Adam Hockemeyer

2020-Mar-16

Introduction to Programming (Python)

Assignment08

**Object Oriented Programming**

**Introduction**

During this assignment, I took our CDInventory script and changed it so that the data would be stored in custom class objects. Each of these CD objects would be stored in a list, and that list would serve as the inventory system. I created a custom class and methods for this function.

**Creating the CD class**

To use a CD object, I had to create the class for it, and any associated properties and methods. The CD class I created stores three main pieces of data about a CD: the ID number in our inventory system, the name of the CD, and the artist name. These are the three properties of each CD we instantiate. The methods I wanted to tie to the CD class both revolve around showing the data held within the object. The first, print\_file(), returns a formatted string to be saved to a CSV file. The second, print\_legible(), returns a forrmatted string that looks good when printed. This second method will serve to be the primary way that any user would see the data stored within the objects.



*Figure 1 - the script being run in Spyder IDE, showing the add CD functionality.*

**Organzing the Code**

In order to work with this different data type, I started from only pseudocode with this script. The primary task would be figuring out which functions would go in which classes, and making sure each of those functions worked with the newly created CD class objects.

I decided that along with the CD class, my script would use three other classes. Firstly, the FileIO class would hold the two functions to load in data from a file to memory, and the reverse of saving the data in memory to file. Both of these functions work very similarly to the same code in the past, except instead of reading and writing to lists, these two functions either instantiate objects with the parameters to give them data or take objects and turn them into data lines to be saved.

My second class would be DataProcessor, to handle the manipulation of data within memory. The two functions in this class would be add\_CD, to instantiate a new CD object given the information for that CD, and show\_inventory, to print the table currently being housed in memory. This second function makes use of the CD object's method to print its data in a legible manner, taking advantage of the fact that our objects are able to work with methods that they work with.

My third and final class would be IO, to house the functions to show a menu and collect user inputs. All three of these functions work the same way as they have in previous weeks, although I did put an error handling block in input\_CD in order to make sure the ID number is input as an int.



*Figure 2 - the script being run in Anaconda Prompt, showing the save to file function.*

**Testing the Changes**

After setting these changes in the code, I ran the script in both Spyder and in Anaconda Prompt, testing out each menu option and taking screenshots.



*Figure 3 - the script being run in Anaconda Prompt, showing the add and show inventory functionality.*

**Summary**

I was able to update our inventory script to use class objects, methods, and properties to aid in the functionality of our inventory system.

GitHub link: <https://github.com/heliotropite/Assignment_08>