Sara Tupponce

05/27/2021

IT FDN 110 A Sp21: Foundations of Programming: Python

Assignment 08

GitHub URL: <https://github.com/Ssara-Tt/IntroToProg-Python-Mod08>

Objects and Classes

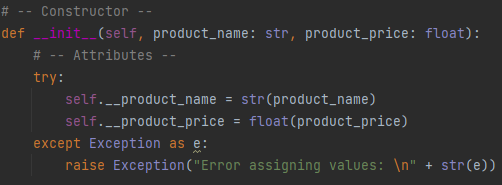
## Introduction

The goal for this assignment was to modify a starter script to show how to use classes indirectly for storing data and directly for processing the data. When creating a class for indirect use, the Fields, Constructors, Properties, and Methods for that class had to be defined. By using the instance method, the object is defined beforehand and an instance of the object is called later. By using the @staticmethod directive, the method can be called directly from the class without needing to make an object first or calling the function from an object instance.

## Creating Python Script

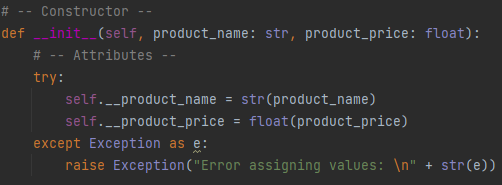
Data Layer

Using the provided starter script with the pseudocode, code was added to the script in the data section to define the Product class. Since this class would be called indirectly with independent instances, Constructor, Properties, and Fields sections had to be included. The Constructor was created first using variables “product\_name” and “product\_price”. Error handling was included in this section to print an error if it was not working correctly. The code can be seen below in Figure 1: Constructor.



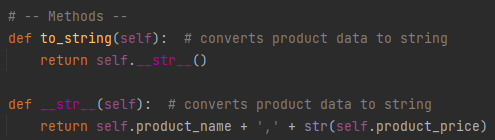
**Figure 1: Constructor**

The Properties section was then created, with one getter and one setter for each variable. Since individual instances of the Product class would be called, the word “self” was included in each definition. Custom error messages were also included for each setter since it would take in information from the user, which could contain user errors. The code can be seen below in Figure 2: Properties.



**Figure 2: Properties**

Next, the Methods section was created, which was for converting the product data into a string. Two different ways to do this were shown, with the “to\_string” function referencing the “\_\_str\_\_” function. The code can be seen below in Figure 3: Methods.



**Figure 3: Methods**

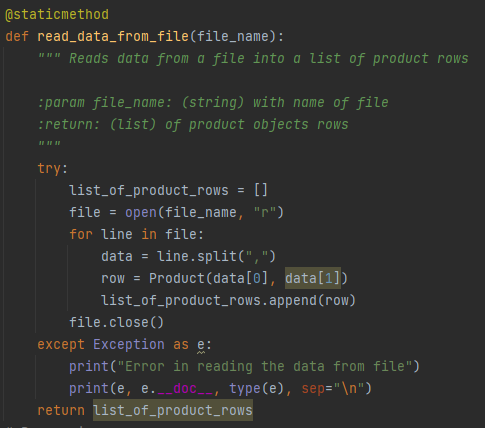
Processing Layer

The processing layer contained the “FileProcessor” class, which would process the data from the data layer and use the static method. Because of this, only Methods section was needed for the functions. The function “save\_data\_to\_file” was for writing the data to the text file. The code in this function is a modified version of what was in the previous assignment since it also included the same function. The names of the parameters changed. Also, the “\_\_str\_\_” method was used when writing the data to the file to avoid a data type error. The function code can be seen below in Figure 4: “save\_data\_to\_file” Function.



**Figure 4: “save\_data\_to\_file” Function**

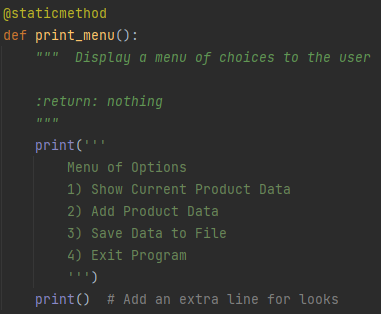
The function “read\_data\_from\_file” was also a modified version of the function from the previous assignment and was for reading the data from a file and saving it into a list of product rows. The file designated in the parameter was opened in read mode and each line of data was appended to the returned variable “list\_of\_product\_rows”. The function code can be seen below in Figure 4: “read\_data\_from\_file” Function.



**Figure 5: “read\_data\_from\_file a” Function**

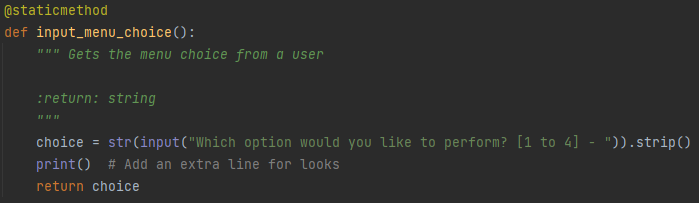
Input/Output Layer

The Input/Output section contained the IO class and the functions that displayed info to the user or captured user inputs. The functions in this section would also be called using the static method. As in previous sections, the functions in this layer were also modified versions of functions that had been used in previous assignments. The “print\_menu” function displayed a menu of modified choices to the user. The code can be seen below in Figure 5: “print\_menu” Function.



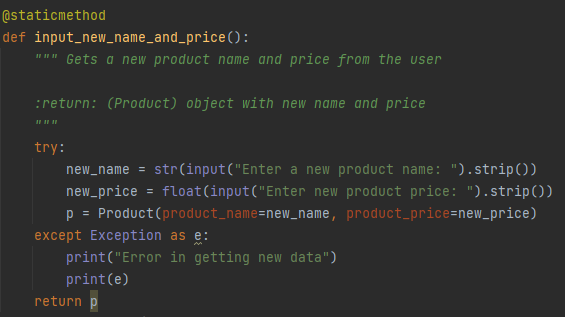
**Figure 6: “print\_menu” Function**

The function “input\_menu\_choice” was also a previously existing function that captured the user input for what menu option they wanted to choose. This function had minimal modifications. The code can be seen below in Figure 5: “input\_menu\_choice” Function.



**Figure 7: “input\_menu\_choice” Function**

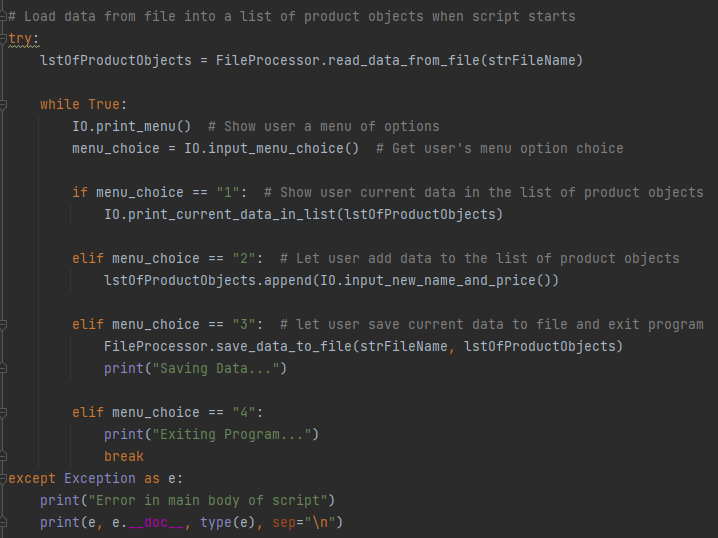
The function “input\_new\_name\_and\_price” was to capture the user input for a new product name and product price. An instance of the Product class was then called and saved in the variable “p”. The user inputs were assigned to the variables defined in the class. A custom error message was also included in this section. The code can be seen below in Figure 8: “input\_new\_name\_and\_price” Function.



**Figure 8: “input\_new\_name\_and\_price” Function**

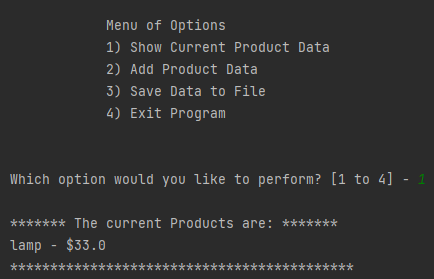
Main Body of Script

The main body of the script of where the functions that were created in the other sections were called based on the menu choice made by the user. As in previous assignments, the if statements connected to each menu choice were nested inside a while loop so that the menu choice would continue to show up until the user made the selection to exit the program. A try/except block was before the loop and gave a custom error saying that there was an error inside the main body of the script. The code can be seen below in Figure 9: Main Body.



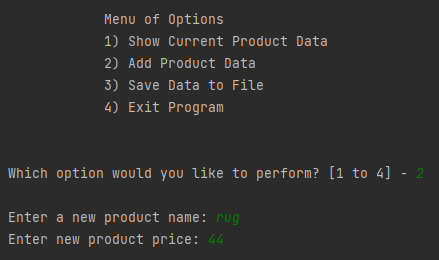
**Figure 9: Main Body**

The fist menu choice was to allow the user to see the current data in the list of product objects. This used the “print\_current\_data\_in\_list” function from the IO class. The results of running this section in PyCharm can be seen below in Figure 10: PyCharm – Choice 1.



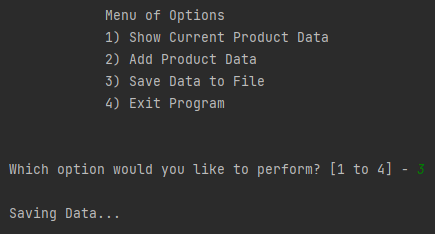
**Figure 10: PyCharm – Choice 1**

The second menu choice was to allow the user to input a new product name and product value. The new product name “run” and product price “44” were added in this example. The results of running this section in PyCharm can be seen below in Figure 11: PyCharm – Choice 2.



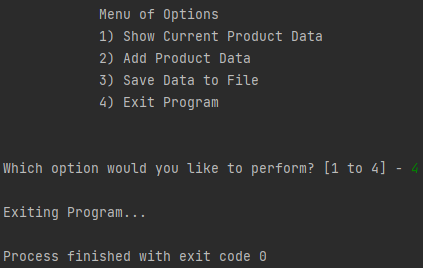
**Figure 11: PyCharm – Choice 2**

The third menu choice was to allow the user to save the data to a text file. A simple print statement indicated to the user that the data was being saved. The results of running this section in PyCharm can be seen below in Figure 12: PyCharm – Choice 3.



**Figure 12: PyCharm – Choice 3**

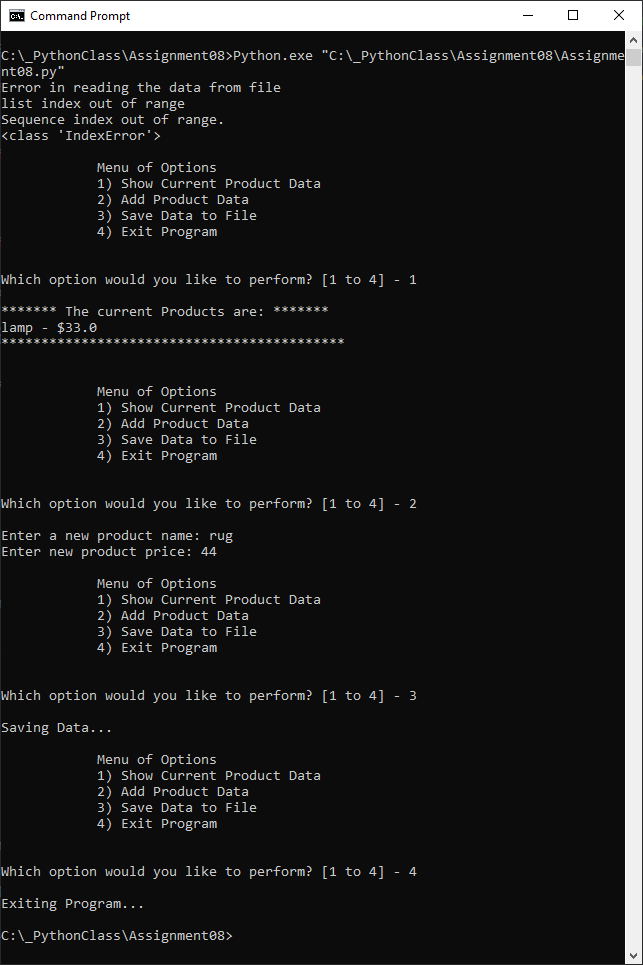
The last menu choice was to allow the user to exit the program. The results of running this section in PyCharm can be seen below in Figure 13: PyCharm – Choice 5.



**Figure 13: PyCharm – Choice 5**

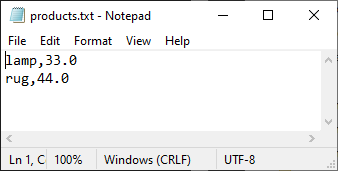
Test

In addition to the incremental tests done in PyCharm, an additional test was done by running the script in the Command Window. The results can be seen below in Figure 14: Command Window Running Script.



**Figure 14: Command Window Running Script**

The resulting text file with the saved data can also be seen in Figure 15: Text File.



**Figure 15: Text File**

## Summary

The difficulty of this assignment increased a fair amount when compared to the previous assignments. The concepts were more difficult to understand, especially the difference between all the parts of the classes that had to be defined. However, the video posted to help with the homework helped greatly with understanding how to practically apply what had been learned in class video.