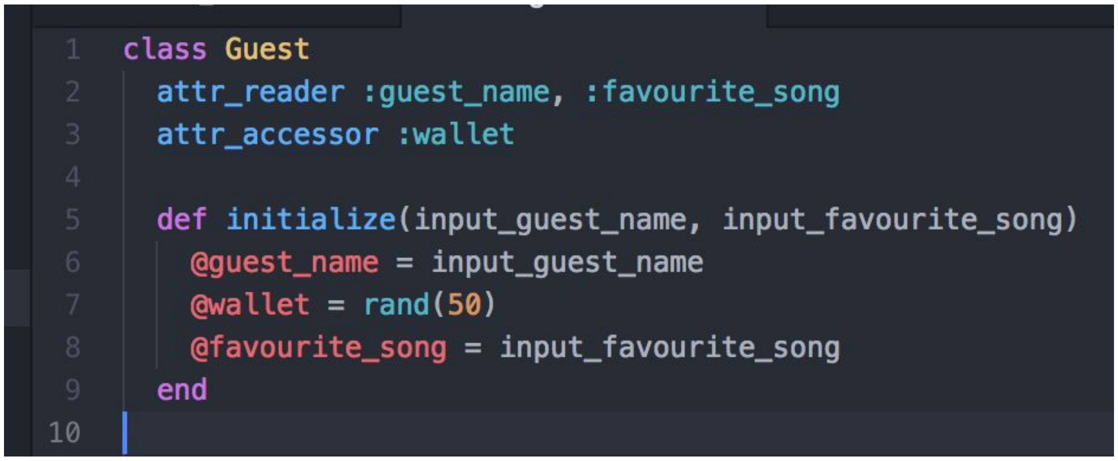
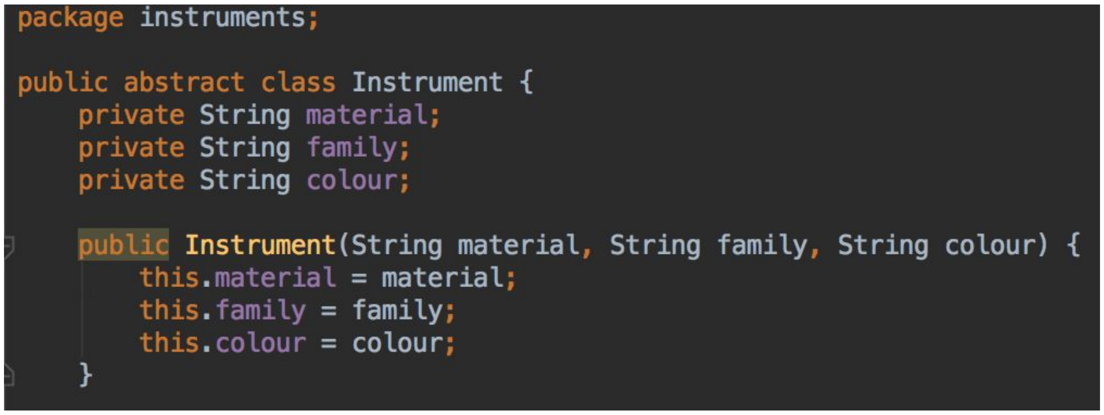
Phil Mitchell - PDA - Implementation & Testing Unit

I.T 1 - Encapsulation

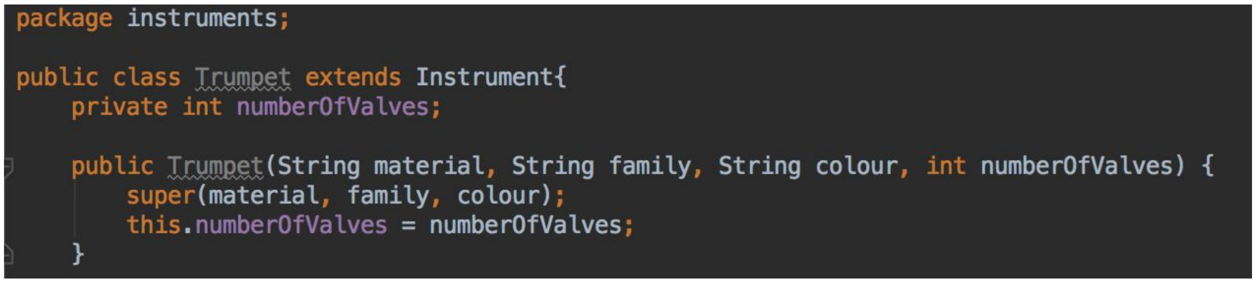


```ruby
class Guest
  attr_reader :guest_name, :favourite_song
  attr_accessor :wallet

  def initialize(input_guest_name, input_favourite_song)
    @guest_name = input_guest_name
    @wallet = rand(50)
    @favourite_song = input_favourite_song
  end

```

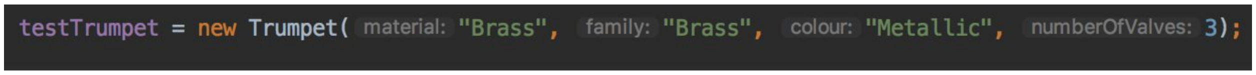
I.T 2 - Inheritance



```java
package instruments;

public abstract class Instrument {
    private String material;
    private String family;
    private String colour;

    public Instrument(String material, String family, String colour) {
        this.material = material;
        this.family = family;
        this.colour = colour;
    }
```

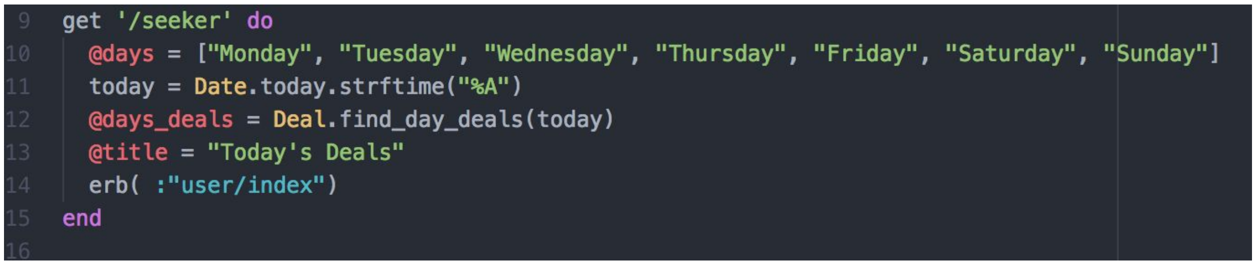


```java
package instruments;

public class Trumpet extends Instrument{
    private int numberOfValves;

    public Trumpet(String material, String family, String colour, int numberOfValves) {
        super(material, family, colour);
        this.numberOfValves = numberOfValves;
    }
```

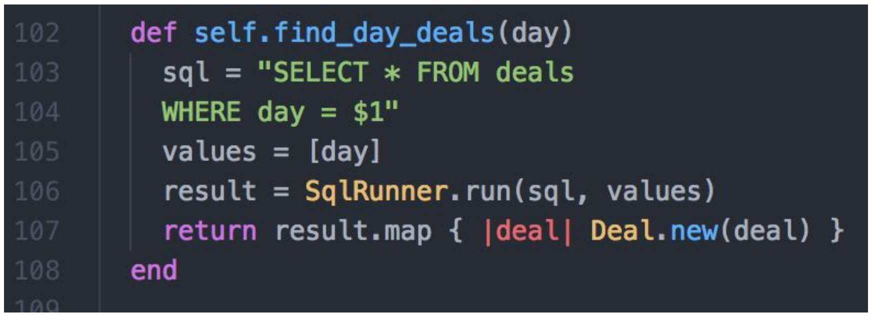


```java
testTrumpet = new Trumpet( material: "Brass",  family: "Brass",  colour: "Metallic",  numberOfValves: 3);
```

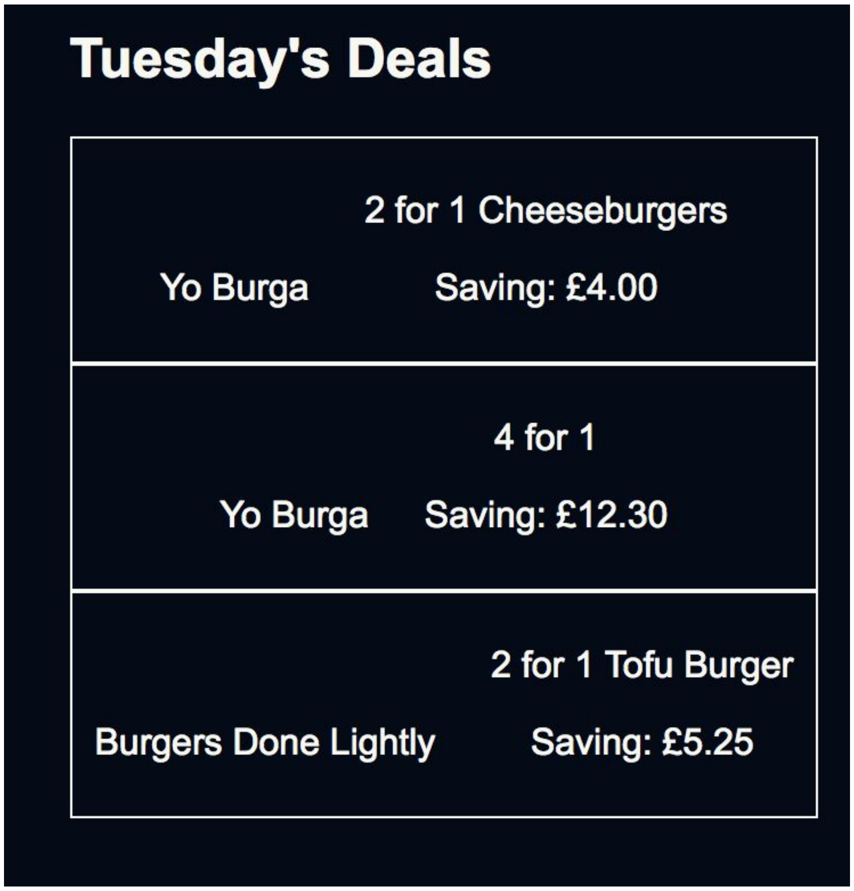
I.T 3 - Search Function & Result



```ruby
get '/seeker' do
  @days = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]
  today = Date.today.strftime("%A")
  @days_deals = Deal.find_day_deals(today)
  @title = "Today's Deals"
  erb( :"user/index")
end
```



```ruby
  def self.find_day_deals(day)
    sql = "SELECT * FROM deals
    WHERE day = $1"
    values = [day]
    result = SqlRunner.run(sql, values)
    return result.map { |deal| Deal.new(deal) }
  end
```



Tuesday's Deals

2 for 1 Cheeseburgers

Yo Burga Saving: £4.00

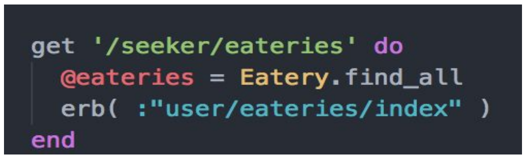
4 for 1

Yo Burga Saving: £12.30

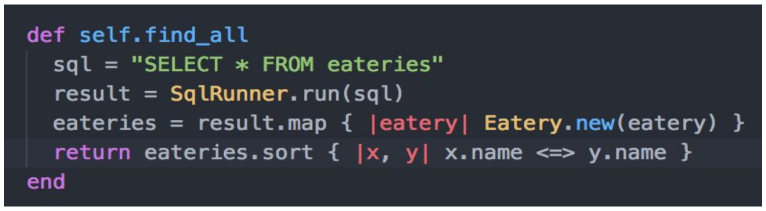
2 for 1 Tofu Burger

Burgers Done Lightly Saving: £5.25

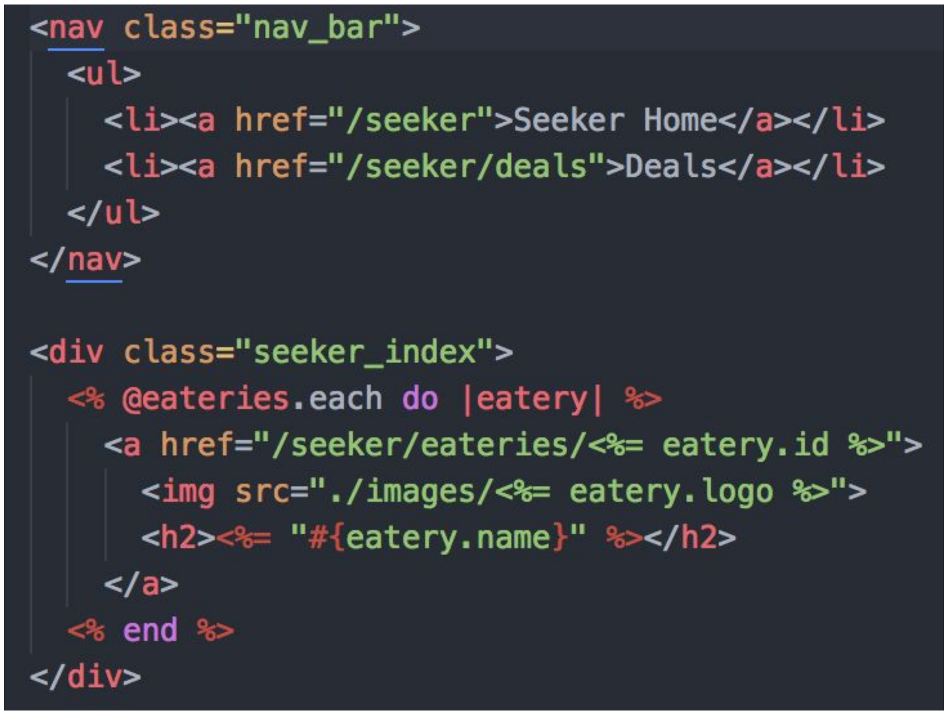
I.T 4 - Sort Function & Result



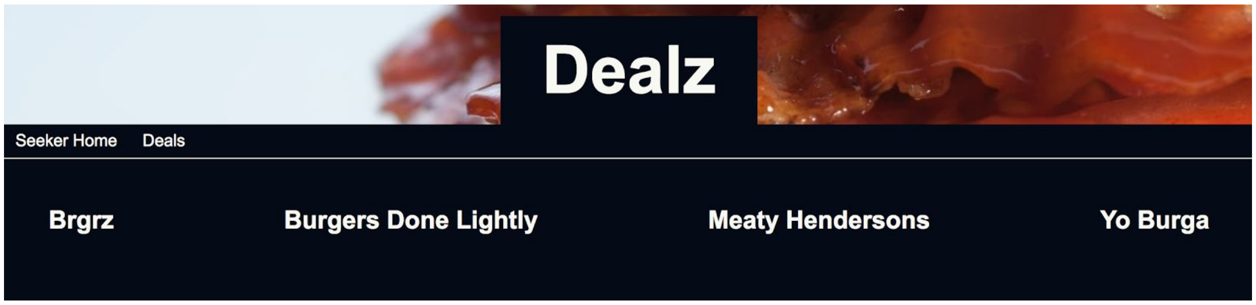
```
get '/seeker/eateries' do
  @eateries = Eatery.find_all
  erb( :"user/eateries/index" )
end
```



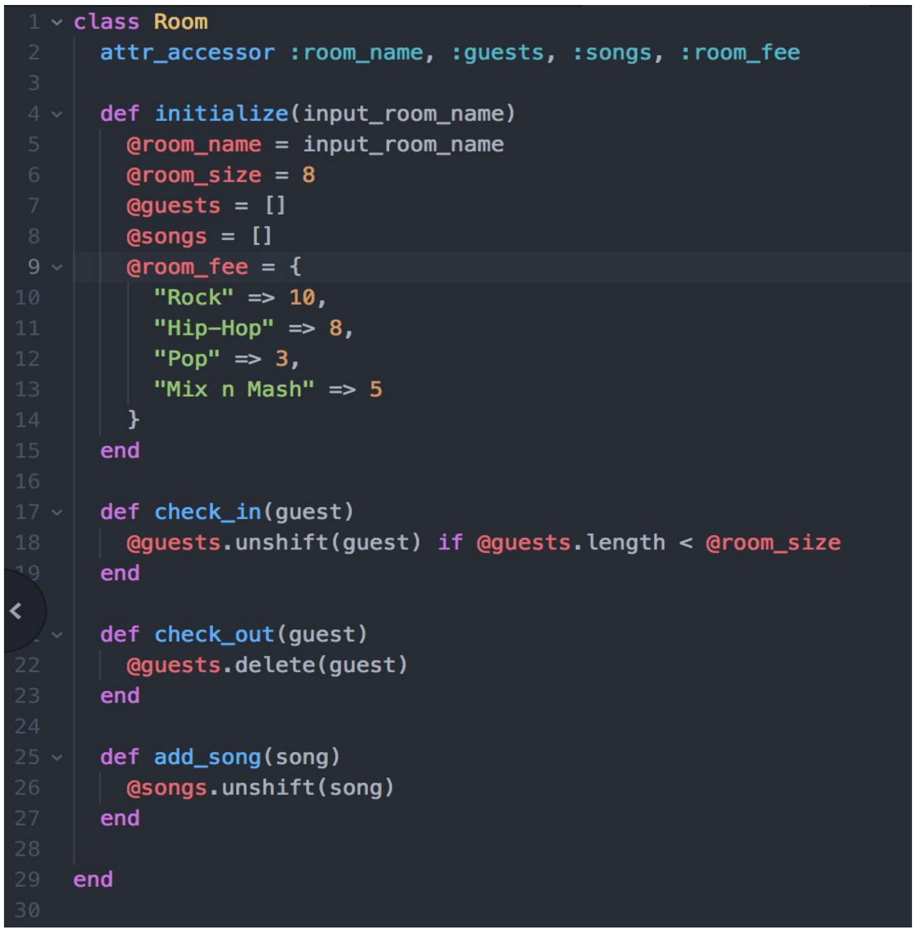
```
def self.find_all
  sql = "SELECT * FROM eateries"
  result = SqlRunner.run(sql)
  eateries = result.map { |eatery| Eatery.new(eatery) }
  return eateries.sort { |x, y| x.name <=> y.name }
end
```



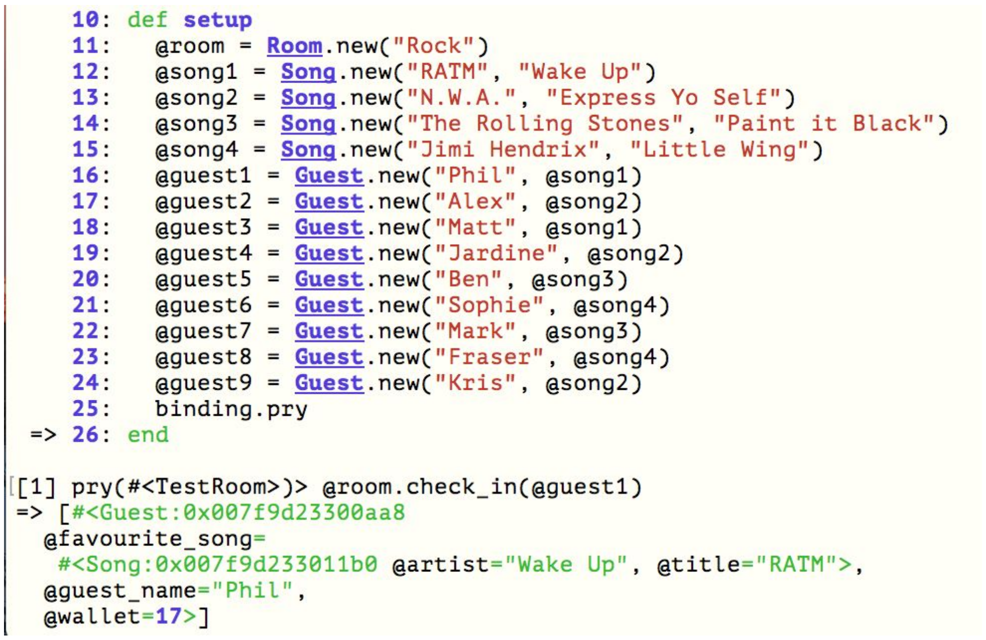
```
<nav class="nav_bar">
  <ul>
    <li><a href="/seeker">Seeker Home</a></li>
    <li><a href="/seeker/deals">Deals</a></li>
  </ul>
</nav>

<div class="seeker_index">
  <% @eateries.each do |eatery| %>
    <a href="/seeker/eateries/<%= eatery.id %>">
      <img src="./images/<%= eatery.logo %>">
      <h2><%= "#{eatery.name}" %></h2>
    </a>
  <% end %>
</div>
```



I.T 5 - Array

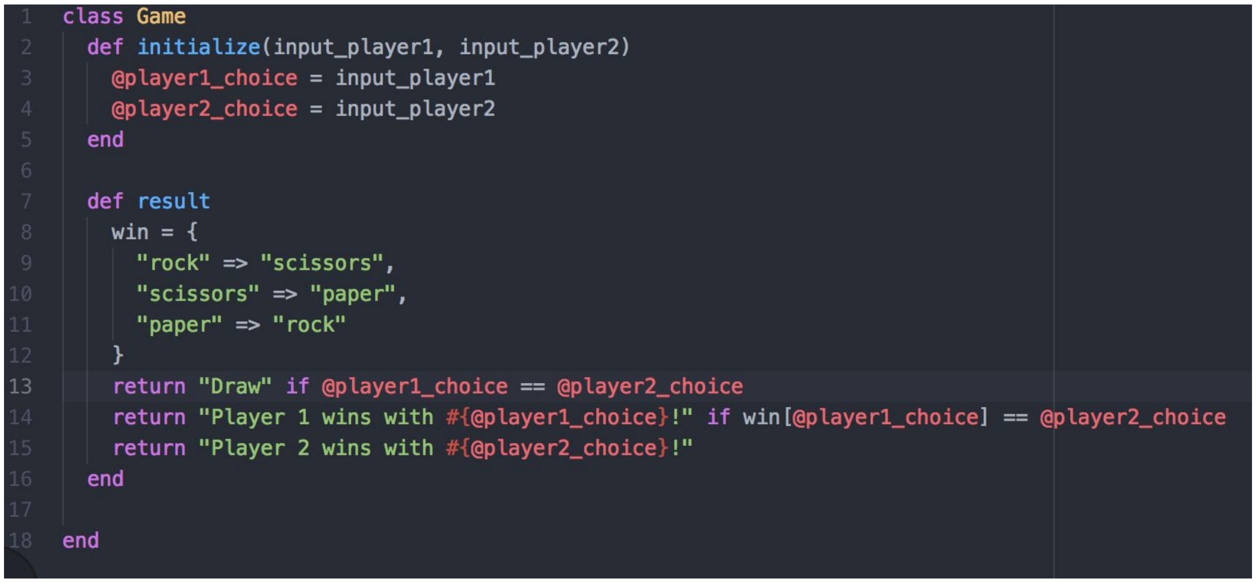


```ruby
class Room
  attr_accessor :room_name, :guests, :songs, :room_fee

  def initialize(input_room_name)
    @room_name = input_room_name
    @room_size = 8
    @guests = []
    @songs = []
    @room_fee = {
      "Rock" => 10,
      "Hip-Hop" => 8,
      "Pop" => 3,
      "Mix n Mash" => 5
    }
  end

  def check_in(guest)
    @guests.unshift(guest) if @guests.length < @room_size
  end

  def check_out(guest)
    @guests.delete(guest)
  end

  def add_song(song)
    @songs.unshift(song)
  end

end
```

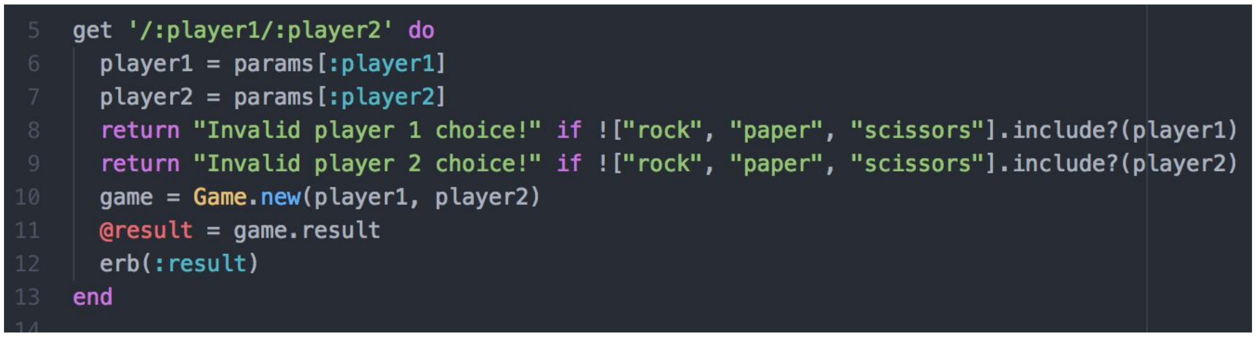


```
    10: def setup
    11:   @room = Room.new("Rock")
    12:   @song1 = Song.new("RATM", "Wake Up")
    13:   @song2 = Song.new("N.W.A.", "Express Yo Self")
    14:   @song3 = Song.new("The Rolling Stones", "Paint it Black")
    15:   @song4 = Song.new("Jimi Hendrix", "Little Wing")
    16:   @guest1 = Guest.new("Phil", @song1)
    17:   @guest2 = Guest.new("Alex", @song2)
    18:   @guest3 = Guest.new("Matt", @song1)
    19:   @guest4 = Guest.new("Jardine", @song2)
    20:   @guest5 = Guest.new("Ben", @song3)
    21:   @guest6 = Guest.new("Sophie", @song4)
    22:   @guest7 = Guest.new("Mark", @song3)
    23:   @guest8 = Guest.new("Fraser", @song4)
    24:   @guest9 = Guest.new("Kris", @song2)
    25:   binding.pry
 => 26: end

[1] pry(#<TestRoom>)> @room.check_in(@guest1)
=> [#<Guest:0x007f9d23300aa8
  @favourite_song=
   #<Song:0x007f9d233011b0 @artist="Wake Up", @title="RATM">,
  @guest_name="Phil",
  @wallet=17>]
```

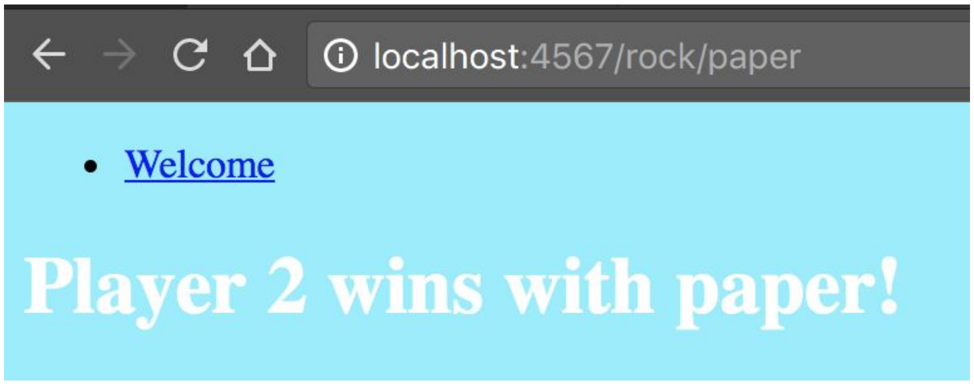
I.T 6 - Hash



```ruby
class Game
  def initialize(input_player1, input_player2)
    @player1_choice = input_player1
    @player2_choice = input_player2
  end

  def result
    win = {
      "rock" => "scissors",
      "scissors" => "paper",
      "paper" => "rock"
    }
    return "Draw" if @player1_choice == @player2_choice
    return "Player 1 wins with #{@player1_choice}!" if win[@player1_choice] == @player2_choice
    return "Player 2 wins with #{@player2_choice}!"
  end

end
```



```ruby
get '/:player1/:player2' do
  player1 = params[:player1]
  player2 = params[:player2]
  return "Invalid player 1 choice!" if !["rock", "paper", "scissors"].include?(player1)
  return "Invalid player 2 choice!" if !["rock", "paper", "scissors"].include?(player2)
  game = Game.new(player1, player2)
  @result = game.result
  erb(:result)
end
```

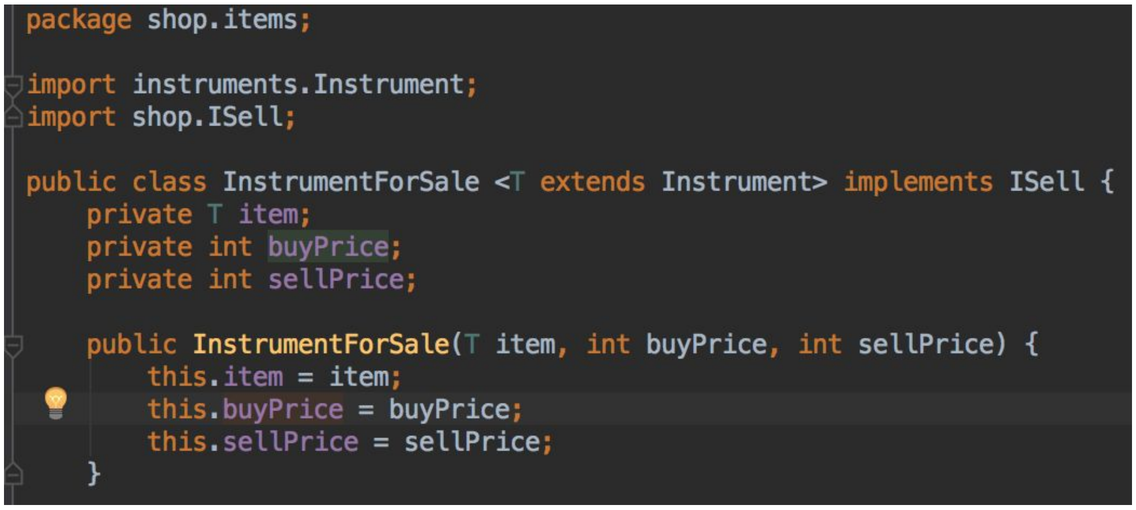


localhost:4567/rock/paper

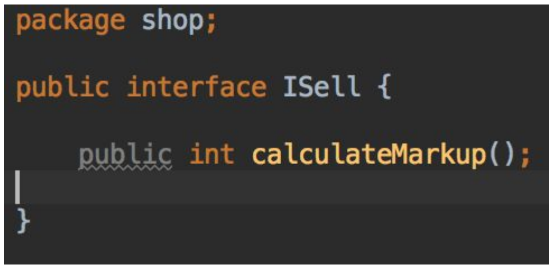
- Welcome

Player 2 wins with paper!

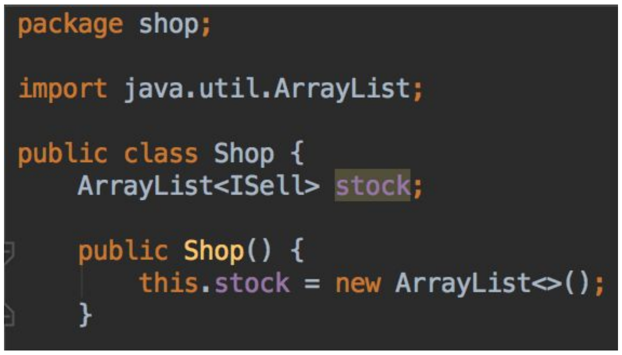
I.T 7 - Polymorphism



```java
package shop.items;

import instruments.Instrument;
import shop.ISell;

public class InstrumentForSale <T extends Instrument> implements ISell {
    private T item;
    private int buyPrice;
    private int sellPrice;

    public InstrumentForSale(T item, int buyPrice, int sellPrice) {
        this.item = item;
        this.buyPrice = buyPrice;
        this.sellPrice = sellPrice;
    }
```



```java
package shop;

public interface ISell {

    public int calculateMarkup();

}
```



```java
package shop;

import java.util.ArrayList;

public class Shop {
    ArrayList<ISell> stock;

    public Shop() {
        this.stock = new ArrayList<>();
    }
```