Jacob Bentzen

June 6th, 2022

Foundations Of Programming: Python

Assignment08

Classes and Components

**Introduction**

In assignment 8, I was tasked with working on a similar data processing project as before, but using a more organized format with classes, constructors, fields, attributes, properties, and methods.

**Understanding the Psuedo-Code:**

Understanding the Assignment:

The first step of the assignment is to review the pseudo code and understand what I will need to complete and how it should be completed. As you can see in fig. 1, there is a substantial amount of code to add to the file and essentially I will be creating it from scratch.

Text

Description automatically generated

*Fig. 1: Header and part of the body of the main code*

All methods for the Product, fileProcessor, and IO class all need to be created and added to the code. However, I already can see that the format is similar to assignment 6 and I can use some of the code I developed for that assignment in this one.

**Writing the Code:**

Product Class:

For the product class, I added in methods for the two data points that I would be utilizing from the user: product and product price (Fig. 2).

Text

Description automatically generated

*Fig. 2: The methods (Getters and setters) for the product and product price in the Product class.*

The class also includes the constructor for the name and price variables. The constructor and the properties have been tied together so that when data is changed in one it is changed in all places.

I also included exceptions in the setters for both name and price, to make sure that the user does not input data that does not match what the variable should be.

FileProcessor Class:

There are three methods in the FileProcessor class: read\_data\_from\_file, write\_data\_to\_file, and add\_data\_to\_list (Fig. 3).

Text

Description automatically generated

*Fig. 3: Methods in the FileProcessor Class.*

Some of this code was recycled from assignment 6, but some of it was unique to this assignment. When utilizing the Product methods to store data, it changes the way that I am interacting with that stored data. That means when I read or write the data to the file I am now using a Product object instead of a dictionary like last time.

So, when I add something to the list, it is now the Product object as in the first method in fig 3.

IO Class:

There are 4 methods within the IO class: output\_menu\_tasks, input\_menu\_choice, print\_data\_from\_list, and input\_new\_product\_and\_price(Fig. 4).

Text

Description automatically generated

*Fig. 4: Methods in the IO Class.*

Again, some of this code was adjusted for this program from assignment 6, but the input\_new\_product\_and\_price class was made specifically with the Product class in mind. The products are received from the user and then added to a Product object instead of a dictionary object. This allows them to go through the Product methods first before they are added to the data list.

Main Body of the Script:

This code is again very similar to what was displayed in assignment 6 (Fig. 5).

Text

Description automatically generated

I utilized the classes by calling them wherever needed and adjusted the options so they matched the requirements of the pseudo code. Overall this is just a while loop running through the options and the main code is occurring in the classes above.

**Testing the Script**

Using PyCharm:

Since the program was written in PyCharm, it is straightforward to run and test the completed program (Fig. 7).

Text

Description automatically generated

*Fig. 7: Testing the program in the PyCharm run window.*

Using the Command Window:

After running the script in PyCharm, I made sure to test it using the Command Window on my computer (Fig. 8).

Text

Description automatically generated

*Fig. 8: Command Window with the Assignment 7 Program completed.*

**Summary**

I was able to complete this program by utilizing my understanding of classes and the proper way to organize them in object oriented programming.

Additionally, here is my GitHub link:

<https://github.com/bentzj2/ITFnd100-Mod07/blob/main/docs/index.md>