Guang-Sin Lu

Mar 19, 2022

Foundations of Programming, Python

Assignment 08

CD Inventory Python Script-4

# Introduction

Python is a pure Object-Oriented Language. In this week, we learned the definition and how to create Classes (blueprint for an object), Fileds(the data stores of a class), Constructors (the dedicated method is invoked when creating an object), Destrictors(), Attributes(internal fields or variables that hold data), Properties, and Methods(like functions). The purpose of this assignment was rewrite the code of CDInventory.py by using Object Oriented Programming (OOP).

# Modify the CD Inventory Program

1. Modify log information of **CDInventory.py** (in Assignment 06) and save it to a new script, **CDInventory.py**, in folder **Assignment08.**

Text

Description automatically generated

Figure 1 – Information and log of CDInventory.py

1. To start, I used the pseudo code in start.py and combined/modified code from the previous assignments. Here, I choose the txt instead binary store form.
2. Declare the data variables. (after create the main body of script, move all data into the main () function.)

Graphical user interface, text

Description automatically generated with medium confidence

Figure 2 – Variables of CDInventory.py

1. Defining the Class, Constructors, methods, attributes, Properties in CD class. Here, I use the \_\_int\_\_ and \_\_str\_\_ methods, setter properties and self-parameter to hold the user data (ID, title and artist) to be initiated for the CD Inventory.

Text

Description automatically generated

Figure 3 – CD class of CDInventory.py

1. Next, I worked on the DataProcessor Class. Here I need to define the code of @Staticmethod of add and delete data from inventory. The adding part is straight forward and simple. But the delete part used the Structured Error Handling(ValueError) to check the input of ID is integer or not. This part was modified by previous code.

Text

Description automatically generated

Figure 4 – Data Processor Class of CDInventory.py

1. Then, I worked on the FileIO Class. I need to define the function of load and save inventory. The same, I modified the code from previous two assignment. Moreover, I read the file as attributes instead from the dictionary or binary file. Also, I add the Structured Error Handling to check the “FileNotFoundError” of load\_inventory function.

Text

Description automatically generated

Figure 5 – FileIO Class of CDInventory.py

1. Next is the presentation of the Code. I need print\_menu function to show the main menu, menu\_choice function to check the typing, show\_inventory function to display, get\_UserInput function to get the new data. The same, I modified the previous code. I add the structed Error heading to interreact with user. Another modification is read the list of attributes instead of the dictionary.

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Figure 6 –IO class of CDInventory.py

1. The final part, I followed the step of provided pusdo-code to finish the main function.

Text

Description automatically generated

Text

Description automatically generated

Figure 7 –MainFunction of CDInventory.py

1. Save the script.

# Spyder Test

1. Ran the code, test no CDInventory.txt file (same result as test option I)

Text

Description automatically generated

Figure 8-1 – Test result 1 (Spyder) of CDInventory.py

1. Ran the code, test invalid option.

Text

Description automatically generated

Figure 8-2 – Test result 2 (Spyder) of CDInventory.py

1. Option A, add new data , and test invalid ID.

Text

Description automatically generated

Text

Description automatically generated

Figure 8-3 – Test result 3 (Spyder) of CDInventory.py

1. Option D, delete temporary data, and test invalid ID.

Text

Description automatically generated

Text

Description automatically generated

Figure 8-4 – Test result 4 (Spyder) of CDInventory.py

1. Option S, save and not to save the text file.

Text

Description automatically generated

Text

Description automatically generated

Figure 8-5 – Test result 5 (Spyder) of CDInventory.py

1. Option i, display the current inventory

Graphical user interface, text, application

Description automatically generated

Figure 8-6 – Test result 6 (Spyder) of CDInventory.py

1. Option l, load and not load the exist text file.

Text

Description automatically generated

Figure 8-7 – Test result 7 (Spyder) of CDInventory.py

1. Option x, exist the program.

# Terminal Test

1. Ran the code, test no CDInventory.txt file (same result as test option I)

A picture containing text

Description automatically generated

Figure 9-1 – Test result 1 (Spyder) of CDInventory.py

1. Ran the code, test invalid option.

Text

Description automatically generated with medium confidence

Figure 9-2 – Test result 2 (Spyder) of CDInventory.py

1. Option A, add new data , and test invalid ID.

Text, letter

Description automatically generated

Figure 9-3 – Test result 3 (Spyder) of CDInventory.py

1. Option D, delete temporary data, and test invalid ID.

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

Figure 9-4 – Test result 4 (Spyder) of CDInventory.py

1. Option S, save and not to save the text file.

Text, letter

Description automatically generated

Figure 9-5 – Test result 5 (Spyder) of CDInventory.py

1. Option i, display the current inventory

Text

Description automatically generated

Figure 9-6 – Test result 6 (Spyder) of CDInventory.py

1. Option l, load and not load the exist text file.

Text

Description automatically generated

Figure 9-7 – Test result 7 (Spyder) of CDInventory.py

1. Option x, exist the program.

# GitHub

I created a new repository “Assignment\_08”. Upload my knowledge document (Assignment\_08.docx), python script (CDInventory.py).

# My GitHub page link is <https://github.com/guangsil/Assignment_08>

Figure 10 –GitHub Page

# Summary

In Assignment 08, I use the different blueprint to re-write/ modify the previous assignment. This script has the same result of previous assignment. The previous code was straight forward. However, this OOP could expend the code clearer, simpler, and more organized.