Dana Defelici

2021-Nov-20

IT FDN 110 B – Foundations of Programming (Python)

Assignment 06

Modifying another CD Inventory Script

# Introduction

This script works with a CD inventory stored in CDInventory.txt and asks the user if they would like to load inventory from file, add a CD, display the current inventory, delete a CD from inventory, save the inventory to file or exit. In this assignment I continued to refactor the script to separate out data processing and file processing into separate classes (Separations of Concern).

# Updating the code from TODO lists

For the most part the code was already implemented. In this assignment I moved the code to processing classes, I added static methods and docstring. I also added parameters and renamed variables.

# File Processing Changes

I moved the write file logic to the processing section and added it to the FileProcessor class. I updated the table name variable and the file name variable.

# Data Processing Changes

I moved both adding and deleting to the processing section and added them to the DataProcessor class.

# I/O Changes

I moved the get CD input to the processing section and added it to the IO class.

# Uploading to GitHub

Using the GitHub account I created last week I then created a public Assignment\_06 repository. I uploaded this script, the knowledge document and made sure there was a README.md file. I’ve included a link to the repository [here](https://github.com/pixel-pusher314/Assignment_06).

# Summary

In this assignment we start with someone else’s code then we updated it by moving logic into the proper Separations of Concern.

# Appendix

## Listing CDInventory.py

1. *#------------------------------------------#*
2. *# Title: CDInventory.py*
3. *# Desc: Working with classes and functions.*
4. *# Change Log: (Who, When, What)*
5. ***# DBiesinger, 2030-Jan-01, Created File***
6. *# DDefelici, 2021-Nov-20, moved code to processing classes, add new static methods and docstring*
7. *#------------------------------------------#*
9. *# -- DATA -- #*
10. **strChoice = '' *# User input***
11. lstTbl = [] *# list of lists to hold data*
12. dicRow = {} *# list of data row*
13. strFileName = 'CDInventory.txt' *# data storage file*
14. objFile = None *# file object*

17. *# -- PROCESSING -- #*
18. **class** DataProcessor:
19. """Processing the data to and from runtime list"""
21. @staticmethod
22. **def** add\_cd(cd, table):
23. """Function to add CD to runtime list
25. **Stores the data from CD into runtime list**
27. Args:
28. cd (tuple): (ID, Title, Artist)
29. table (list of dict): 2D data structure (list of dicts) that holds the data during runtime
31. Returns:
32. None.
33. """
34. intID = int(cd[0])
35. **dicRow = {'ID': intID, 'Title': cd[1], 'Artist': cd[2]}**
36. table.append(dicRow)
38. @staticmethod
39. **def** del\_cd(id, table):
40. **"""Function to remove CD to runtime list**
42. Removes the CD data from the runtime list
44. Args:
45. **ID (int): ID of CD to remove**
46. table (list of dict): 2D data structure (list of dicts) that holds the data during runtime
48. Returns:
49. boolean: indicates if sucessfullly removed.
50. **"""**
51. intRowNr = -1
52. blnCDRemoved = False
53. **for** row **in** table:
54. intRowNr += 1
55. **if row['ID'] == id:**
56. **del** table[intRowNr]
57. blnCDRemoved = True
58. **break**
59. **return** blnCDRemoved
61. **class** FileProcessor:
62. """Processing the data to and from text file"""
64. @staticmethod
65. **def read\_file(file\_name, table):**
66. """Function to manage data ingestion from file to a list of dictionaries
68. Reads the data from file identified by file\_name into a 2D table
69. (list of dicts) table one line in the file represents one dictionary row in table.
71. Args:
72. file\_name (string): name of file used to read the data from
73. table (list of dict): 2D data structure (list of dicts) that holds the data during runtime
75. **Returns:**
76. None.
77. """
78. table.clear() *# this clears existing data and allows to load data from file*
79. objFile = open(file\_name, 'r')
80. **for line in objFile:**
81. data = line.strip().split(',')
82. dicRow = {'ID': int(data[0]), 'Title': data[1], 'Artist': data[2]}
83. table.append(dicRow)
84. objFile.close()
86. @staticmethod
87. **def** write\_file(file\_name, table):
88. """Function to manage data ingestion to file of a list of dictionaries
90. **Writes the data to the file identified by file\_name from a 2D table**
91. (list of dicts) table one line in the file represents one dictionary row in table.
93. Args:
94. file\_name (string): name of file used to write the data to
95. **table (list of dict): 2D data structure (list of dicts) that holds the data during runtime**
97. Returns:
98. None.
99. """
100. **objFile = open(file\_name, 'w')**
101. **for** row **in** table:
102. lstValues = list(row.values())
103. lstValues[0] = str(lstValues[0])
104. objFile.write(','.join(lstValues) + '**\n**')
105. **objFile.close()**

108. *# -- PRESENTATION (Input/Output) -- #*
110. **class IO:**
111. """Handling Input / Output"""
113. @staticmethod
114. **def** print\_menu():
115. **"""Displays a menu of choices to the user**
117. Args:
118. None.
120. **Returns:**
121. None.
122. """
124. **print**('Menu**\n\n**[l] load Inventory from file**\n**[a] Add CD**\n**[i] Display Current Inventory')
125. **print('[d] delete CD from Inventory\n[s] Save Inventory to file\n[x] exit\n')**
127. @staticmethod
128. **def** menu\_choice():
129. """Gets user input for menu selection
131. Args:
132. None.
134. Returns:
135. **choice (string): a lower case sting of the users input out of the choices l, a, i, d, s or x**
137. """
138. choice = ' '
139. **while** choice **not** **in** ['l', 'a', 'i', 'd', 's', 'x']:
140. **choice = input('Which operation would you like to perform? [l, a, i, d, s or x]: ').lower().strip()**
141. **print**() *# Add extra space for layout*
142. **return** choice
144. @staticmethod
145. **def show\_inventory(table):**
146. """Displays current inventory table

149. Args:
150. **table (list of dict): 2D data structure (list of dicts) that holds the data during runtime.**
152. Returns:
153. None.
155. **"""**
156. **print**('======= The Current Inventory: =======')
157. **print**('ID**\t**CD Title (by: Artist)**\n**')
158. **for** row **in** table:
159. **print**('{}**\t**{} (by:{})'.format(\*row.values()))
160. **print('======================================')**
162. @staticmethod
163. **def** get\_CD():
164. """Get data for CD from input
166. Args:
167. None.
169. Returns:
170. **cd (tuple): (ID, Title, Artist)**
172. """
173. strID = input('Enter ID: ').strip()
174. strTitle = input('What is the CD**\'**s title? ').strip()
175. **strArtist = input('What is the Artist\'s name? ').strip()**
176. **return** (strID, strTitle, strArtist)
178. *# 1. When program starts, read in the currently saved Inventory*
179. FileProcessor.read\_file(strFileName, lstTbl)
181. *# 2. start main loop*
182. **while** True:
183. *# 2.1 Display Menu to user and get choice*
184. IO.print\_menu()
185. **strChoice = IO.menu\_choice()**
187. *# 3. Process menu selection*
188. *# 3.1 process exit first*
189. **if** strChoice == 'x':
190. **break**
191. *# 3.2 process load inventory*
192. **if** strChoice == 'l':
193. **print**('WARNING: If you continue, all unsaved data will be lost and the Inventory re-loaded from file.')
194. strYesNo = input('type **\'**yes**\'** to continue and reload from file. otherwise reload will be canceled: ')
195. **if strYesNo.lower() == 'yes':**
196. **print**('reloading...')
197. FileProcessor.read\_file(strFileName, lstTbl)
198. IO.show\_inventory(lstTbl)
199. **else**:
200. **input('canceling... Inventory data NOT reloaded. Press [ENTER] to continue to the menu.')**
201. IO.show\_inventory(lstTbl)
202. **continue** *# start loop back at top.*
203. *# 3.3 process add a CD*
204. **elif** strChoice == 'a':
205. ***# 3.3.1 Ask user for new ID, CD Title and Artist***
206. cd = IO.get\_CD()
208. *# 3.3.2 Add item to the table*
209. DataProcessor.add\_cd(cd, lstTbl)
210. **IO.show\_inventory(lstTbl)**
211. **continue** *# start loop back at top.*
212. *# 3.4 process display current inventory*
213. **elif** strChoice == 'i':
214. IO.show\_inventory(lstTbl)
215. **continue *# start loop back at top.***
216. *# 3.5 process delete a CD*
217. **elif** strChoice == 'd':
218. *# 3.5.1 get Userinput for which CD to delete*
219. *# 3.5.1.1 display Inventory to user*
220. **IO.show\_inventory(lstTbl)**
221. *# 3.5.1.2 ask user which ID to remove*
222. intIDDel = int(input('Which ID would you like to delete? ').strip())
223. *# 3.5.2 search thru table and delete CD*
224. blnCDRemoved = DataProcessor.del\_cd(intIDDel, lstTbl)
225. **if blnCDRemoved:**
226. **print**('The CD was removed')
227. **else**:
228. **print**('Could not find this CD!')
229. IO.show\_inventory(lstTbl)
230. **continue *# start loop back at top.***
231. *# 3.6 process save inventory to file*
232. **elif** strChoice == 's':
233. *# 3.6.1 Display current inventory and ask user for confirmation to save*
234. IO.show\_inventory(lstTbl)
235. **strYesNo = input('Save this inventory to file? [y/n] ').strip().lower()**
236. *# 3.6.2 Process choice*
237. **if** strYesNo == 'y':
238. *# 3.6.2.1 save data*
239. FileProcessor.write\_file(strFileName, lstTbl)
240. **else:**
241. input('The inventory was NOT saved to file. Press [ENTER] to return to the menu.')
242. **continue** *# start loop back at top.*
243. *# 3.7 catch-all should not be possible, as user choice gets vetted in IO, but to be save:*
244. **else**:
245. **print('General Error')**

Listing - function demo

Using [highlight.hohli’s](https://highlight.hohli.com/index.php) (external reference) webpage[[1]](#endnote-1) to highlight code.

Text

Description automatically generated

Figure - Code working in Spyder 1 of 3

Text

Description automatically generated

Figure - Code working in Spyder 2 of 3

Text

Description automatically generated

Figure - Code working in Spyder 3 of 3

Text

Description automatically generated

Figure - Code working in terminal 1 of 2

Text

Description automatically generated

Figure - Code working in terminal 2 of 2

1. Retrieved 2021-Nov-20 [↑](#endnote-ref-1)