Jason Noumeh
November 26, 2023
IT FDN 110 A
Assignment06
https://github.com/WaThaFAQ/IntroToProg-Python-Mod06

Course Registration - Functions and Classes

Introduction

Assignment 06 involves creating a Python script which gives the user the following options:

1. Register a student for a course
2. Show current data
3. Save data to a file
4. Exit the program

In addition to the above actions, the program will initially open up an existing file named "**Enrollments.json**", store the existing data in a variable, and write it back to the truncated file with any additional data the user provides.

Code

The decision making process of the code allowing the user to have multiple options is handled through **if** statements. The ability given to the user to make multiple choices is handled through an infinite looping process using the **while TRUE** statement, until the user decides to close the program.

```
# ------------------------------------------------------------------------------------------ #
# Title: Assignment06
# Desc: This assignment demonstrates using functions
# with structured error handling
# Change Log: (Who, When, What)
#   Jason Noumeh,11/26/2023,Created Script
#   <Your Name Here>,<Date>,<Activity>
# ------------------------------------------------------------------------------------------ #
```

Figure 1: Script Header

```python
# Present and Process the data
while True:  # Will continue the iterative process indefinitely

    # Presents user with menu of options
    IO.output_menu(menu=MENU)

    # Stores user menu choice
    menu_choice = IO.input_menu_choice()

    if menu_choice == "1":  # Registers a student for a course
        IO.input_student_data(student_data=students)
        continue
    if menu_choice == "2":  # Show current data
        IO.output_student_courses(student_data=students)
        continue
    if menu_choice == "3":  # save data to file
        FileProcessor.write_data_to_file(file_name=FILE_NAME, student_data=students)
        continue
    if menu_choice == "4":  # exits program
        break
```

Figure 2: Script Processing Overview

A more organized way of laying out the program using classes and functions has been introduced in this assignment. In addition, using the separation of concerns pattern, the script is divided into three sections: Data, Processing, and Presentation.

Data

The code begins with the initialization of constants and variables. Constants are set with predefined values which do not change, and variables are initialized to empty values with a string or list data type. Note the usage of triple apostrophe for the purpose of preserving the format of multi-line strings.

In addition to the data, the json module is imported to make handling the files more convenient.

```python
import json

# Data ---------------------------------------------- #
# Define the Data Constants
MENU: str = '''
---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------
'''
FILE_NAME: str = "Enrollments.json"

# Define the Data Variables
students: list = []  # a table of student data
menu_choice: str = ''  # hold the choice made by the user.
```

Figure 3: Initialization of Constants and Variables

Processing

This section of the code contains the class **FileProcessor** containing two static methods. The first method, **read_data_from_file**, carries out the function of opening a specified file with pre-existing data and storing this to a parameter called **student_data**. The other method, **write_data_to_file**, writes the data from a parameter called **student_data** to a specified file. Both of these methods contain structured error handling.

```python
# Processing ---------------------------------------- #
2 usages
class FileProcessor:
    """A collection of processing layer functions that work with Json files..."""

    1 usage
    @staticmethod
    def read_data_from_file(file_name: str, student_data: list):
        """..."""
        try:
            file = open(file_name, "r")
            student_data = json.load(file)
            file.close()
        except FileNotFoundError as e:
            IO.output_error_messages( message: "Text file must exist before running this script!", e)
        except Exception as e:


            IO.output_error_messages( message: "There was a non-specific error!", e)
        finally:
            if file.closed == False:
                file.close()
        return student_data

    1 usage
    @staticmethod
    def write_data_to_file(file_name: str, student_data: list):
        """..."""
        try:
            file = open(file_name, "w")
            json.dump(student_data, file)
            file.close()
        except TypeError as e:
            IO.output_error_messages( message: "Please check that the data is a valid JSON format", e)
        except Exception as e:
            IO.output_error_messages( message: "There was a non-specific error!", e)
        finally:
            if file.closed == False:
                file.close()
```

Figure 4: Processing

Presentation

This section of the script contains the class **IO** which handles the static methods used to manage user input and output.

```python
class IO:
    """
    A collection of presentation layer functions that manage user input and output

    ChangeLog: (Who, When, What)
    JNoumeh,11/25/2023,Created Class
    JNoumeh,11/25/2023,Added menu output and input functions
    JNoumeh,11/25/2023,Added a function to display the data
    JNoumeh,11/25/2023,Added a function to display custom error messages
    """
    pass
```

Figure 5: Presentation: Class IO

The first method, **output_error_message**, develops a template for error handling to be called in other portions of methods containing exception blocks. This block of code prints the error, any documentation associated with it and its type.

```python
@staticmethod
def output_error_messages(message: str, error: Exception = None):
    """
    This function displays a custom error messages to the user.

    ChangeLog: (Who, When, What)
    JNoumeh,11/25/2023,Created method

    """
    print(message, end="\n\n")
    if error is not None:
        print("-- Technical Error Message -- ")
        print(error, error.__doc__, type(error), sep='\n')
```

Figure 6: IO Class - Error Message Method

The next method, **output_menu**, prints the menu out to the user. Naturally, the following method, input_menu_choice, requests the option from the menu the user desires. An **if** statement assists the user in making a choice the script can acknowledge. The stored variable **choice** is then returned to make it globally accessible.

```python
@staticmethod
def output_menu(menu: str):
    """
    This method prints the menu to the user.

    ChangeLog: (Who, When, What)
    JNoumeh,11/25/2023,Created Method
    """
    print()
    print(menu)
    print()

1 usage
@staticmethod
def input_menu_choice():
    """
    This method collects the users choice after being presented the menu and
    assigns it to the variable choice and returns the value to the main
    module.

    ChangeLog: (Who, When, What)
    JNoumeh,11/25/2023,Created Method
    """
    choice: str = "0"
    try:
        choice = input("Enter your menu choice number: ")
        if choice not in ("1","2","3","4"):  # Note these are strings
            raise Exception("Please, choose only 1, 2, 3, or 4")
    except Exception as e:
        IO.output_error_messages(e.__str__())  # Not passing the exception object to avoid the technical message
    return choice
```

Figure 7: IO Class - Output/Input Menu

In anticipation of the first menu choice which allows the user to input a registration for a student, a method named **input_student_data** is created. This method requests the necessary information from the user while using structured error handling to help the user avoid any errors in inputting names.

```python
@staticmethod
def input_student_data(student_data: list):
    """
    This method gets triggered when the user chooses option 1 which collects
    the students first, last and course name from the user. These values are
    then appended to the variable student_data which contains initial data.

    ChangeLog: (Who, When, What)
    JNoumeh,11/25/2023,Created Method
    """
    try:
        # Input the data
        student_first_name = input("What is the student's first name? ")
        if not student_first_name.isalpha():
            raise ValueError("The first name should not contain numbers.")

        student_last_name = input("What is the student's last name? ")
        if not student_last_name.isalpha():
            raise ValueError("The last name should not contain numbers.")

        course_name = input("Please enter the name of the course: ")

        student = {"FirstName": student_first_name,
                   "LastName": student_last_name,
                   "CourseName": course_name}
        student_data.append(student)

    except ValueError as e:
        IO.output_error_messages( message: "That value is not the correct type of data!", e)
    except Exception as e:
        IO.output_error_messages( message: "There was a non-specific error!", e)
    return student_data
```

Figure 8: IO Class - User Inputs

The final method in the class **IO** is **output_student_courses**. This method uses a list parameter to print out all stored data. The intent is for the data to be a combination of initial data located on the file and the addition of user input.

```python
@staticmethod
def output_student_courses(student_data: list):
    """...."""
    print("-" * 50)
    for student in student_data:
        print(f'Student {student['FirstName']} {student['LastName']} is enrolled in {student['CourseName']}')
    print('-' * 50)
```

Figure 9: IO Class - Registration Output

Process

```python
# When the program starts, read the file data into a list of dictionaries (table)
students = FileProcessor.read_data_from_file(file_name=FILE_NAME,student_data=students)


# Present and Process the data
while True:  # Will continue the iterative process indefinitely

    # Presents user with menu of options
    IO.output_menu(menu=MENU)

    # Stores user menu choice
    menu_choice = IO.input_menu_choice()

    if menu_choice == "1":  # Registers a student for a course
        IO.input_student_data(student_data=students)
        continue
    if menu_choice == "2":  # Show current data
        IO.output_student_courses(student_data=students)
        continue
    if menu_choice == "3":  # save data to file
        FileProcessor.write_data_to_file(file_name=FILE_NAME, student_data=students)
        continue
    if menu_choice == "4":  # exits program
        break
```

Figure 10: Script Process

Figure 10 shows the process of the script utilizing the classes and methods previously discussed. The first step occurs before the while loop is initiated. That is to read and store the data from the **FILE_NAME** constant set to "**Enrollments.json**". The variable list **students** receives its first set of data in this line of code.

The next line begins the indefinite loop (**while True**) of presenting the menu of options then processing which option the user chooses with a series of **if** statements. The returned local variable **choice** is assigned to the global variable **menu_choice** through **IO.input_menu_choice()**.

If the user decides to register a student for a course (Option 1), the data gets appended to the global variable **students**. This will add the data to the previously stored data read from the json file. Option 2 displays all information stored in **students** at the time of selection. Option 3 will write this information back to a truncated "**Enrollments.json**" file.

Summary

The Python script is able to read a json file with existing data, save that data to a list, add additional data per user input and finally write this information back to the same file and save it. The following attached pages display the successful run of the code within both the Pycharm IDE and Command Prompt.

Pycharm Run

```
---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------


Enter your menu choice number: 1
What is the student's first name? John
What is the student's last name? Jacob
Please enter the name of the course: Python 100


---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------


Enter your menu choice number: 2
-------------------------------------------------
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student John Jacob is enrolled in Python 100
-------------------------------------------------


---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
```

```
Enter your menu choice number: 1
What is the student's first name? Stacey
What is the student's last name? Sanchez
Please enter the name of the course: Python 100


---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------
```

```
Enter your menu choice number: 2
-------------------------------------------------
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student John Jacob is enrolled in Python 100
Student Stacey Sanchez is enrolled in Python 100
-------------------------------------------------


---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------


Enter your menu choice number: 3


---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------


Enter your menu choice number: 4

Process finished with exit code 0
```

Enrollments.json

File Edit View


```
[{"FirstName": "Bob", "LastName": "Smith", "CourseName": "Python 100"},
{"FirstName": "Sue", "LastName": "Jones", "CourseName": "Python 100"},
{"FirstName": "John", "LastName": "Jacob", "CourseName": "Python 100"},
{"FirstName": "Stacey", "LastName": "Sanchez", "CourseName": "Python 100"}]
```


Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Command Prompt Run

```
C:\Users\Laptop>cd C:\Users\Laptop\Documents\IT FDN 110 A - Foundations of Programming - Python\Lab Scripts\Mod6

C:\Users\Laptop\Documents\IT FDN 110 A - Foundations of Programming - Python\Lab Scripts\Mod6>python Assignment06.py


---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------


Enter your menu choice number: 1
What is the student's first name? Kyle
What is the student's last name? Hong
Please enter the name of the course: Python 100


---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------


Enter your menu choice number: 3


---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------


Enter your menu choice number: 2
--------------------------------------------------
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Kyle Hong is enrolled in Python 100
--------------------------------------------------


---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------


Enter your menu choice number: 4

C:\Users\Laptop\Documents\IT FDN 110 A - Foundations of Programming - Python\Lab Scripts\Mod6>
```

Enrollments.json | Enrollments.json

File Edit View


```
[{"FirstName": "Bob", "LastName": "Smith", "CourseName": "Python 100"},
{"FirstName": "Sue", "LastName": "Jones", "CourseName": "Python 100"},
{"FirstName": "Kyle", "LastName": "Hong", "CourseName": "Python 100"}]
```


Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8