Jake Murphy

November 29, 2019

IT FDN 100

Assignment08

**Classes and Their Components**

Contents

[Introduction 2](#_Toc24220569)

[Outline 2](#_Toc24220570)

[Operation 3](#_Toc24220571)

[Menu: 3](#_Toc24220572)

[Show current task list 3](#_Toc24220573)

[Add a new task 4](#_Toc24220574)

[Remove an existing task 4](#_Toc24220575)

[Save Data to File 4](#_Toc24220576)

[Exit Program 5](#_Toc24220577)

[Summary 5](#_Toc24220578)

[Script 6](#_Toc24220579)

# Introduction

This intent of this paper is to describe the procedures used in successfully performing this assignment.

# Outline

Given a “Starter-file” provided by the instructor, and building off the last assignments, we (the students) are to create a new project in PyCharm that incorporates classes and their components. The list manages a user inputted list of products and their prices.

# Operation

### Menu:

When the program launches, the user is prompted with a menu with four options.

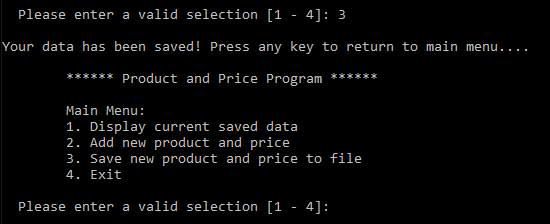
### Display current saved data

If the user selects option 1, they are presented with their list data.

### Add new product and price

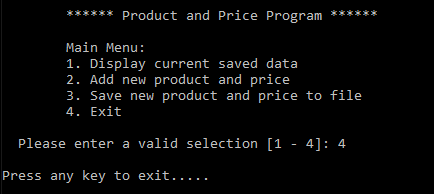
* The image to the right shows how a user would input option 2, to add new product and price.
* The image shows the user entered “Product C” followed by “14.99” for its price.
* The user is then prompted with a message saying, “Whatever product, whatever price” Was added to your data!
* The user is then routed back to the main menu.

### Save new product and price to file



When its time for the user to save their data to file they input “3” to call the save function.

As seen in the image to the right, a message stating “Your data has been saved! Press any key to return to main menu…” is displayed after the users save is completed.



### Exit

When the user is ready to exit the program, they choice option 4. The user is then prompted with “Press any key to exit….”

# Summary

This assignment was more difficult than anticipated. Classes are the blueprints that put together objects. Within classes you will find constructors, attributes, objects, setters, getters, and more. You really need to have a firm grasp on the basics to dive deep into classes. But you cant be afraid to dive in and since Python is an object oriented language, you better start learning classes sooner than later.

# Script

# ------------------------------------------------------------------------ #

# Title: Assignment 08

# Description: Working with classes

# ChangeLog (Who,When,What):

# RRoot,1.1.2030,Created started script

# RRoot,1.1.2030,Added pseudo-code to start assignment 8

# JMurphy,11.26.2019,Modified code to start assignment 8

# JMurphy,11.27.2019,Modified code to complete assignment 8

# JMurphy,11.28.2019,Modified code to complete assignment 8

# JMurphy,11.29.2019,Modified code and completed assignment 8

# ------------------------------------------------------------------------ #

# Data -------------------------------------------------------------------- #

strFileName = 'products.txt'

lstOfProductObjects = []

class Product:

"""Stores data about a product:

properties:

product\_name: (string) with the products's name

product\_price: (float) with the products's standard price

methods:

changelog: (When,Who,What)

RRoot,1.1.2030,Created Class

JMurphy,11.29.2019,Modified code to complete assignment 8

"""

def \_\_init\_\_(self, product\_name: str, product\_price: float):

"""Sets the user inputted name and price of product(s)"""

try:

self.p\_name = str(product\_name)

self.p\_price = float(product\_price)

except Exception as e:

raise Exception("Error setting initial values: \n" + str(e))

# Properties--------------------------------------------------------------- #

@property

def product\_name(self):

return str(self.p\_name)

@product\_name.setter

def product\_name(self, value: str):

if str(value).isnumeric():

self.p\_name = value

else:

print("Please enter a valid name for your product")

@property

def product\_price(self):

return float(self.p\_price)

@product\_price.setter

def product\_price(self, value: float):

if str(value).isnumeric():

self.p\_price = float(value)

else:

print("Prices must be numeric!")

# Methods ------------------------------------------------------------------#

def convert\_str(self):

return self.\_\_str\_\_()

def \_\_str\_\_(self):

return self.product\_name + ',' + str(self.product\_price)

# Data -------------------------------------------------------------------- #

# Processing ------------------------------------------------------------- #

class FileProcessor:

"""Processes data to and from a file and a list of product objects:

methods:

save\_data\_to\_file(file\_name, list\_of\_product\_objects):

read\_data\_from\_file(file\_name): -> (a list of product objects)

changelog: (When,Who,What)

RRoot,1.1.2030,Created Class

JMurphy,11.26.2019,Modified code to complete assignment 8

JMurphy,11.29.2019,Modified code to complete assignment 8

"""

@staticmethod

def read\_data\_from\_file(file\_name: str):

"""Shows the user the current items within products.txt

:param: file\_name: string

:return: a list of products in rows

"""

list\_of\_rows = []

file = open(strFileName, 'r')

for line in file:

data = line.split(",")

row = Product(data[0], data[1])

list\_of\_rows.append(row)

file.close()

return list\_of\_rows

# TODO: Add Code to process data to a file

@staticmethod

def save\_data\_to\_file(file\_name: str, list\_of\_product\_objects: list):

"""Write data to a file from a list

:param file\_name: (string) file name

:param list\_of\_product\_objects: (list) of products

"""

file = open(file\_name, "w")

for product in list\_of\_product\_objects:

file.write(product.\_\_str\_\_() + "\n")

file.close()

# ------- Processing ------------------------------------------------------------- #

# Presentation (Input/Output) -------------------------------------------- #

class inputOutput:

""" A class for performing Input and Output

methods:

mainMenu(): displays user menu

inputMenuChoice(): captures the users choice within mainMenu()

print\_current\_lst(list\_of\_rows): prints a list (list\_of\_rows)

add\_obj\_lst(): captures the users inputs before being transferred to a list

"""

@staticmethod

def mainMenu():

print('''

\*\*\*\*\*\* Product and Price Program \*\*\*\*\*\*

Main Menu:

1. Display current saved data

2. Add new product and price

3. Save new product and price to file

4. Exit

''')

@staticmethod

def inputMenuChoice():

""" Gets the menu choice from a user

:return: string

"""

choice = str(input(" Please enter a valid selection [1 - 4]: ")).strip()

print()

return choice

@staticmethod

def print\_current\_lst(list\_of\_rows: list):

"""Prints the current list

:param list\_of\_rows: (list) of rows to be displayed

:return: list of rows

"""

print("Current product and prices in list:")

for row in list\_of\_rows:

print(row.product\_name + "(" + str(row.product\_price) + ")")

@staticmethod

def add\_obj\_lst():

"""adds objects to the current list

:param: \_\_name: placeholder to save in name of product

:param: \_\_price: same as above but with price

:return: product(inputted user data)

"""

\_\_name = str(input('Enter product name: ')).strip()

\_\_price = float(input('Enter product price (ex. 9.99): ').strip())

print()

newProd = Product(product\_name=\_\_name, product\_price=\_\_price)

print(newProd, " " "Was added to your data!")

return newProd

# Presentation (Input/Output) -------------------------------------------- #

# Main Body of Script ---------------------------------------------------- #

try:

lstOfProductObjects = FileProcessor.read\_data\_from\_file(strFileName)

while True:

inputOutput.mainMenu()

strChoice = inputOutput.inputMenuChoice()

if strChoice.strip() == '1':

inputOutput.print\_current\_lst(lstOfProductObjects)

continue

elif strChoice.strip() == '2':

lstOfProductObjects.append(inputOutput.add\_obj\_lst())

continue

elif strChoice.strip() == '3':

FileProcessor.save\_data\_to\_file(strFileName, lstOfProductObjects)

input("Your data has been saved! Press any key to return to main menu....")

continue

elif strChoice.strip() == '4':

input("Press any key to exit.....")

break

except Exception as e:

print("There was an error! Check file permissions.")

print(e, e.\_\_doc\_\_, type(e), sep='\n')

# Main Body of Script ---------------------------------------------------- #