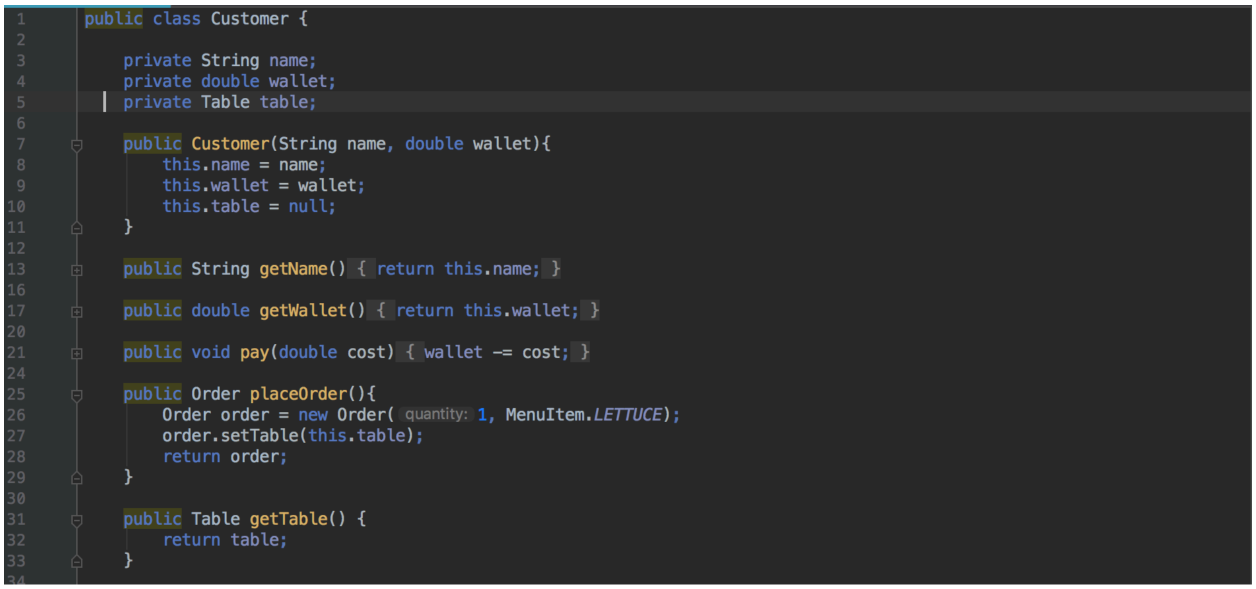
Evidence for Implementation and Testing Unit.

Jaime Lopez
Cohort E20

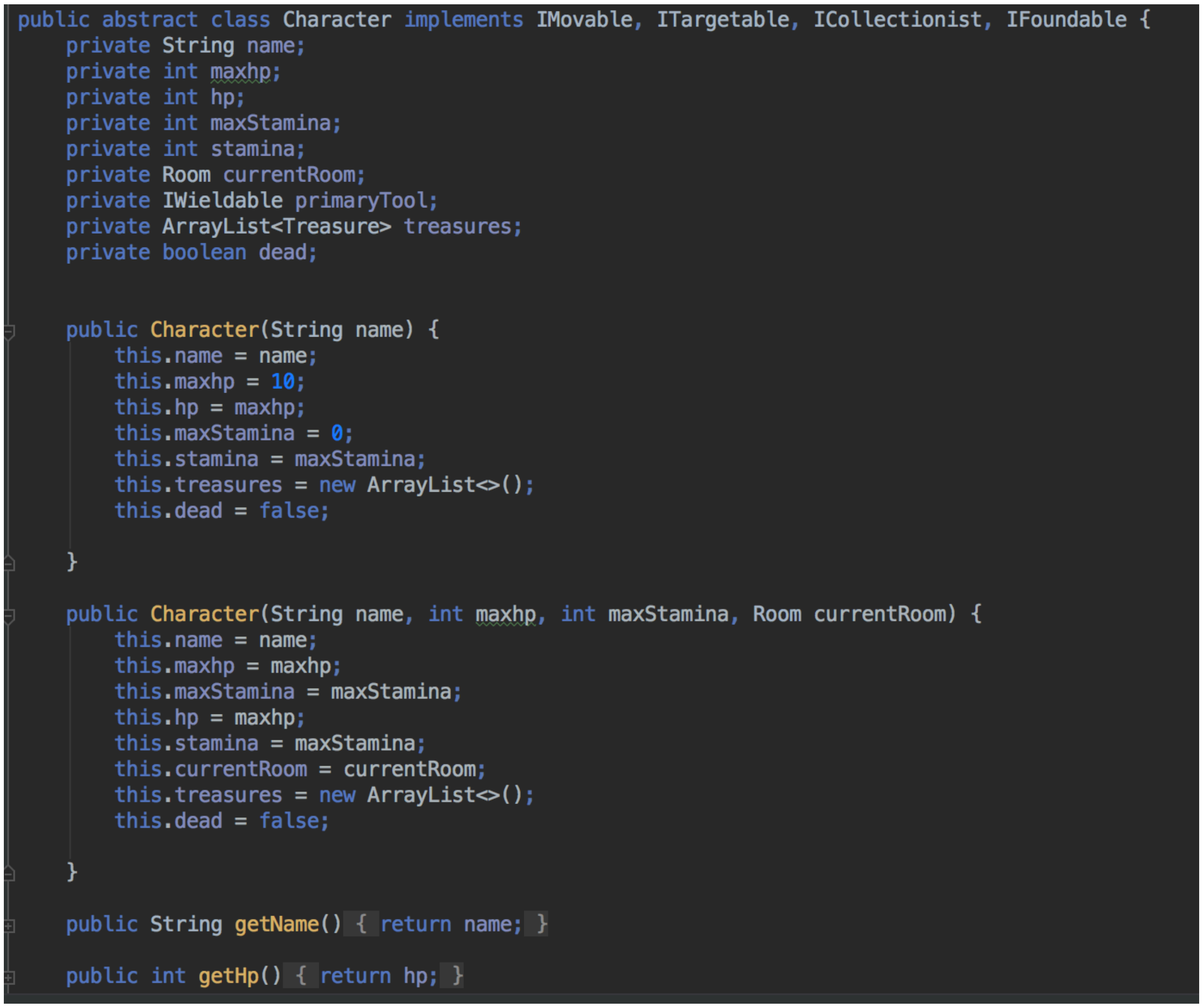
I.T 1- Demonstrate one example of encapsulation you have written in a program.



```
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;
    }
```

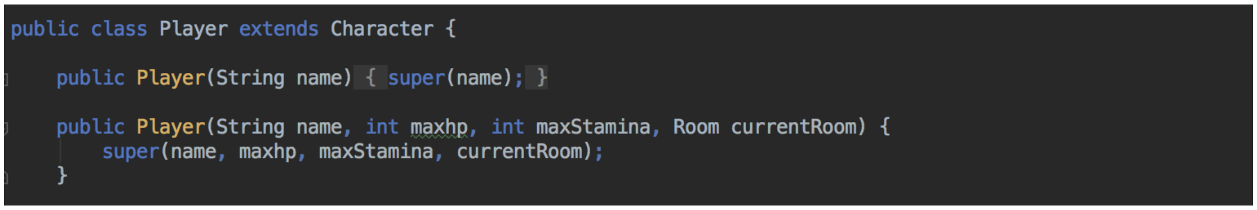
I.T 2 - Example the use of inheritance in a program.

Screenshot showing the parent class and the getName() method



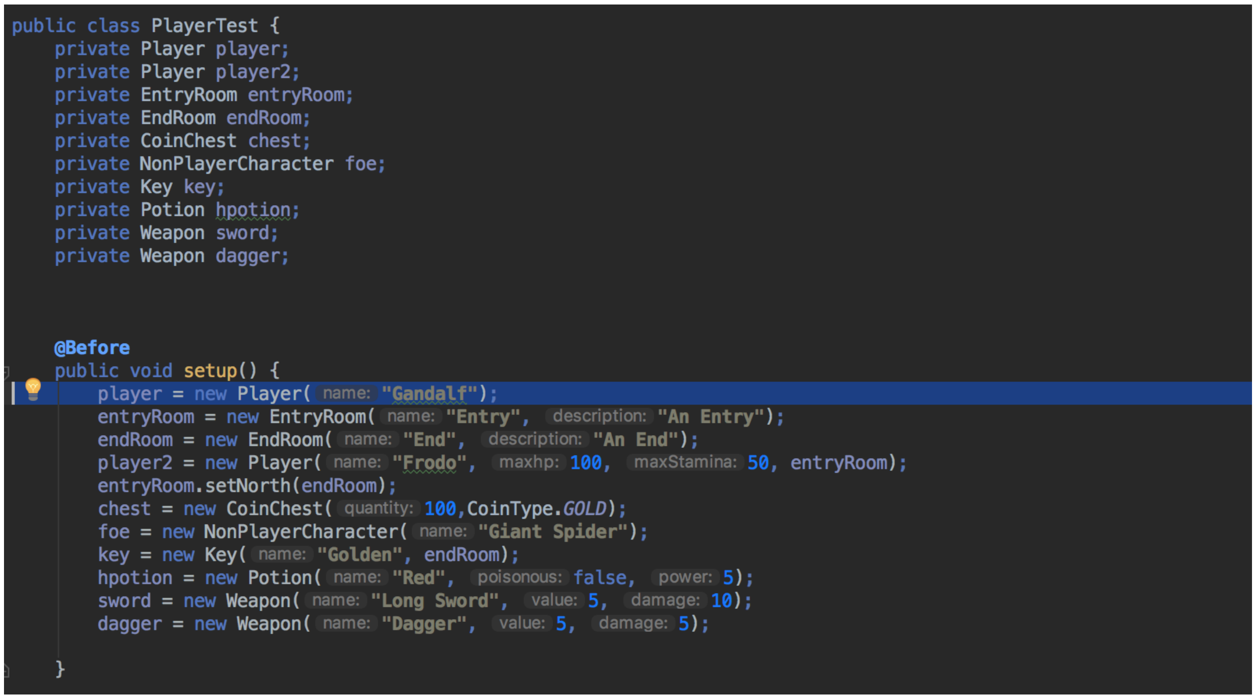
```java
public abstract class Character implements IMovable, ITargetable, ICollectionist, IFoundable {
    private String name;
    private int maxhp;
    private int hp;
    private int maxStamina;
    private int stamina;
    private Room currentRoom;
    private IWieldable primaryTool;
    private 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; }
```

Screenshot showing the child class (that call the super method to inherit from parent)



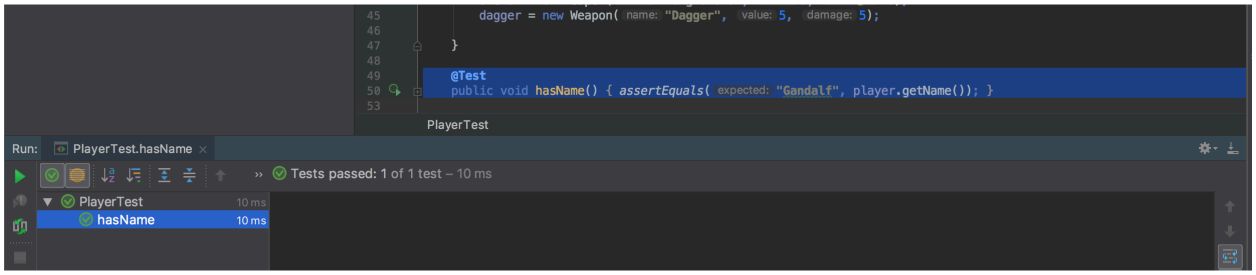
```java
public class Player extends Character {

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

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

Screenshot where we show the player being created on a Test File



```java
public class PlayerTest {
    private Player player;
    private Player player2;
    private EntryRoom entryRoom;
    private EndRoom endRoom;
    private CoinChest chest;
    private NonPlayerCharacter foe;
    private Key key;
    private Potion hpotion;
    private Weapon sword;
    private Weapon dagger;


    @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);

    }
```

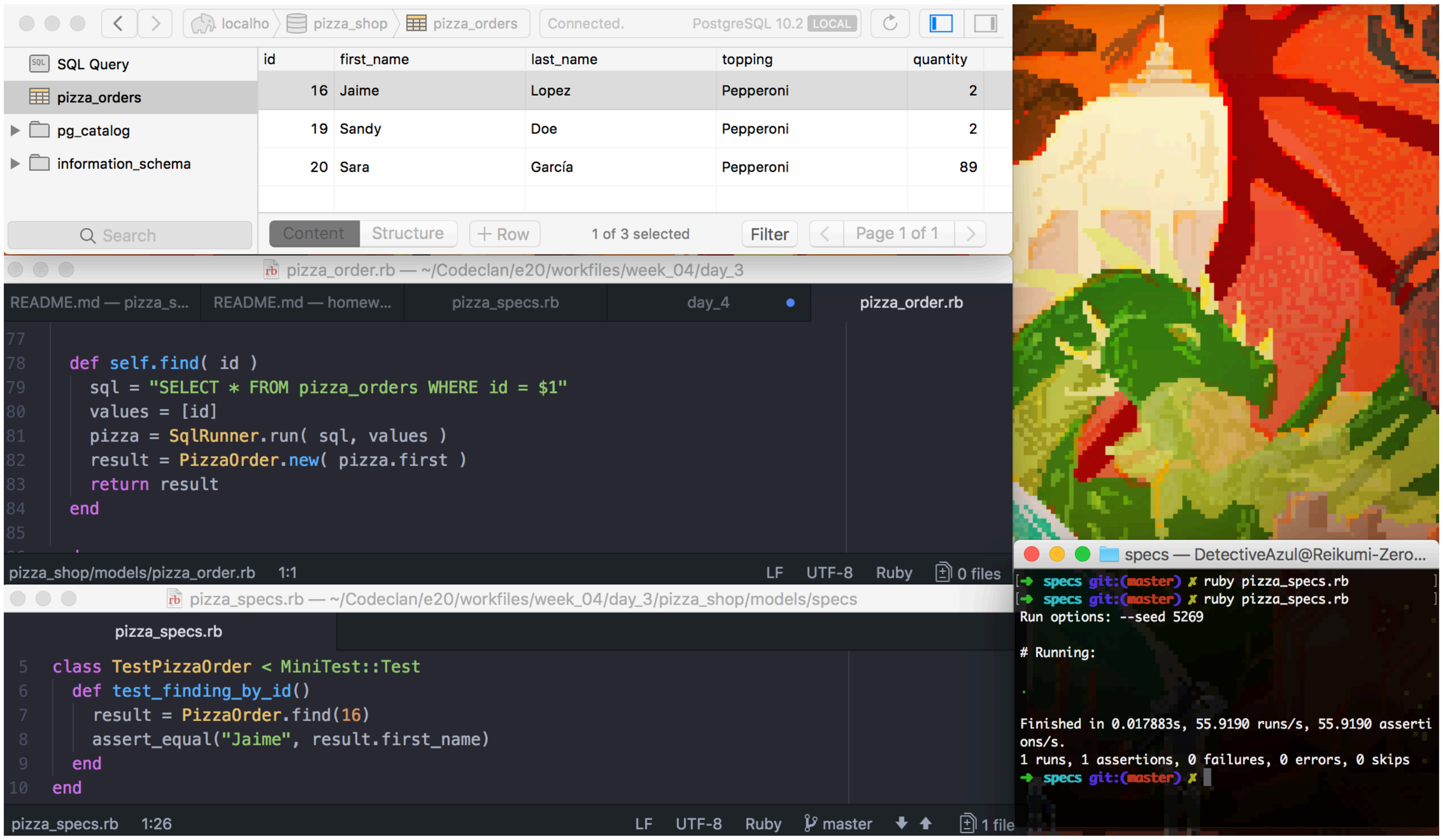
Screenshot showing the test that checks the call of the method getName() inherited from the parent class Character, and the test being passed



```java
        dagger = new Weapon( name: "Dagger",  value: 5,  damage: 5);

    }

    @Test
    public void hasName() { assertEquals( expected: "Gandalf", player.getName()); }
```

```
PlayerTest
Run:  PlayerTest.hasName
Tests passed: 1 of 1 test – 10 ms
▼ PlayerTest   10 ms
    hasName    10 ms
```

I.T 3 - Example of searching



id	first_name	last_name	topping	quantity
16	Jaime	Lopez	Pepperoni	2
19	Sandy	Doe	Pepperoni	2
20	Sara	García	Pepperoni	89

pizza_order.rb — ~/Codeclan/e20/workfiles/week_04/day_3

```ruby
  def self.find( id )
    sql = "SELECT * FROM pizza_orders WHERE id = $1"
    values = [id]
    pizza = SqlRunner.run( sql, values )
    result = PizzaOrder.new( pizza.first )
    return result
  end
```

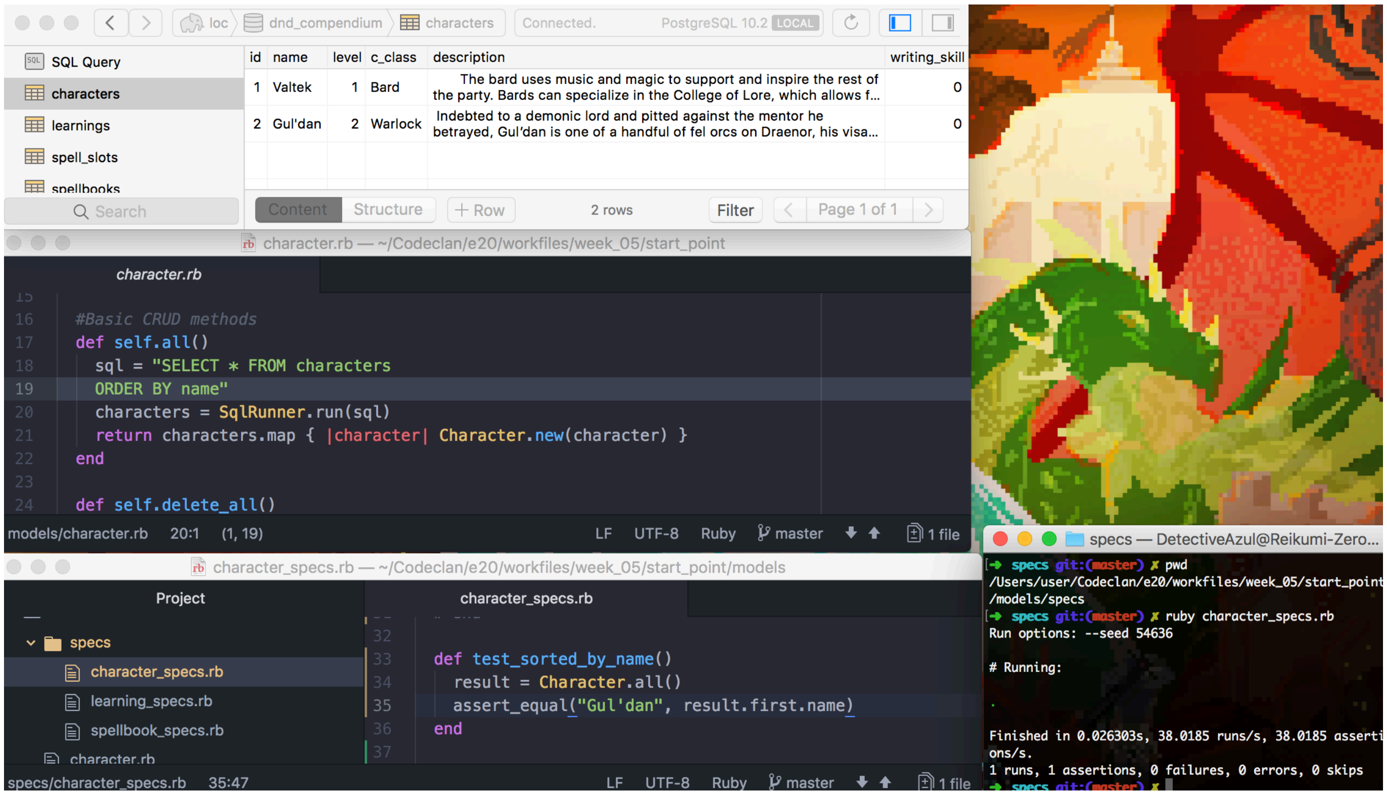
pizza_specs.rb — ~/Codeclan/e20/workfiles/week_04/day_3/pizza_shop/models/specs

```ruby
class TestPizzaOrder < MiniTest::Test
  def test_finding_by_id()
    result = PizzaOrder.find(16)
    assert_equal("Jaime", result.first_name)
  end
end
```

pizza_specs.rb 1:26
LF UTF-8 Ruby master 1 file
specs — DetectiveAzul@Reikumi-Zero...

```
➜ specs git:(master) ✗ ruby pizza_specs.rb
➜ specs git:(master) ✗ ruby pizza_specs.rb
Run options: --seed 5269

# Running:

.

Finished in 0.017883s, 55.9190 runs/s, 55.9190 asserti
ons/s.
1 runs, 1 assertions, 0 failures, 0 errors, 0 skips
➜ specs git:(master) ✗
```

I.T 4 - Example of sorting



id	name	level	c_class	description	writing_skill
1	Valtek	1	Bard	The bard uses music and magic to support and inspire the rest of the party. Bards can specialize in the College of Lore, which allows f...	0
2	Gul'dan	2	Warlock	Indebted to a demonic lord and pitted against the mentor he betrayed, Gul'dan is one of a handful of fel orcs on Draenor, his visa...	0

```ruby
  #Basic CRUD methods
  def self.all()
    sql = "SELECT * FROM characters
    ORDER BY name"
    characters = SqlRunner.run(sql)
    return characters.map { |character| Character.new(character) }
  end

  def self.delete_all()
```

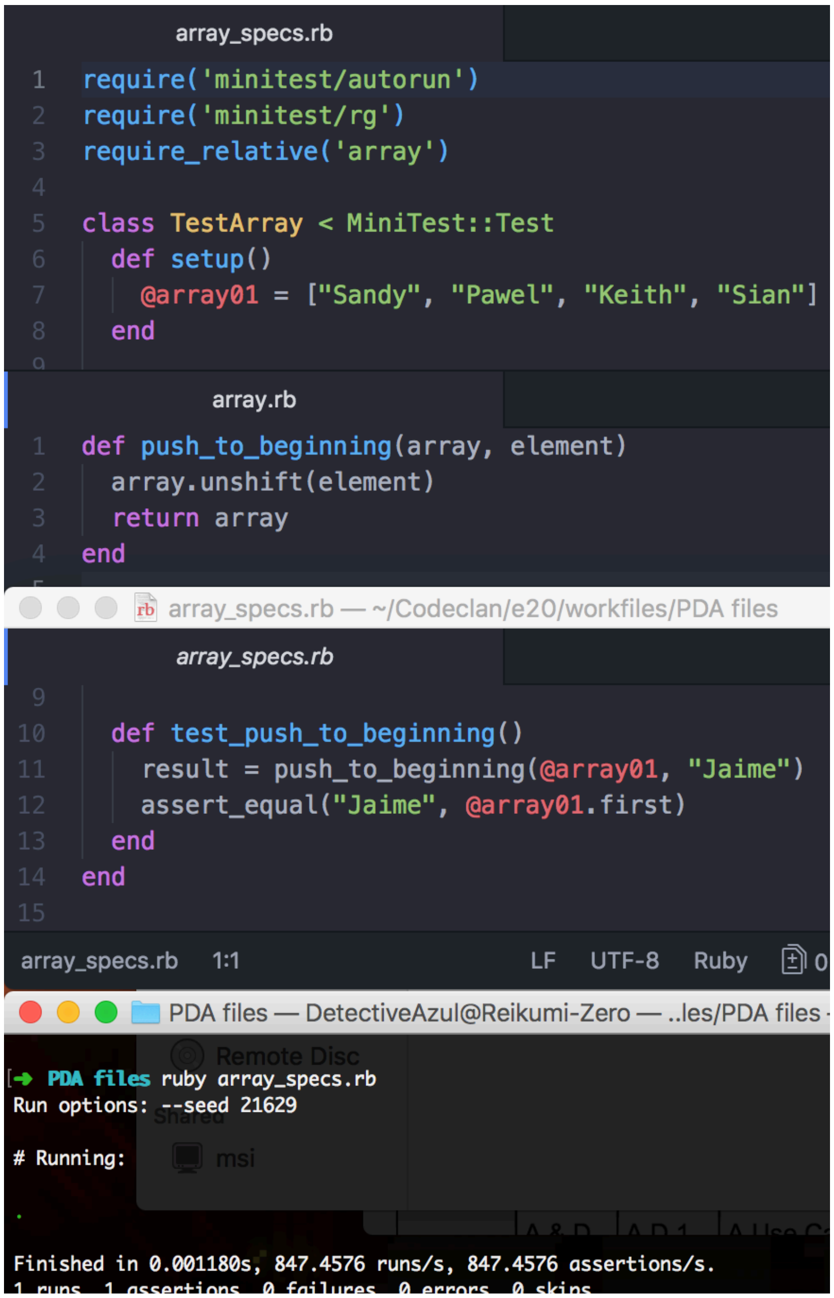
models/character.rb 20:1 (1, 19) LF UTF-8 Ruby master 1 file
character_specs.rb — ~/Codeclan/e20/workfiles/week_05/start_point/models
Project
specs
character_specs.rb
learning_specs.rb
spellbook_specs.rb
character.rb
character_specs.rb

```ruby
  def test_sorted_by_name()
    result = Character.all()
    assert_equal("Gul'dan", result.first.name)
  end
```

specs/character_specs.rb 35:47 LF UTF-8 Ruby master 1 file
specs — DetectiveAzul@Reikumi-Zero...

```
➜ specs git:(master) ✗ pwd
/Users/user/Codeclan/e20/workfiles/week_05/start_point
/models/specs
➜ specs git:(master) ✗ ruby character_specs.rb
Run options: --seed 54636

# Running:

.

Finished in 0.026303s, 38.0185 runs/s, 38.0185 asserti
ons/s.
1 runs, 1 assertions, 0 failures, 0 errors, 0 skips
➜ specs git:(master) ✗
```

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



array_specs.rb

```ruby
require('minitest/autorun')
require('minitest/rg')
require_relative('array')

class TestArray < MiniTest::Test
  def setup()
    @array01 = ["Sandy", "Pawel", "Keith", "Sian"]
  end
```

array.rb

```ruby
def push_to_beginning(array, element)
  array.unshift(element)
  return array
end
```

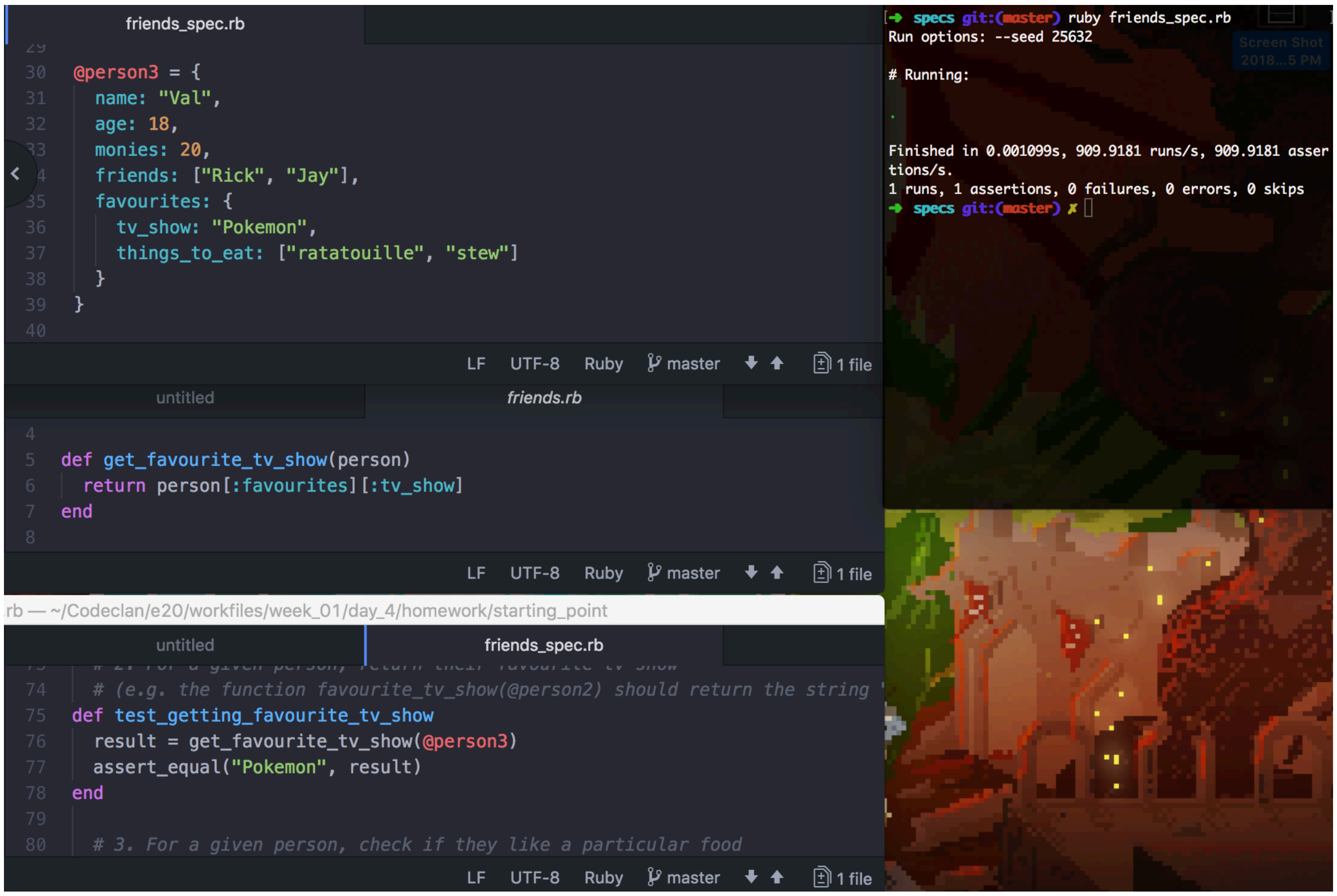
array_specs.rb — ~/Codeclan/e20/workfiles/PDA files

```ruby

  def test_push_to_beginning()
    result = push_to_beginning(@array01, "Jaime")
    assert_equal("Jaime", @array01.first)
  end
end

```

PDA files — DetectiveAzul@Reikumi-Zero — ..les/PDA files

```
➜ PDA files ruby array_specs.rb
Run options: --seed 21629

# Running:

.

Finished in 0.001180s, 847.4576 runs/s, 847.4576 assertions/s.
1 runs, 1 assertions, 0 failures, 0 errors, 0 skips
```

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



friends_spec.rb

```ruby
@person3 = {
  name: "Val",
  age: 18,
  monies: 20,
  friends: ["Rick", "Jay"],
  favourites: {
    tv_show: "Pokemon",
    things_to_eat: ["ratatouille", "stew"]
  }
}

```

LF UTF-8 Ruby master 1 file

untitled friends.rb

```ruby

def get_favourite_tv_show(person)
  return person[:favourites][:tv_show]
end

```

LF UTF-8 Ruby master 1 file

.rb — ~/Codeclan/e20/workfiles/week_01/day_4/homework/starting_point

untitled friends_spec.rb

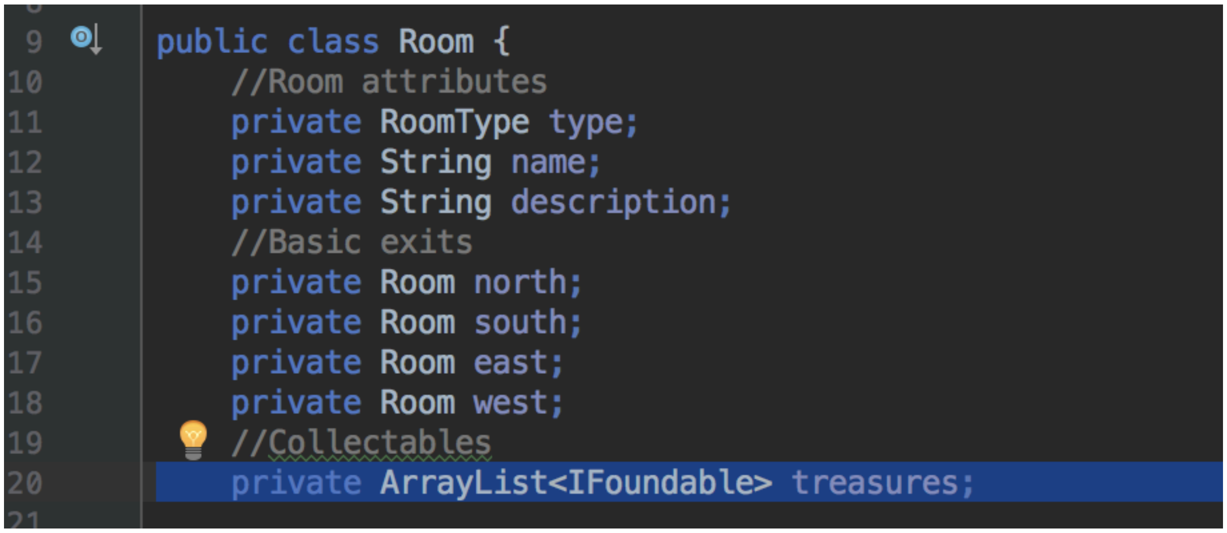
```ruby
  # 2. For a given person, return their favourite tv show
  # (e.g. the function favourite_tv_show(@person2) should return the string '
def test_getting_favourite_tv_show
  result = get_favourite_tv_show(@person3)
  assert_equal("Pokemon", result)
end

  # 3. For a given person, check if they like a particular food
```

LF UTF-8 Ruby master 1 file

```
➜ specs git:(master) ruby friends_spec.rb
Run options: --seed 25632

# Running:

.

Finished in 0.001099s, 909.9181 runs/s, 909.9181 asser
tions/s.
1 runs, 1 assertions, 0 failures, 0 errors, 0 skips
➜ specs git:(master) ✗
```

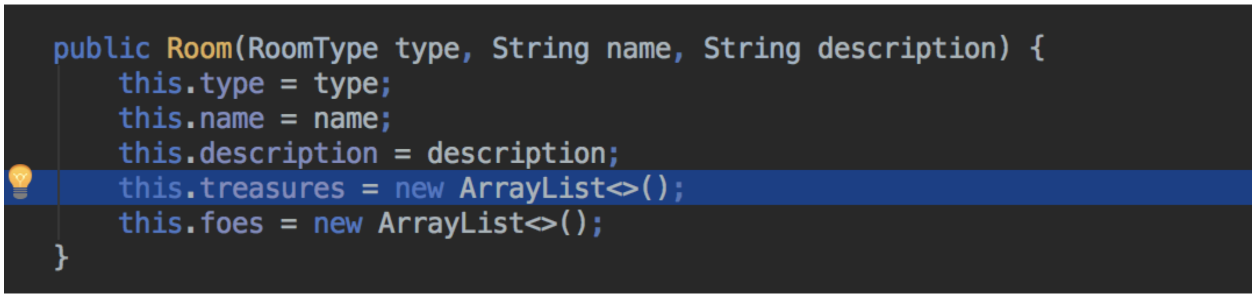
I.T 7 - Example of polymorphism in a program

A class that has an ArrayList of IFoundables



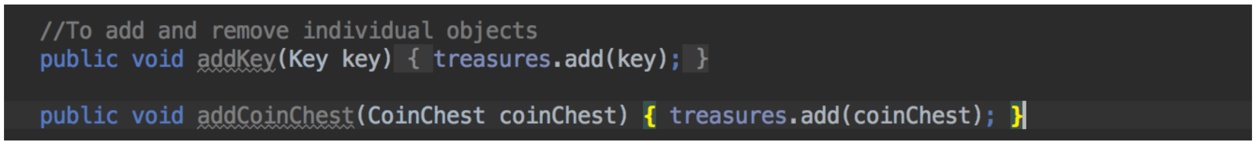
```java
public class Room {
    //Room attributes
    private RoomType type;
    private String name;
    private String description;
    //Basic exits
    private Room north;
    private Room south;
    private Room east;
    private Room west;
    //Collectables
    private ArrayList<IFoundable> treasures;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

The constructor, where we initiate the empty ArrayList of IFoundables



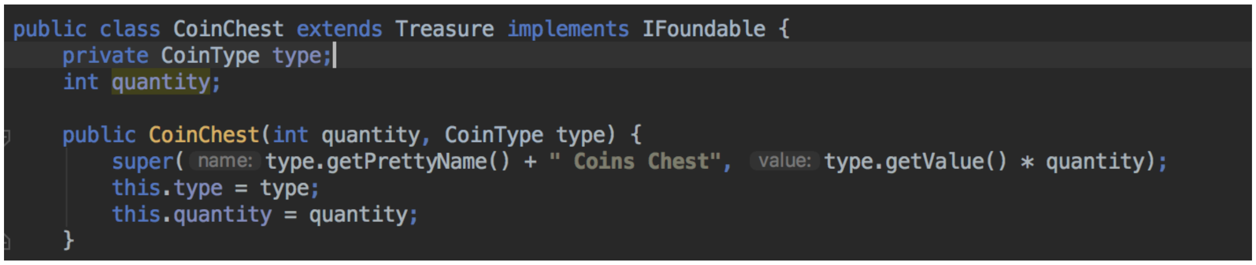
```java
public Room(RoomType type, String name, String description) {
    this.type = type;
    this.name = name;
    this.description = description;
    this.treasures = new ArrayList<>();
    this.foes = new ArrayList<>();
}
```

Different methods to add different kind of objects to this IFoundable array list

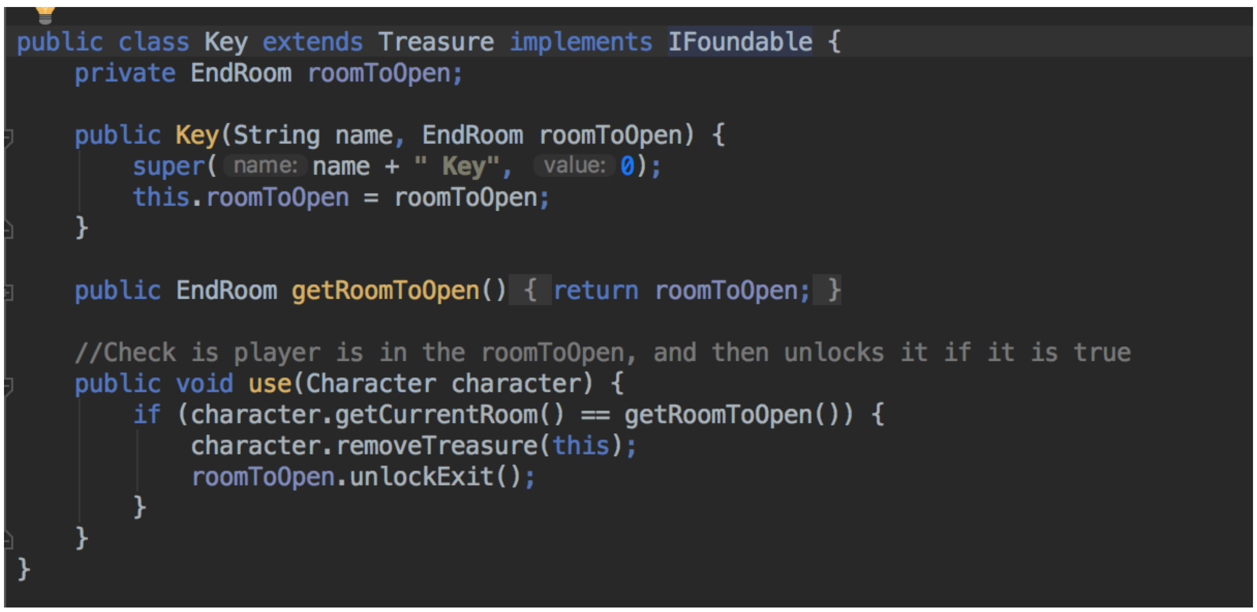


```java
//To add and remove individual objects
public void addKey(Key key) { treasures.add(key); }

public void addCoinChest(CoinChest coinChest) { treasures.add(coinChest); }
```

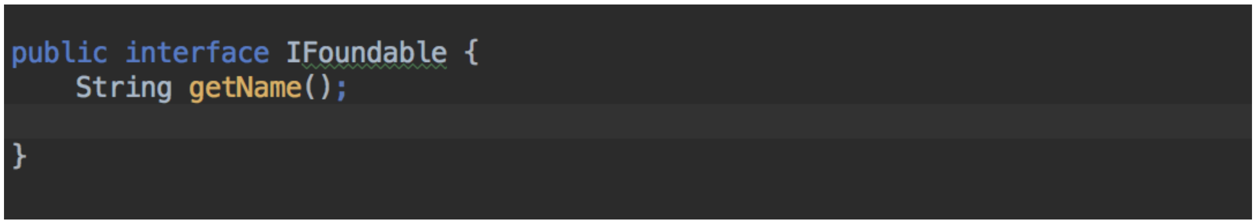
Examples of this two classes implementing the IFoundable interface



```java
public class CoinChest extends Treasure implements IFoundable {
    private CoinType type;
    int quantity;

    public CoinChest(int quantity, CoinType type) {
        super( name: type.getPrettyName() + " Coins Chest",  value: type.getValue() * quantity);
        this.type = type;
        this.quantity = quantity;
    }
```


```java
public class Key extends Treasure implements IFoundable {
    private EndRoom roomToOpen;

    public Key(String name, EndRoom roomToOpen) {
        super( name: name + " Key",  value: 0);
        this.roomToOpen = roomToOpen;
    }

    public EndRoom getRoomToOpen() { return roomToOpen; }

    //Check is player is in the roomToOpen, and then unlocks it if it is true
    public void use(Character character) {
        if (character.getCurrentRoom() == getRoomToOpen()) {
            character.removeTreasure(this);
            roomToOpen.unlockExit();
        }
    }
}
```

The IFoundable interface


```java
public interface IFoundable {
    String getName();

}
```