Mariia Kot
December 4, 2022
Foundations of Programming, Python
Assignment 08

CD Inventory Program with classes

Introduction

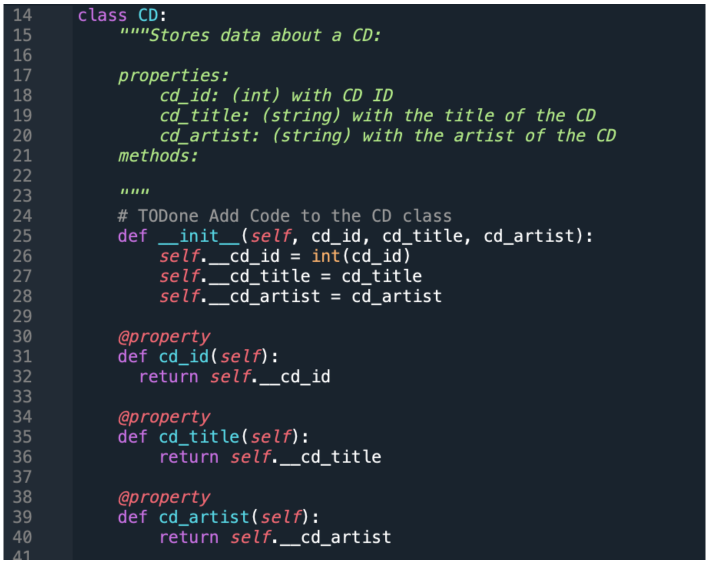
In this module, I learned about Object Oriented Programming. The main idea of it is that everything is an object. An object has attributes and methods and an object itself is an instance of a class. Each object of the class has the same functionality as the class, but can be customized by the attributes. This module is still a bit confusing for me, but I tried my best to understand at least the main ideas.

Applying this knowledge to assignment 8, I have created the following program.

Step-by-Step

In this document, I will go over the changes that I have made since the last assignment07/06.

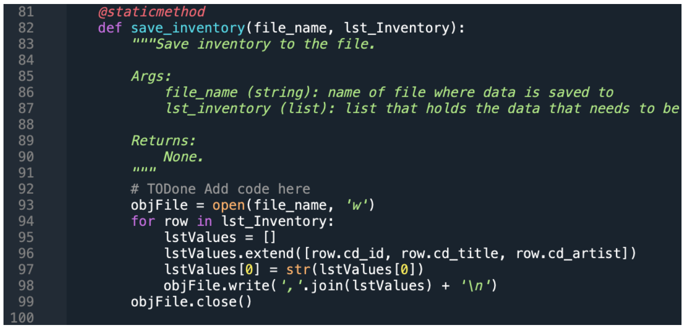
Firstly, I created a constructor using __init__() method and I passed 3 arguments to it. As we do not want the user to have access to these attributes, I made them private by adding a double underscore before the name. Next, we need to check cd_id to be an integer, for that, I tried to implement a getter/setter method, but I could not figure it out, so I added an error handling method later in the code from Assignment 07 instead.



```python
class CD:
    """Stores data about a CD:

    properties:
        cd_id: (int) with CD ID
        cd_title: (string) with the title of the CD
        cd_artist: (string) with the artist of the CD
    methods:

    """
    # TODone Add Code to the CD class
    def __init__(self, cd_id, cd_title, cd_artist):
        self.__cd_id = int(cd_id)
        self.__cd_title = cd_title
        self.__cd_artist = cd_artist

    @property
    def cd_id(self):
      return self.__cd_id

    @property
    def cd_title(self):
        return self.__cd_title

    @property
    def cd_artist(self):
        return self.__cd_artist
```

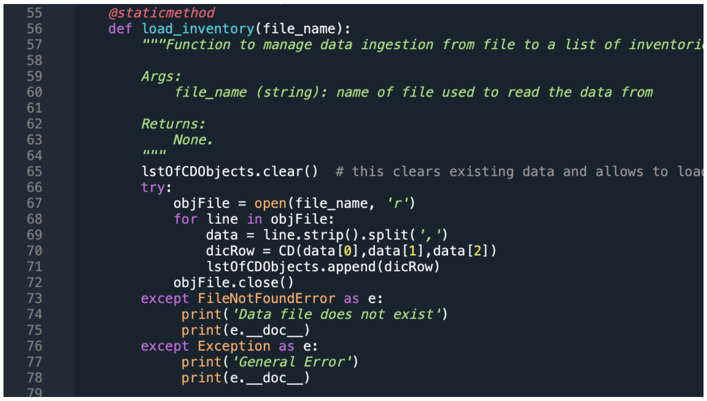
Following the pseudocode provided, I created a code for the class FileIO. I used @staticmethod to process the data and create load_inventory and save_inventory. I want these methods to be called on the class level and not on the instance level, thus I used @staticmethod. As it does not run on the object level, no “self” attribute is required.

For save_inventory I used my code from assignment06, but as I could not use dictionary values, I created an empty list lstValues where I added my instances cd_id,cd_title, and cd_artist and then I write this list to the file.



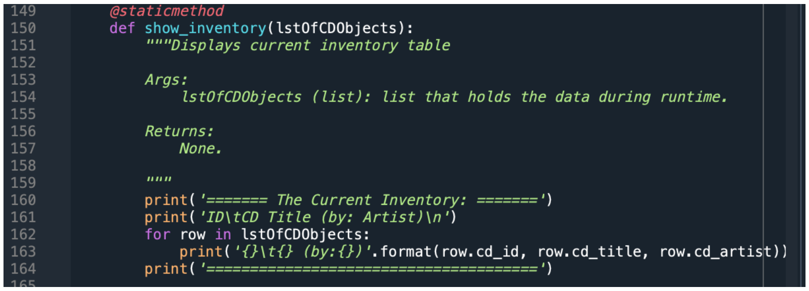
```python
    @staticmethod
    def save_inventory(file_name, lst_Inventory):
        """Save inventory to the file.

        Args:
            file_name (string): name of file where data is saved to
            lst_inventory (list): list that holds the data that needs to be

        Returns:
            None.
        """
        # TODone Add code here
        objFile = open(file_name, 'w')
        for row in lst_Inventory:
            lstValues = []
            lstValues.extend([row.cd_id, row.cd_title, row.cd_artist])
            lstValues[0] = str(lstValues[0])
            objFile.write(','.join(lstValues) + '\n')
        objFile.close()
```

For load_inventory I added an error handling code from Wednesday's class and modifies the dicRow variable to store the instances of the CD class. Then I append dicRow to the lstOFCDObjects.



```python
    @staticmethod
    def load_inventory(file_name):
        """Function to manage data ingestion from file to a list of inventori

        Args:
            file_name (string): name of file used to read the data from

        Returns:
            None.
        """
        lstOfCDObjects.clear()  # this clears existing data and allows to loa
        try:
            objFile = open(file_name, 'r')
            for line in objFile:
                data = line.strip().split(',')
                dicRow = CD(data[0],data[1],data[2])
                lstOfCDObjects.append(dicRow)
            objFile.close()
        except FileNotFoundError as e:
             print('Data file does not exist')
             print(e.__doc__)
        except Exception as e:
             print('General Error')
             print(e.__doc__)
```

For class IO, I used @staticmethod again for the same reasons. I mainly copied and pasted the code from previous assignments. The only modification that I made is that I used CD instances for show_inventory instead of the dictionary row.value().

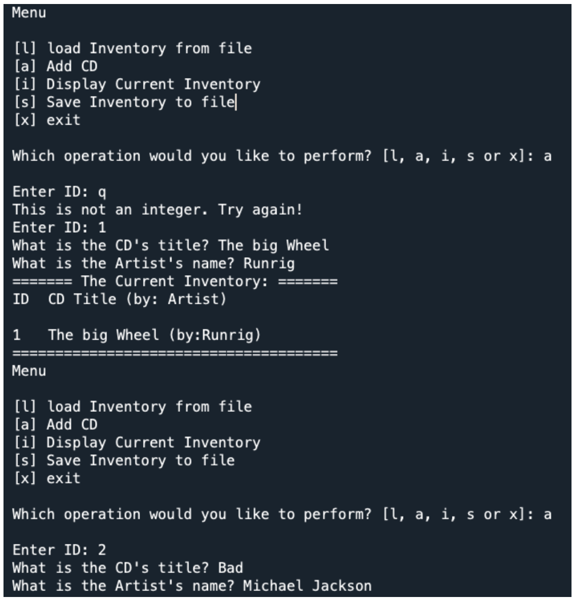


```
    @staticmethod
    def show_inventory(lstOfCDObjects):
        """Displays current inventory table

        Args:
            lstOfCDObjects (list): list that holds the data during runtime.

        Returns:
            None.

        """
        print('======= The Current Inventory: =======')
        print('ID\tCD Title (by: Artist)\n')
        for row in lstOfCDObjects:
            print('{}\t{} (by:{})'.format(row.cd_id, row.cd_title, row.cd_artist))
        print('======================================')
```

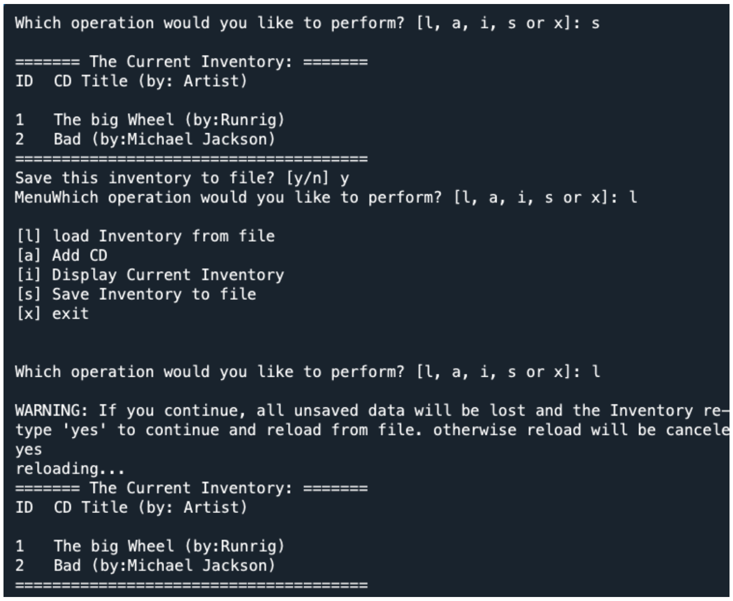
For the main menu, I also copied the code from the previous assignment and modified the names of the classes and functions.

Results

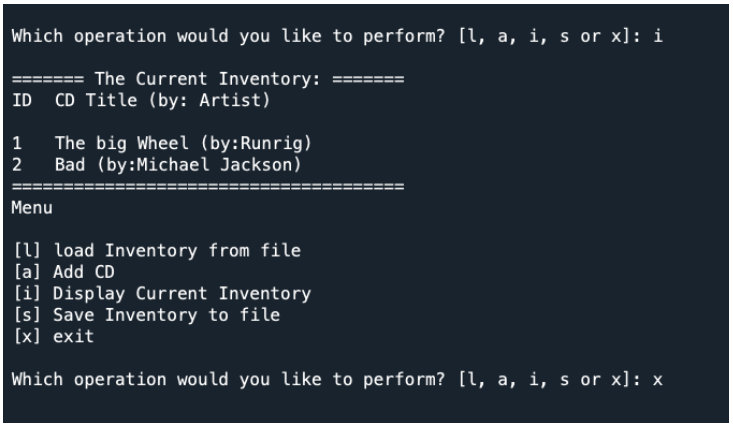
Here are some results from running this program in Spyder and Terminal.



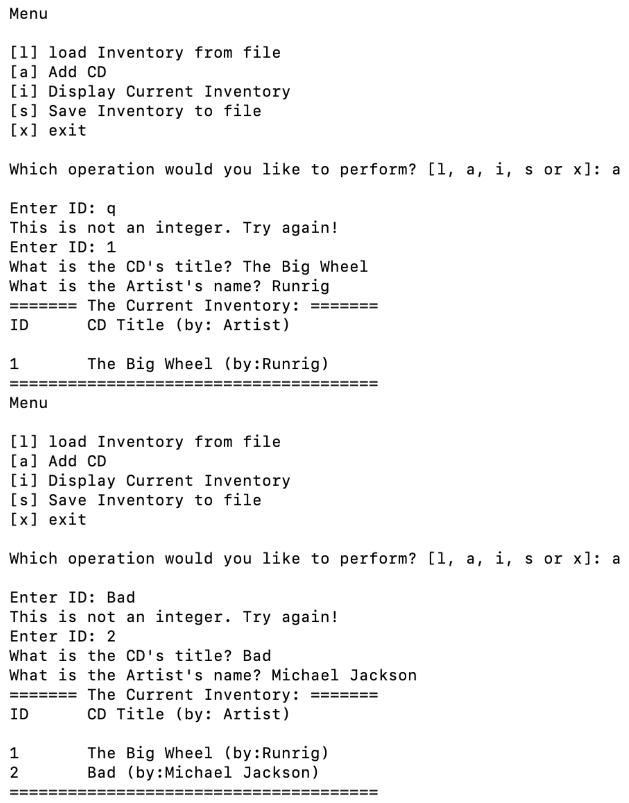
```
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, s or x]: a

Enter ID: q
This is not an integer. Try again!
Enter ID: 1
What is the CD's title? The big Wheel
What is the Artist's name? Runrig
======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   The big Wheel (by:Runrig)
======================================
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, s or x]: a

Enter ID: 2
What is the CD's title? Bad
What is the Artist's name? Michael Jackson
```



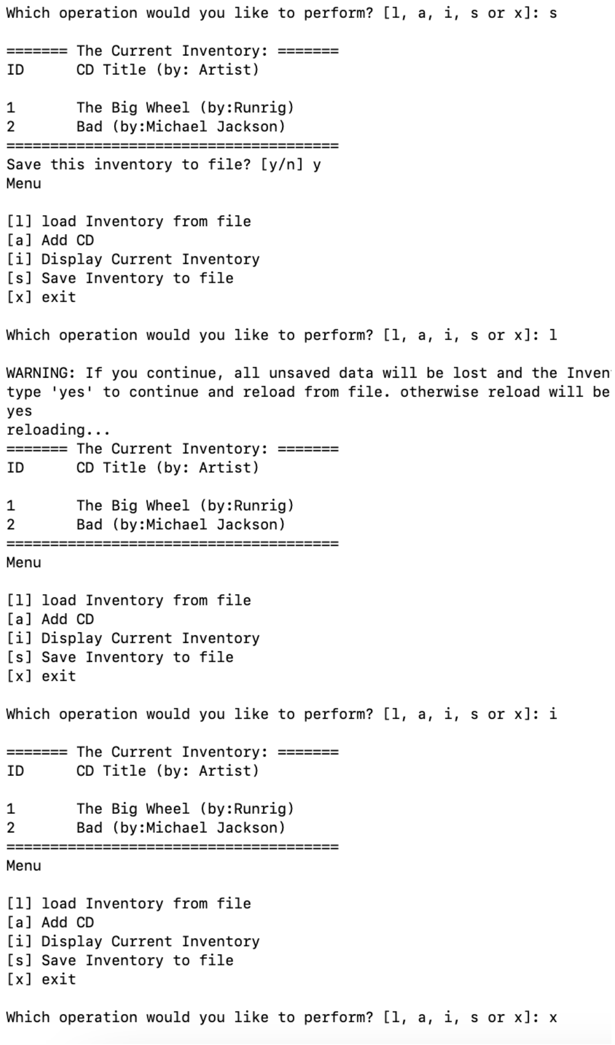
```
Which operation would you like to perform? [l, a, i, s or x]: s

======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   The big Wheel (by:Runrig)
2   Bad (by:Michael Jackson)
======================================
Save this inventory to file? [y/n] y
MenuWhich operation would you like to perform? [l, a, i, s or x]: l

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to file
[x] exit


Which operation would you like to perform? [l, a, i, s or x]: l

WARNING: If you continue, all unsaved data will be lost and the Inventory re-
type 'yes' to continue and reload from file. otherwise reload will be cancele
yes
reloading...
======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   The big Wheel (by:Runrig)
2   Bad (by:Michael Jackson)
======================================
```



```
Which operation would you like to perform? [l, a, i, s or x]: i

======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   The big Wheel (by:Runrig)
2   Bad (by:Michael Jackson)
======================================
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, s or x]: x
```



```
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, s or x]: a

Enter ID: q
This is not an integer. Try again!
Enter ID: 1
What is the CD's title? The Big Wheel
What is the Artist's name? Runrig
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       The Big Wheel (by:Runrig)
======================================
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, s or x]: a

Enter ID: Bad
This is not an integer. Try again!
Enter ID: 2
What is the CD's title? Bad
What is the Artist's name? Michael Jackson
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       The Big Wheel (by:Runrig)
2       Bad (by:Michael Jackson)
======================================
```



```
Which operation would you like to perform? [l, a, i, s or x]: s

======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       The Big Wheel (by:Runrig)
2       Bad (by:Michael Jackson)
======================================
Save this inventory to file? [y/n] y
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, s or x]: l

WARNING: If you continue, all unsaved data will be lost and the Inven
type 'yes' to continue and reload from file. otherwise reload will be
yes
reloading...
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       The Big Wheel (by:Runrig)
2       Bad (by:Michael Jackson)
======================================
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, s or x]: i

======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       The Big Wheel (by:Runrig)
2       Bad (by:Michael Jackson)
======================================
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, s or x]: x
```

Summary

In this assignment, I worked with classes, it was very challenging for me to understand how everything is connected and how to call for the object from the class in the program. I still feel a bit lost and cannot quite understand how to use getter/setter for error handling and how to use __str__ to load instances to the program.

My GitHub: https://github.com/mustbekot/Assignment08