Name: Gustavo Limongi Araujo

Date: 02/19/2024

Course: Introduction to Python

Assignment 06: Introduction to Programming with Python

Intro

Create a Python program that demonstrates using constants, variables, and print statements to display a message about a student's registration for a Python course. This program is very similar to Assignment05, but **It adds the use of functions, classes, and using the separation of concerns pattern.**

Note: Start by opening and reviewing the starter file Assignment06-Starter.py!

Topic

Using PyCharm as an IDE, the header is the following:

The menu choice and the file name were defined below the header:

```
# 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

----------------------------------------
"""
```

The While loop is used in conjunction with 3 functions to call options 1 to 3:

1) Function **input student data**: read data from student using the *input* function
2) Function **show student data**: print what was stored in the variables above.
3) Function **save_data_to_file**: save the data in the FILE_NAME in the json format file

```
# Main program loop
while menu_choice != '4':
    IO.output_menu(MENU)
    menu_choice = IO.input_menu_choice()

    if menu_choice == '1':
        IO.input_student_data(students)
    elif menu_choice == '2':
        IO.output_student_courses(students)
    elif menu_choice == '3':
        FileProcessor.write_data_to_file(FILE_NAME, students)
        print("Data saved to file.")
    elif menu_choice == '4':
        print("Exiting program...")
    else:
        IO.output_error_messages("Invalid menu choice. Please select a valid option.")
```

The static decorator was used before each function as instructed and 2 classes were used, one called File Processor to read and write data in the json file and the other one called class IO to call the functions to display the menu choice, read data (option 1), show the data in a dictionary format (option 2) and save the data (3).

There is error handling for readind and writing the data as well as if the input names are empty.

CLASS FileProcessor:

```
class FileProcessor:
    """Processes file data"""

    @staticmethod
    def read_data_from_file(file_name: str):
        """Reads data from a file"""
        try:
            with open(file_name, 'r') as file:
                data = json.load(file)
        except FileNotFoundError:
            data = []
        except Exception as e:
           IO.output_error_messages("Error readding data from file",e)
           data=[]
        return data

    @staticmethod
    def write_data_to_file(file_name: str, data: list):
        """Writes data to a file"""
        try:
            with open(file_name, 'w') as file:
                json.dump(data, file)
        except Exception as e:
            IO.output_error_messages("Error writing data to a file", e)
```

Class IO:

```
    @staticmethod
    def output_error_messages(message: str, error: Exception = None):
        """Displays error messages"""
        print(f"Error: {message}")
        if error:
            print(f"Details: {error}")

    @staticmethod
    def output_menu(menu: str):
        """Displays the program menu"""
        print(menu)

    @staticmethod
    def input_menu_choice():
        """Prompts the user for a menu choice"""
        return input("Enter your choice: ")

    @staticmethod
    def output_student_courses(student_data: list):
        """Displays student data"""
        for student in student_data:
            print(f"Student: {student['first_name']} {student['last_name']} - Course: {student['course']}")

    @staticmethod
    def input_student_data(student_data: list):
        """Prompts the user to enter student data"""
        while True:
            first_name = input("Enter student's first name: ")
            if first_name:
                break
            else:
                print("First name can not be empty")
        while True:
            last_name = input("Enter student's last name: ")
            if last_name:
                break
            else:
                print("Last name can not be empty")
        while True:
            course = input("Enter course name: ")
            if course:
                break
            else:
                print("Course name can not be empty")
        student_data.append({'first_name': first_name, 'last_name': last_name, 'course': course})
```

The program was tested in IDE and Gitbash and worked as intended.

It is shared in Github and link posted in the module 06.

Summary

The videos and course notes of Lesson 06 were used to help write this code.

The code was tested with PyCharm and gitbash and worked as intended.