Vivian Ta

IT FDN 100 A

Assignment 8 Notes

Nov. 25, 2018

I decided to make two classes, one to handle the product itself and another as an organizational class to hold the methods that are interacting with the Product file itself.

My `Product` class includes 3 fields that I’m initializing: `ID`, `Name`, and `Price` and I’ve decided to omit declaring the field names since declaring them in the constructor will create them for me. Each field has a getter/setter property and I’m overriding the default \_\_str\_\_ method with something custom that includes all 3 fields plus some formatting.

I end up using the `Product` class in the `WriteProductUserInput` def and substitute strUserInput for my `Product` object and write that into the file instead.

*# Processing  
def* WriteProductUserInput(*File*):  
 *try*:  
 *while*(*True*):  
 *try*:  
 choice = input("Continue or Exit? ")  
 *if* choice.lower() == "exit": *break* id = int(input("Enter the ID: "))  
 name = input("Enter the name of the product: ")  
 price = float(input("Enter the price of the product: "))  
 product = Product(id, name, price).ToString()  
 *File*.write(product + "\n")  
 *except* ValueError *as* e:  
 print("Unexpected value type, exiting...")  
 *break  
 except* Exception *as* e:  
 print("Error: " + str(e))

Since I added an organizational class for the write and read defs, I made sure to change the method calls to refer to the class dot notation so they’re called properly.

*try*:  
 AccessProductData.DisplayMenu()  
 objFile = open("Products.txt", "r+")  
 AccessProductData.ReadAllFileData(objFile, "Here is the current data:")  
 AccessProductData.WriteProductUserInput(objFile)  
 AccessProductData.ReadAllFileData(objFile, "Here is the data that has been saved so far:")  
*except* FileNotFoundError *as* e:  
 print("Error: " + str(e) + "\n Please check the file name")  
*except* Exception *as* e:  
 print("Error: " + str(e))  
*finally*:  
 *if*(objFile != *None*):objFile.close()