

T1A3 - Terminal Application

—

Pizza Store

Why a pizza store?

I chose a pizza store because it has real life purpose that could be used to help a business with getting orders online. One ethical issue is taking the users number so the store can contact the user.

Application Class's/Objects

- Menus
- Menu items
- Seed method
- Order
- Checkout
- Total price
- Estimated time of pick-up

```
src > seed.rb
1 require_relative './menu.rb'
2 require_relative './menu_items.rb' | You, 4 days ago • all menu items added to menu
3
4 def seed_pizza_menu
5
6   pizza_menu = Menu.new([MenuItems.new("Pepperoni", 12.0),MenuItems.new("Hawaiian", 12.0),MenuItems.new("Margarita", 12.0),MenuItems.new("Hawaiian", 12.0)])
7
8 end
9
10 def seed_sides_menu
11   sides_menu = Menu.new([MenuItems.new("Garlic Bread", 5.0),MenuItems.new("Chicken Nuggets", 5.0),MenuItems.new("Calzone", 7.0)])
12 end
13
14 def seed_drinks_menu
15   drinks_menu = Menu.new([MenuItems.new("Solo", 3.5),MenuItems.new("Coca-Cola", 3.5),MenuItems.new("Pepsi", 3.5),MenuItems.new("Sprite", 3.5)])
16 end
17
```

Functions

- User input
- Display Menus
- Add menu items to user's order
- Place an order and get a pickup time

The purpose of this application is to take a customer's order and display a pick up time and order total for their order.

Error handling

- Pry GEM

Pry gem allows you to start a pry/irb session at any point running an application with the syntax `binding.pry`. This helped me use the `.class` method to help understand what `tty-prompt` was doing with the return values.

```
33
34 #method that displays list of menu items
35 def select_pizza_menu(pizza_menu)
36   system "clear"
37   select_pizza = $prompt.select("Pizza Menu Options: ", [pizza_menu.print_menu, "Back to Main Menu"])
38   if $prompt.yes?("Would you like to add #{select_pizza} to you're order?") == true
39     select_pizza
40   else
41     main_menu
42   end
43
44   binding.pry
45   system "clear"
46 end
47
48 def select_sides_menu(sides_menu)
49   system "clear"
50   select_sides = $prompt.select("Would you like to add any sides? " [sides_menu.print_menu, "Back to Main Menu"])
51 end
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

From: /Users/calvindexter/Documents/coder/ruby-projects/CalvinDexter_T1A3/src/main.rb:44 Object#select_pizza_menu:

```
35: def select_pizza_menu(pizza_menu)
36:   system "clear"
37:   select_pizza = $prompt.select("Pizza Menu Options: ", [pizza_menu.print_menu, "Back to Main Menu"])
38:   if $prompt.yes?("Would you like to add #{select_pizza} to you're order?") == true
39:     select_pizza
40:   else
41:     main_menu
42:   end
43:
=> 44:   binding.pry
45:   system "clear"
46: end
```

[1] pry(main)> select_pizza
=> #<MenuItems:0x00000001332d5a80 @name="Pepperoni", @price=12.0>
[2] pry(main)> select_pizza.class
=> MenuItems
[3] pry(main)> █

Error handling

- rspec GEM

Using the rspec gem i tested the Menu and MenuItem's class's taking their desired inputs name and price

```
src > spec > basic_test_spec.rb
You, 2 hours ago | 1 author (You)
1 require_relative '../menu.rb'
2 require_relative '../menu_items.rb'
3
4
5 describe "Test MenuItem's" do
6   it "Should store the attributes of the MenuItem's" do
7     menu_items = MenuItem.new("Pepperoni", 12.0)
8     @name = "Pepperoni"
9     @price = 12.0
10    expect(menu_items.name).to eq(@name)
11    expect(menu_items.price).to eq(@price)
12  end
13 end
14
15 describe "Test Menu" do
16   it "Should store the array of the MenuItem's" do
17     menu_items = [MenuItem.new("Pepperoni", 12.0), MenuItem.new("Hawaiian", 12.0)]
18     pizza_menu = Menu.new(menu_items)
19     expect(pizza_menu.menu_items.length).to eq(2)
20   end
21 end
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

CalvinDexter_T1A3/src on `main` via `v3.0.2`

```
> rspec
..
```

Finished in 0.00175 seconds (files took 0.057 seconds to load)
2 examples, 0 failures

Review

Development/Build process

- Challenges
- Favourite Parts

