Chanel M. Leung

June 07, 2020

Foundations of Programming: Python

Assignment 08

https://github.com/cml-python/ITFnd100-Mod07

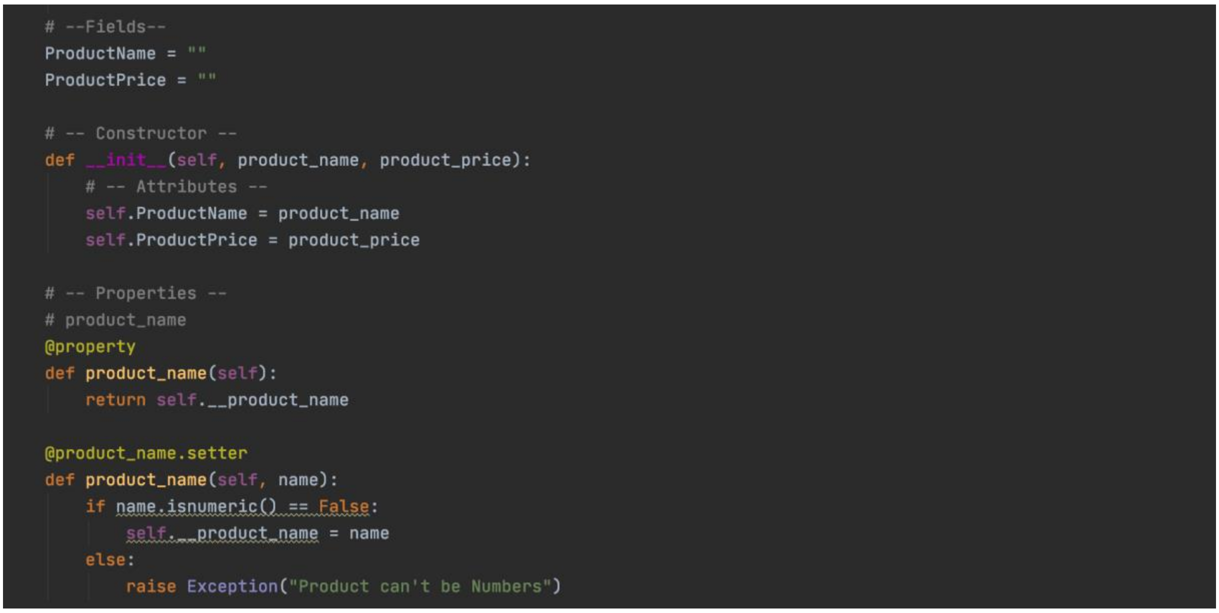
Software Objects

Introduction

This module helped me use object-oriented programming techniques to create a script that would be able to capture a product and price to save the data into a text file. I learned more about classes and the patterns that it takes where this module taught about fields, constructors, attributes, properties and methods. I was able to learn about setters and getters which would probably utilize in the script. This script that I will create will do the above and this will outline the different techniques I used while creating the script.

Class Patterns

Diving more into depth on classes, I was able to lab by lab learn how to use them, first fields that would hold a name to data, constructors are “setters” that help set the initial value of the objects, attributes aren’t really seen in the output but hold the data within the script itself, properties manage the fields that were created in the first place and methods are used to help organize the other functions inside. In the script I tried using it in the beginning of the script to help set up the class Product which I would use at some points throughout the script. I first set the constructor with self attributes such as product_name and product_price. then in for properties I set the product name to I wanted to call it when typing it in the script what I wanted it to as well as what I wanted to set that to where I also added error handling if the user typed in numbers or letters for both product name and price (Figure 1).

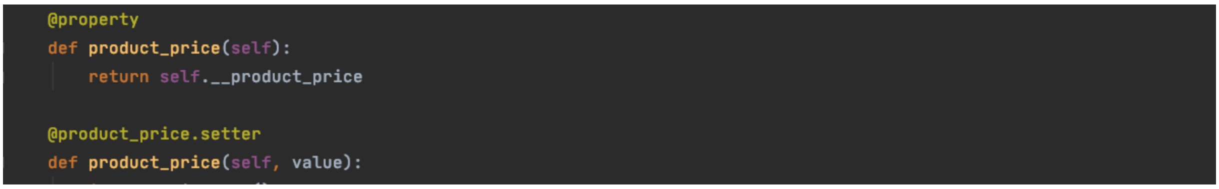


```
# --Fields--
ProductName = ""
ProductPrice = ""

# -- Constructor --
def __init__(self, product_name, product_price):
    # -- Attributes --
    self.ProductName = product_name
    self.ProductPrice = product_price

# -- Properties --
# product_name
@property
def product_name(self):
    return self.__product_name

@product_name.setter
def product_name(self, name):
    if name.isnumeric() == False:
        self.__product_name = name
    else:
        raise Exception("Product can't be Numbers")
```

Figure 1

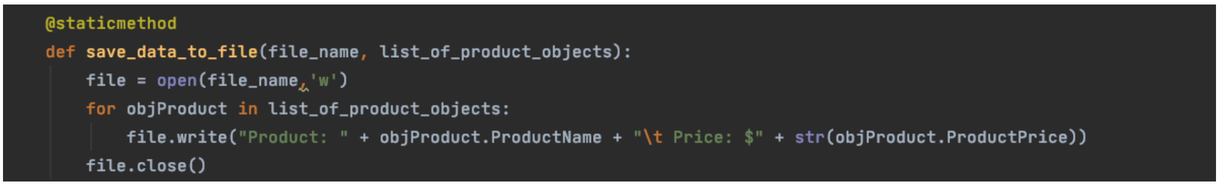
Methods

There were different methods that were used in this script, static, property and a setter. The statics method is more directive than other methods than others which helps process the data and stores it for more instance methods that would multiple copies of it (Figure 3). The property directive shows the getter function which allows use of the defined function to use it in the script in the future. The setter is then used after created a getter in order to directly use what happens within the setter (Figure 3). I was able to in the end use all the methods and call them in the body of the script in order to easily see where the script could go wrong when using the debugger. The methods help organize the script creating along with the pseudo guiding me on creating the script.



```
@property
def product_price(self):
    return self.__product_price

@product_price.setter
def product_price(self, value):
```

Figure 2



```
@staticmethod
def save_data_to_file(file_name, list_of_product_objects):
    file = open(file_name,'w')
    for objProduct in list_of_product_objects:
        file.write("Product: " + objProduct.ProductName + "\t Price: $" + str(objProduct.ProductPrice))
    file.close()
```

Figure 3

Summary

Through this module I was able to make a script that would be able to create a script that would be able to capture the product the user wants to keep track of as well as the products price. The script would be able to read the data from the text file, add new products with prices and save the data to the program the exit it. I was able to learn object-oriented programming techniques to use methods to be able to call the created classes in the scripts main body in order to have the script function properly.