Craig Muth

12/5/2022

IT FDN 110 A Au 22: Foundations Of Programming: Python

Assignment 08

<<https://github.com/cjmuth/IntroToProg-Python-Mod08>>

Classes and Objects

# Introduction

The goal of this project is …

A partial program has been provided to start from. All the necessary structure exists, but operational code has been omitted in several places so it will not run as it currently exists. So we will need to map the logic for the existing code, identify where it is lacking, and develop the logic and code to make it work.

# Designing the program

Examining the starter file, the logic flow looks like this - with areas where code is missing is indicated.

* Data
  + declare variables and constants
  + class Product
    - *MISSING CODE*
* Processing
  + class FileProcesser
  + Process data from a file
    - *MISSING CODE*
  + Process data to a file
    - *MISSING CODE*
* Presentation (Input/Output)
  + class IO
    - def print\_menu\_items
      * print list of options to screen
    - Get user’s choice
      * *MISSING CODE*
    - Show the current data from the file to user
      * *MISSING CODE*
    - Get product data from the user
      * *MISSING CODE*
* Main Body of Script
  + Load data from file into a list of product objects when script starts
    - *MISSING CODE*
  + Show user a menu of options
    - *MISSING CODE*
  + Get user’s menu option choice
    - *MISSING CODE*
    - Show user current data in the list of product objects
      * *MISSING CODE*
    - Let user add data to the list of product objects
      * *MISSING CODE*
    - Let user save current data to file and exit program
      * *MISSING CODE*

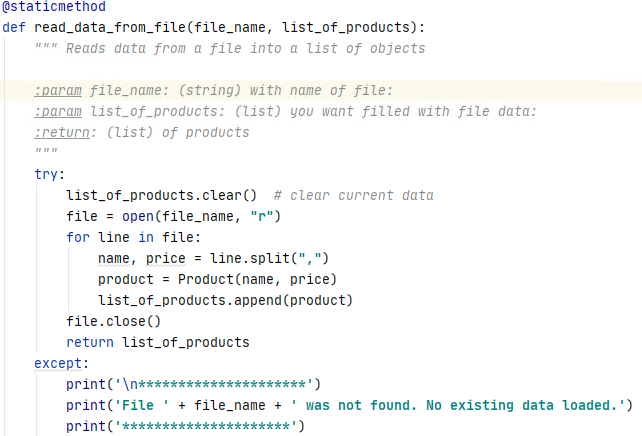
As the program is already divided into classes, we will address the updates for each class separately to make it easier to follow the changes as they are introduced.

### The Product class

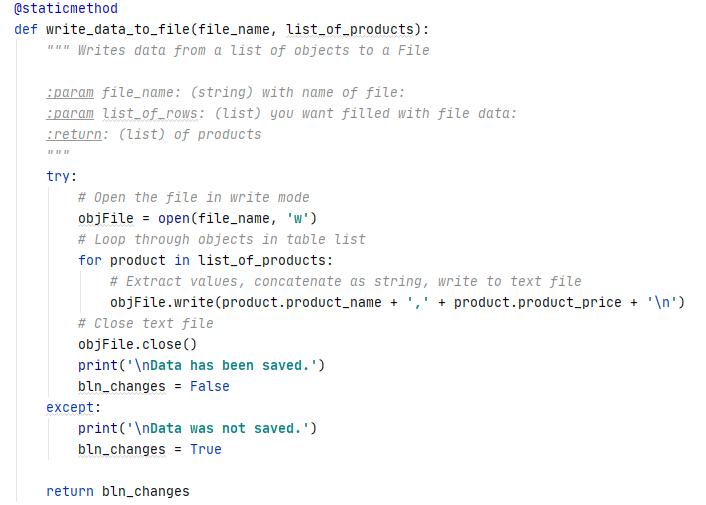
* class Product
  + MISSING CODE

### The FileProcessor class

* + Process data from a file
    - Receive file name and product list from main program
    - If file exists
      * Open file in read mode
      * Read line from file
      * Call Product class to create object
      * Add object to product list
      * If not end of file
        + Go to Read line from file
      * Else
        + Close file
        + Return product list to main program



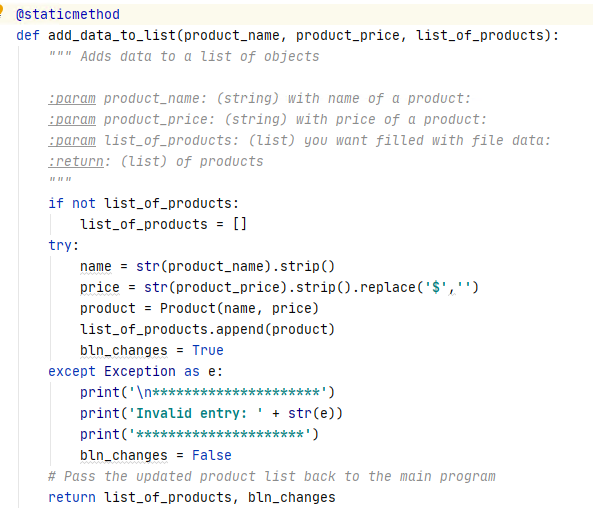
* + Process data to a file
    - Receive file name and product list from main program
    - Open file in write mode
    - Get object from list
      * Write product name and price to file
      * If not end of list
        + Go to Get object from list
      * Else
        + Close file



### The DataProcessor class

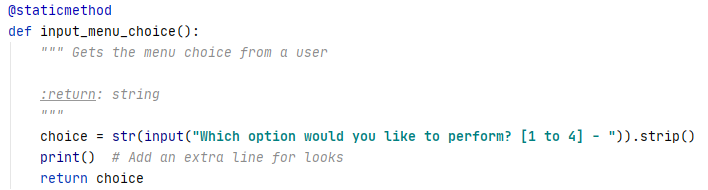
This class was not in the original file, but was added because the functionality doesn’t quite fit with the methods in the FileProcesser class.

* Add data to list
  + Receive part name, price, and product list from main program
  + Remove dollar sign from price if present
  + Call Product class to create object
  + Add object to product list
  + Set change flag to True
  + Return product list and change flag to main program

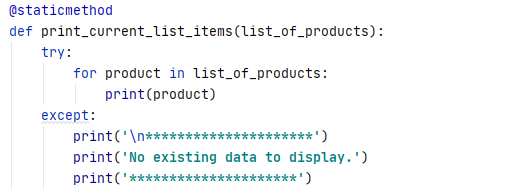


### The IO class

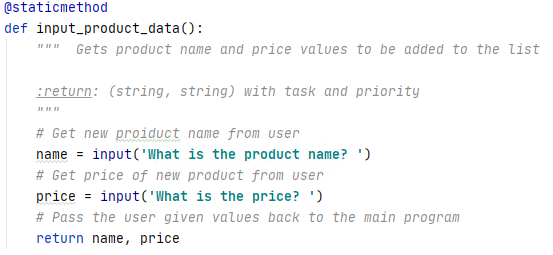
* + - Get user’s choice
      * choice = str(input("Which option would you like to perform? [1 to 4] - ")).strip()
      * Return choice to main program



* + - Show the current data from the file to user
      * Receive product list
    - Get object from list
    - print(product)
    - if not end of file
      * Go to Get object from list



* + - Get product data from the user
      * Request product name from user
      * Request product price from user
      * Return name and price to main program



### The main body of script

* Load data from file into a list of product objects when script starts
  + Call FileProcessor.read\_data\_from\_file and assign value to lstOfProductObjects
* Show user a menu of options
  + Call IO.print\_menu\_items
* Get user’s menu option choice
  + Call IO.input\_menu\_choice
  + If show current data in the list of product objects
    - Call IO.print\_current\_list\_items
  + If add data to the list of product objects
    - Call IO.input\_product\_data
    - Call DataProcessor.add\_data\_to\_list
  + If save current data to file
    - FileProcessor.write\_data\_to\_file
  + If exit program
    - If change flag is True
      * Notify user of unsaved changes and give option to save
      * If Yes
        + Call FileProcessor.write\_data\_to\_file
      * Else
        + Display message that data was not saved
      * Close program
  + Else
    - Display invalid selection message
    - Go to Get user’s menu option choice



# Running the program

### Executing the program in Pycharm:

### Executing in a Terminal window:

# Summary

In this project