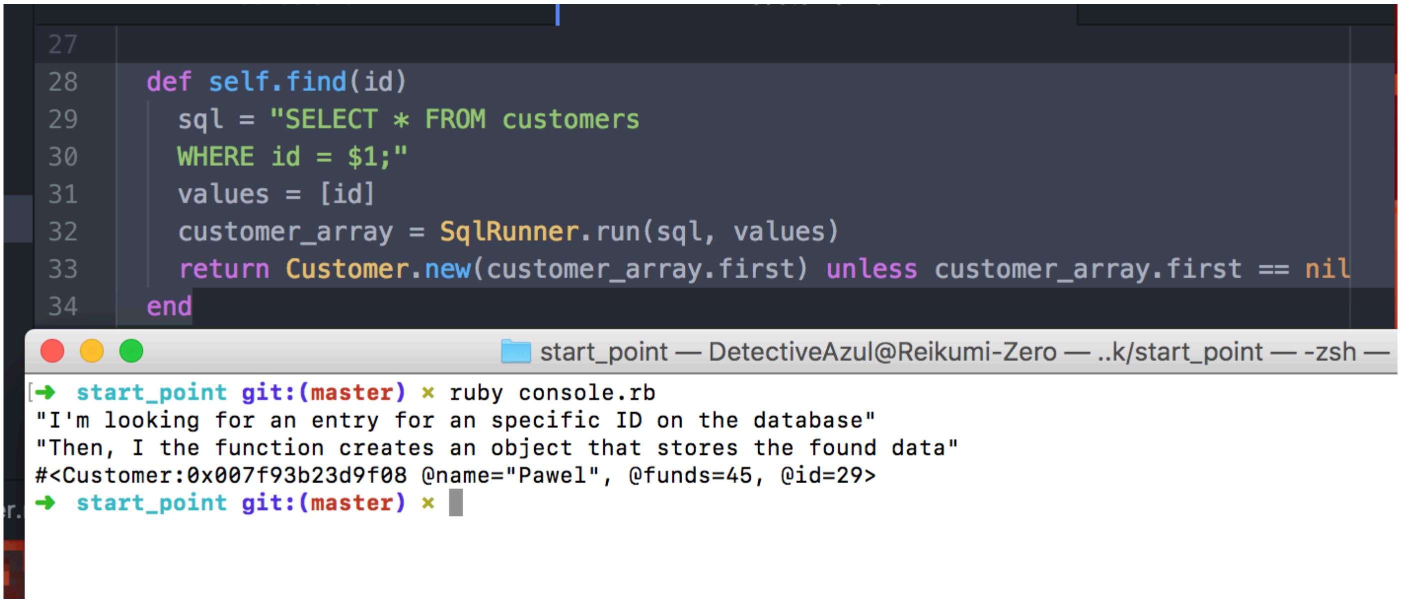
```
@Before
public void setup() {
    player = new Player( name: "Gandalf");
    entryRoom = new EntryRoom( name: "Entry",  description: "An Entry");
    endRoom = new EndRoom( name: "End",  description: "An End");
    player2 = new Player( name: "Frodo",  maxhp: 100,  maxStamina: 50, entryRoom);
    entryRoom.setNorth(endRoom);
    chest = new CoinChest( quantity: 100,CoinType.GOLD);
    foe = new NonPlayerCharacter( name: "Giant Spider");
    key = new Key( name: "Golden", endRoom);
    hpotion = new Potion( name: "Red",  poisonous: false,  power: 5);
    sword = new Weapon( name: "Long Sword",  value: 5,  damage: 10);
    dagger = new Weapon( name: "Dagger",  value: 5,  damage: 5);

}
```

```
Tests passed: 1 of 1 test – 20 ms

PlayerTest   20 ms
  hasName    20 ms

/Library/Java/JavaVirtualMachines/jdk-9.0.1.jdk/Contents/Home/bin/java ...

Process finished with exit code 0
```



```
public class Player extends Character {

    public Player(String name) { super(name); }

    public Player(String name, int maxhp, int maxStamina, Room currentRoom) {
        super(name, maxhp, maxStamina, currentRoom);
    }
```

I.T 3 - Example of searching

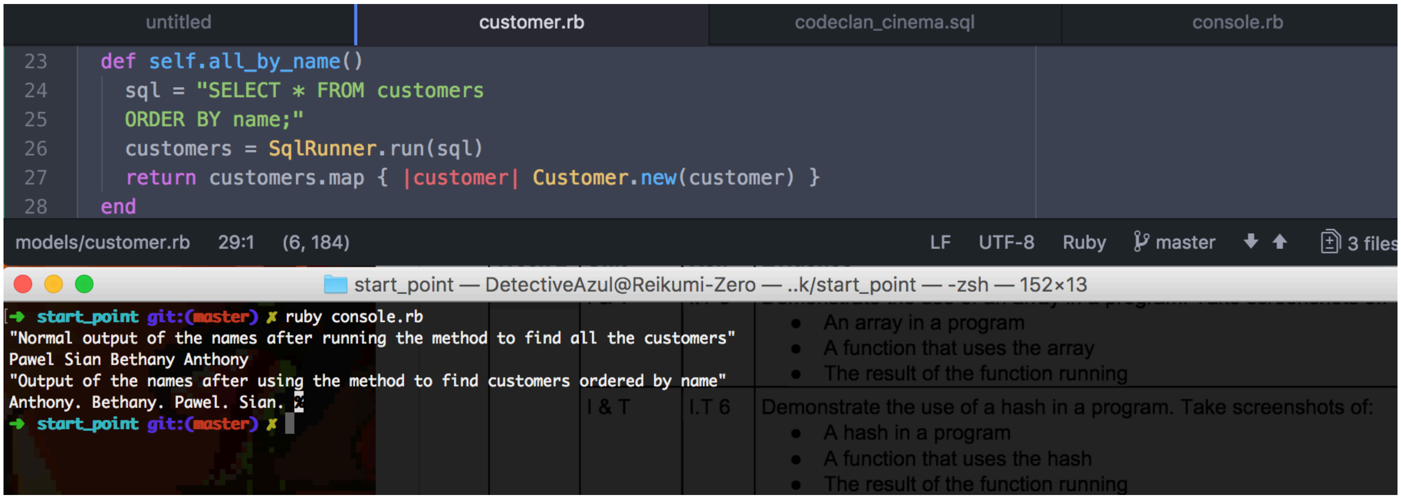


```ruby

def self.find(id)
  sql = "SELECT * FROM customers
  WHERE id = $1;"
  values = [id]
  customer_array = SqlRunner.run(sql, values)
  return Customer.new(customer_array.first) unless customer_array.first == nil
end
```

start_point — DetectiveAzul@Reikumi-Zero — ..k/start_point — -zsh —

```
➜ start_point git:(master) ✗ ruby console.rb
"I'm looking for an entry for an specific ID on the database"
"Then, I the function creates an object that stores the found data"
#<Customer:0x007f93b23d9f08 @name="Pawel", @funds=45, @id=29>
➜ start_point git:(master) ✗
```

I.T 4 - Example of sorting



untitled
customer.rb
codeclan_cinema.sql
console.rb

```ruby
def self.all_by_name()
  sql = "SELECT * FROM customers
  ORDER BY name;"
  customers = SqlRunner.run(sql)
  return customers.map { |customer| Customer.new(customer) }
end
```

models/customer.rb 29:1 (6, 184)
LF UTF-8 Ruby master
3 files
start_point — DetectiveAzul@Reikumi-Zero — ..k/start_point — -zsh — 152×13

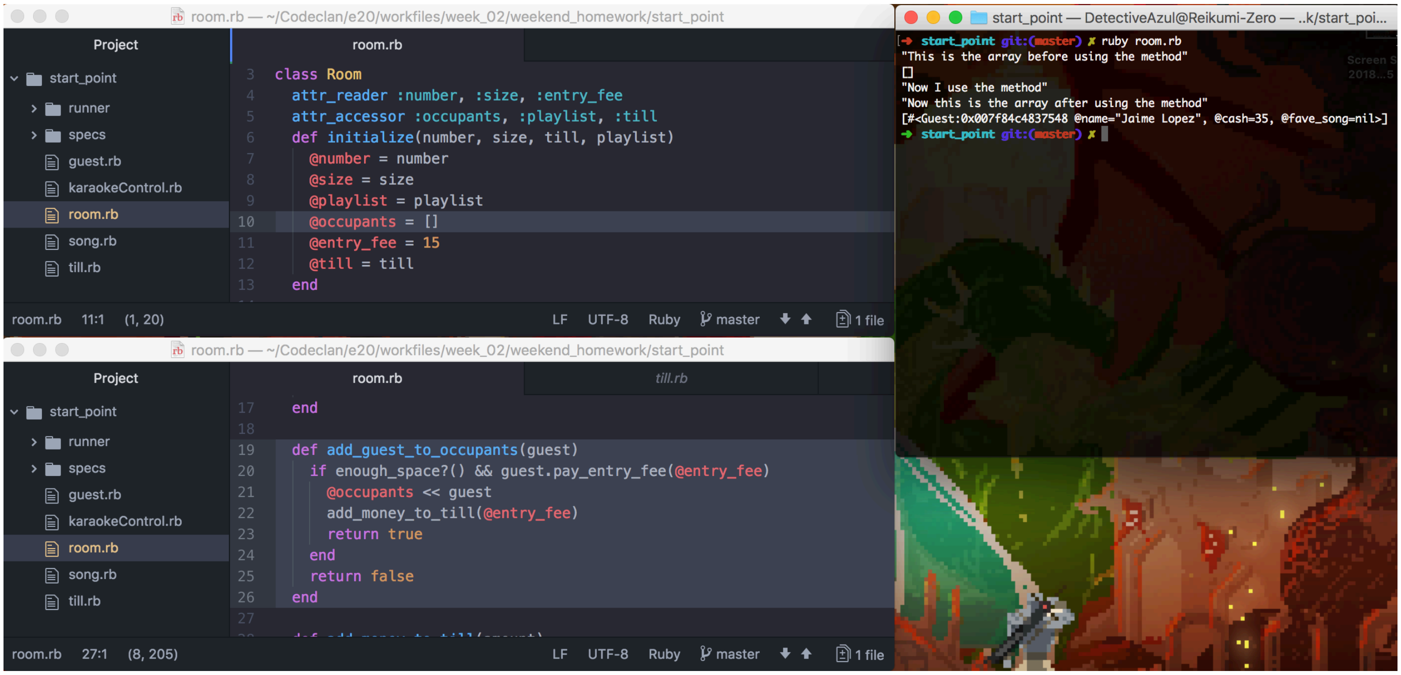
```
➜ start_point git:(master) ✗ ruby console.rb
"Normal output of the names after running the method to find all the customers"
Pawel Sian Bethany Anthony
"Output of the names after using the method to find customers ordered by name"
Anthony. Bethany. Pawel. Sian. ✗
➜ start_point git:(master) ✗
```

- An array in a program
- A function that uses the array
- The result of the function running

I & T I.T 6 Demonstrate the use of a hash in a program. Take screenshots of:

- A hash in a program
- A function that uses the hash
- The result of the function running

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

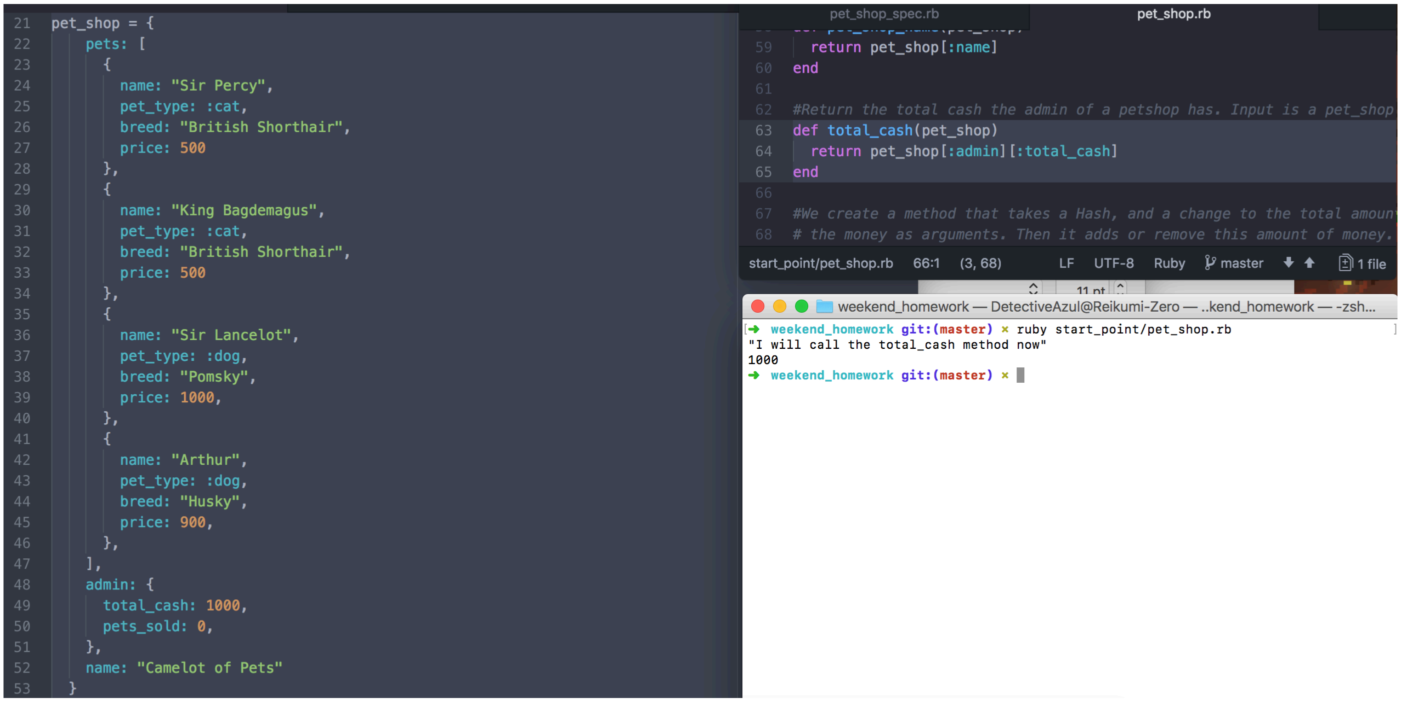

room.rb — ~/Codeclan/e20/workfiles/week_02/weekend_homework/start_point
Project
start_point
runner
specs
guest.rb
karaokeControl.rb
room.rb
song.rb
till.rb
room.rb
room.rb 11:1 (1, 20) LF UTF-8 Ruby master 1 file
start_point — DetectiveAzul@Reikumi-Zero — ..k/start_poi...
room.rb — ~/Codeclan/e20/workfiles/week_02/weekend_homework/start_point
room.rb
till.rb
room.rb 27:1 (8, 205) LF UTF-8 Ruby master 1 file

```ruby
class Room
  attr_reader :number, :size, :entry_fee
  attr_accessor :occupants, :playlist, :till
  def initialize(number, size, till, playlist)
    @number = number
    @size = size
    @playlist = playlist
    @occupants = []
    @entry_fee = 15
    @till = till
  end
```

```ruby
  end

  def add_guest_to_occupants(guest)
    if enough_space?() && guest.pay_entry_fee(@entry_fee)
      @occupants << guest
      add_money_to_till(@entry_fee)
      return true
    end
    return false
  end

  def add_money_to_till(amount)
```

```
[➜ start_point git:(master) ✗ ruby room.rb
"This is the array before using the method"
[]
"Now I use the method"
"Now this is the array after using the method"
[#<Guest:0x007f84c4837548 @name="Jaime Lopez", @cash=35, @fave_song=nil>]
➜ start_point git:(master) ✗
```

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

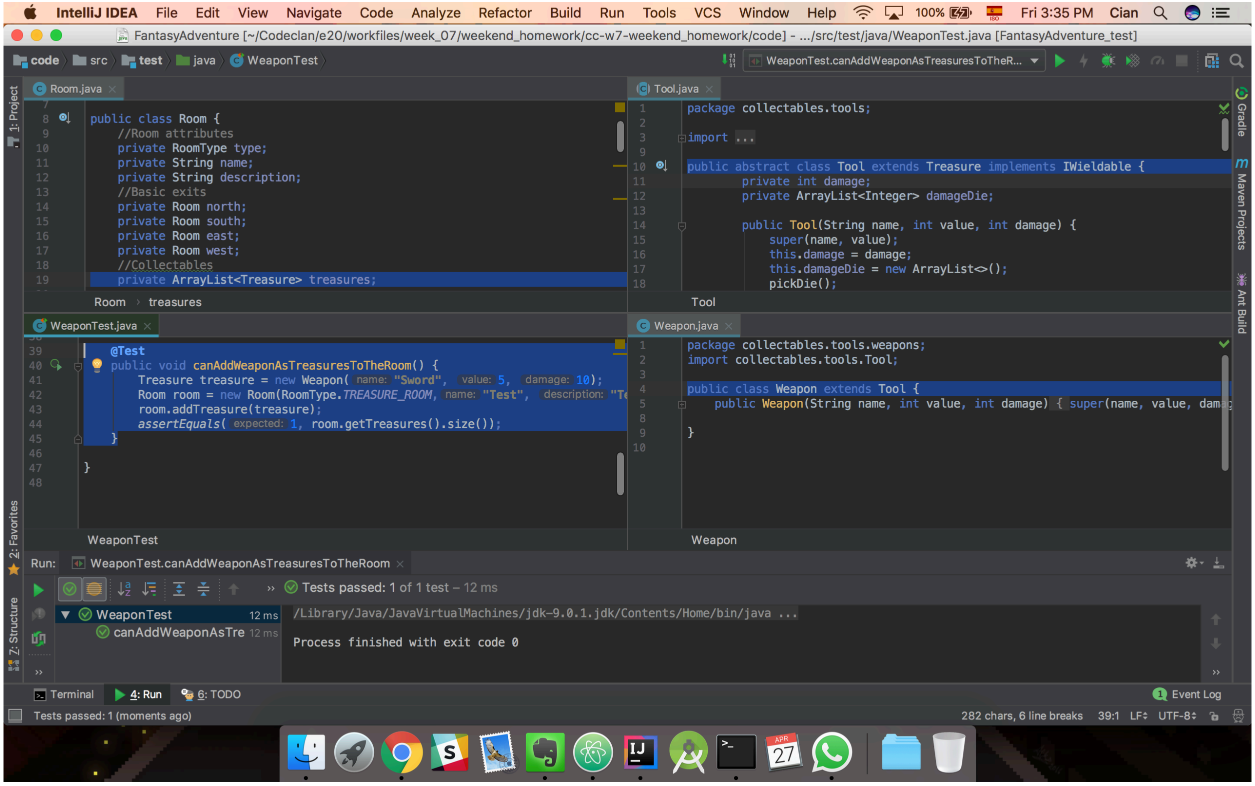

pet_shop_spec.rb
pet_shop.rb
start_point/pet_shop.rb 66:1 (3, 68) LF UTF-8 Ruby master 1 file
11 pt
weekend_homework — DetectiveAzul@Reikumi-Zero — ..kend_homework — -zsh...

```ruby
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,
      },
    ],
    admin: {
      total_cash: 1000,
      pets_sold: 0,
    },
    name: "Camelot of Pets"
  }
```

```ruby
def pet_shop_name(pet_shop)
  return pet_shop[:name]
end

#Return the total cash the admin of a petshop has. Input is a pet_shop
def total_cash(pet_shop)
  return pet_shop[:admin][:total_cash]
end

#We create a method that takes a Hash, and a change to the total amoun
# the money as arguments. Then it adds or remove this amount of money.
```

```
[➜ weekend_homework git:(master) × ruby start_point/pet_shop.rb
"I will call the total_cash method now"
1000
➜ weekend_homework git:(master) × 
```

I.T 7 - Example of polymorphism in a program



FantasyAdventure [~/Codeclan/e20/workfiles/week_07/weekend_homework/cc-w7-weekend_homework/code] - .../src/test/java/WeaponTest.java [FantasyAdventure_test]

Room.java

```java
public class Room {
    //Room attributes
    private RoomType type;
    private String name;
    private String description;
    //Basic exits
    private Room north;
    private Room south;
    private Room east;
    private Room west;
    //Collectables
    private ArrayList<Treasure> treasures;
```

Tool.java

```java
package collectables.tools;

import ...

public abstract class Tool extends Treasure implements IWieldable {
        private int damage;
        private ArrayList<Integer> damageDie;

        public Tool(String name, int value, int damage) {
            super(name, value);
            this.damage = damage;
            this.damageDie = new ArrayList<>();
            pickDie();
```

WeaponTest.java

```java
    @Test
    public void canAddWeaponAsTreasuresToTheRoom() {
        Treasure treasure = new Weapon( name: "Sword",  value: 5,  damage: 10);
        Room room = new Room(RoomType.TREASURE_ROOM,  name: "Test",  description: "Te
        room.addTreasure(treasure);
        assertEquals( expected: 1, room.getTreasures().size());
    }

}
```

Weapon.java

```java
package collectables.tools.weapons;
import collectables.tools.Tool;

public class Weapon extends Tool {
    public Weapon(String name, int value, int damage) { super(name, value, damag

}
```

Run: WeaponTest.canAddWeaponAsTreasuresToTheRoom

Tests passed: 1 of 1 test – 12 ms

WeaponTest 12 ms
canAddWeaponAsTre 12 ms

```
/Library/Java/JavaVirtualMachines/jdk-9.0.1.jdk/Contents/Home/bin/java ...

Process finished with exit code 0
```

Tests passed: 1 (moments ago)