

Naga Anusha.Perali

March 14, 2022

Python Script Assignment 9

GitHub Repository Link: <https://github.com/NagaAusha/ITFnd100-mod-9>

Python Modules and Importing modules

Introduction

In this assignment, I'm going to explain how I create a Python Program where user can choose from the menu of options to perform the program by using Import

Data Classes

Using the listing (6 and 9) from the class notes I created Person class and Employee classes code and to test the listings I further added the code from listings (8 and 10) check the image below

```

# Description: Working with Modules

# ChangeLog (Who,When,What):
# RRoot,1.1.2030,Created started script
# RRoot,1.1.2030,Added pseudo-code to start assignment 9
# Naga Anusha,14.3.2022,Modified code to complete assignment 9
# ----- #
from builtins import Exception, print

if __name__ == "__main__":
    from DataClasses import Employee as Emp
    from ProcessingClasses import FileProcessor as Fp
    from IOClasses import EmployeeIO as Eio
    else:
        raise Exception("This file was not created to be imported")

# Main Body of Script ----- #
# -----Data----- #
lstEmployeeTable = [] # A list of Employee objects
lstFileData = [] # A list of string objects in a list
# TODO: Add Data Code to the Main body
# Load data from file into a list of employee objects when script starts
lstFileData = Fp.read_data_from_file("EmployeeData.txt")
for row in lstFileData:
    lstEmployeeTable.append(Emp(row[0], row[1], row[2].strip()))

# Show user a menu of options
while True:
    Eio.print_menu_items()
    # Get user's menu option choice
    strOption = Eio.input_menu_options()
    if strOption == "1":
        # Show user current data in the list of employee objects
        Eio.print_current_list_items(lstEmployeeTable)
        continue
    elif strOption == "2":
        # Let user add data to the list of employee objects
        lstEmployeeTable.append(Eio.input_employee_data())
        continue
    elif strOption == "3":
        # let user save current data to file
        Fp.save_data_to_file("EmployeeData.txt", lstEmployeeTable)
        print("Data was saved")
        continue
    elif strOption == "4":
        print("Program finished good bye")
        break # user exit program

```

Pic 1: Code for DataClasses

Processing Classes

For the Processing class, code from the listings 7 and 10 is added to test and create the FileProcessorclass to Test Harness script. Check the image below

```

from builtins import Exception, str, list, open, print, staticmethod, type

if __name__ == "__main__":
    raise Exception("This file is not meant to ran by itself")

class FileProcessor:
    """Processes data to and from a file and a list of objects:

    methods:
        save_data_to_file(file_name,list_of_objects):
            read_data_from_file(file_name): -> (a list of objects)

    changelog: (When,Who,What)
        RRoot,1.1.2030,Created Class
    """

    @staticmethod
    def save_data_to_file(file_name: str, list_of_objects: list):
        """ Write data to a file from a list of object rows

        :param file_name: (string) with name of file
        :param list_of_objects: (list) of objects data saved to file
        :return: (bool) with status of success status
        """
        success_status = False
        try:
            file = open(file_name, "w")
            for row in list_of_objects:
                file.write(row.__str__() + "\n")
            file.close()
            success_status = True
        except Exception as e:
            print("There was a general error!")
            print(e, e.__doc__, type(e), sep='\n')
        return success_status

    @staticmethod
    def read_data_from_file(file_name: str):
        """ Reads data from a file into a list of object rows

        :param file_name: (string) with name of file
        :return: (list) of object rows
        """
        list_of_rows = []
        try:
            file = open(file_name, "r")
            for line in file:
                row = line.split(",")
                list_of_rows.append(row)
            file.close()
        except Exception as e:
            print("There was a general error!")
            print(e, e.__doc__, type(e), sep='\n')
        return list_of_rows

```

Pic 2: Code for ProcessingClasses

Input/output Classes

To create the IOClasses script, I used listing 11 this listing contained the code to create the Employee IO class and to test this and all the previous classes, I use the code from listing 12 from in my Test Harnes script. Check the image below

```

class EmployeeIO:
    """ A class for performing Employee Input and Output

    methods:
        print_menu_items():
        print_current_list_items(list_of_rows):
        input_employee_data():

    changelog: (When,Who,What)
        Naga Anusha,14.3.2022,Modified the script
    """
    @staticmethod
    def print_menu_items():
        """ Print a menu of choices to the user """
        print('''
        Menu of Options
        1) Show current employee data
        2) Add new employee data.
        3) Save employee data to File
        4) Exit program
        ''')
        print() # Add an extra line for looks

    @staticmethod
    def input_menu_options():
        """ Gets the menu choice from a user

        :return: string
        """
        choice = str(input("Which option would you like to perform? [1 to 4] -
        print() # Add an extra line for looks
        return choice

    @staticmethod
    def print_current_list_items(list_of_rows: list):
        """ Print the current items in the list of Employee rows

        :param list_of_rows: (list) of rows you want to display
        """
        print("***** The current items employees are: *****")
        for row in list_of_rows:
            print(str(row.employee_id)
                  + " "
                  + row.first_name
                  + " "
                  + row.last_name)
        print("*****")
        print() # Add an extra line for looks

    @staticmethod
    def input_employee_data():
        """ Gets data for an employee object

        :return: (employee) object with input data
        """

```

Pic 3: Code for IO class

Main Module

I started with the code from listing 13. At first, I created an if ... else statement by using the `__name__` variable to check that the main.Py script was running as the main program. If the script was not running from the main program, an error message will be displayed. I choose to use the 'from module name import as name' method to import a specific class within the module. And then I added a try ... except statement to read in the data from the EmolyeeData text file. If not, exception is raised, then a while loop is executed that calls the `print_menu_items()` and the `input_menu_options()` functions from the IOClasses module. Once user choose a menu of choice, if ... elf statement is used to call each function from module (either the ProcessingClasses or IOClasses) that which is associated to the menu of options. If the user choose option 1, Current list of items is displayed and if user choose option 2 user can enter new employee ID's, if option 3 is chosen entered data can be save to the file and if option 4 is chosen user exits the program. Check the image below

```

# ----- #
# Title: Assignment 09
# Description: Working with Modules

# ChangeLog (Who,When,What):
# RRoot,1.1.2030,Created started script
# RRoot,1.1.2030,Added pseudo-code to start assignment 9
# Naga Anusha,14.3.2022,Modified code to complete assignment 9
# ----- #
from builtins import Exception, print

if __name__ == "__main__":
    from DataClasses import Employee as Emp
    from ProcessingClasses import FileProcessor as Fp
    from IOClasses import EmployeeIO as Eio
else:
    raise Exception("This file was not created to be imported")

# Main Body of Script ----- #
# ----- Data ----- #
lstEmployeeTable = [] # A list of Employee objects
lstFileData = [] # A list of string objects in a list
# TODO: Add Data Code to the Main body
# Load data from file into a list of employee objects when script starts
lstFileData = Fp.read_data_from_file("EmployeeData.txt")
for row in lstFileData:
    lstEmployeeTable.append(Emp(row[0], row[1], row[2].strip()))

# Show user a menu of options
while True:
    Eio.print_menu_items()
    # Get user's menu option choice
    strOption = Eio.input_menu_options()
    if strOption == "1":
        # Show user current data in the list of employee objects
        Eio.print_current_list_items(lstEmployeeTable)
        continue
    elif strOption == "2":
        # Let user add data to the list of employee objects
        lstEmployeeTable.append(Eio.input_employee_data())
        continue
    elif strOption == "3":
        # let user save current data to file
        Fp.save_data_to_file("EmployeeData.txt", lstEmployeeTable)
        print("Data was saved")
        continue
    elif strOption == "4":
        print("Program finished good bye")
        break # user exit program
# Main Body of Script ----- #

```

Pic 4: Code for Main.py

Below are the results of the program in PyCharm and terminal

```
/Users/nagaanushaperali/Documents/_Pythonclass/Assignment9/venv/bin/py
```

```
Menu of Options
```

- 1) Show current employee data
- 2) Add new employee data.
- 3) Save employee data to File
- 4) Exit program

```
Which option would you like to perform? [1 to 4] - 1
```

```
***** The current items employees are: *****
```

```
1,Bob,Smith
```

```
2,Sue,Jones
```

```
33,Tom,Jerry
```

```
4,James,Robert
```

```
555,Eee,Rrr
```

```
*****
```

Pic 5: Displaying the current data

```
Menu of Options
```

- 1) Show current employee data
- 2) Add new employee data.
- 3) Save employee data to File
- 4) Exit program

```
Which option would you like to perform? [1 to 4] - 2
```

```
What is the employee Id? - 6
```

```
What is the employee First Name? - abcd
```

```
What is the employee Last Name? - efg
```

Pic 6: Adding new employee data

```
Menu of Options
1) Show current employee data
2) Add new employee data.
3) Save employee data to File
4) Exit program
```

```
Which option would you like to perform? [1 to 4] - 3
```

```
Data was saved
```

```
Menu of Options
1) Show current employee data
2) Add new employee data.
3) Save employee data to File
4) Exit program
```

```
Which option would you like to perform? [1 to 4] - 4
```

```
Program finished good bye
```

```
Process finished with exit code 0
```

Pic 7: saving and exiting the program

```
Assignment9 — Python Main.py — 80x24

3) Save employee data to File
4) Exit program

Which option would you like to perform? [1 to 4] - 1

***** The current items employees are: *****
1,Bob,Smith
2,Sue,Jones
33,Tom,Jerry
4,James,Robert
555,Eee,Rrr
6,Abcd,Efg
*****

Menu of Options
1) Show current employee data
2) Add new employee data.
3) Save employee data to File
4) Exit program

Which option would you like to perform? [1 to 4] -
```

Pic 8: Results in terminal

Summary

In this assignment by referring to module 9 video and by referring the information in the text book and some webpages I learned how to import one module into another, and testing it in the Test Harness.py I was able to create the program successfully.