Jiahua (Joey) Zhang

6/16/2025

IT FDN 110 A

Assignment 08

GitHub Link

Employee Ratings Application

Introduction

This file is to document the thought process and steps in creating and testing an application to list multiple employee performance reviews to and from a JSON file using main, class, processing and presentation python files. The first step of creating this program is creating a new JSON file named EmployeeRatings.json that contains employee review data for two employees with data types that match the assignment requirements (e.g. the date is YYYY-MM-DD and the rating is an integer instead of a string). The goal of module 9 is to illustrate how redesigning an existing program into separate modules increases code flexibility and reusability.

Applications and Modules

Previously, we learned how to organize code within a python single file. An application includes multiple code files that perform a set of functions. In most cases, these code files are used by other code files which are known as code modules. The current employee ratings application uses code modules to split the initial assignment code into the main module, data classes, processing classes and presentation classes into their own files where the main module calls from the other three modules to run the program using the "import" command.

To simplify the module names, I used import aliases to shorten the module names to avoid naming conflicts. For example, the "data_classes" module is renamed to "data" using the import as command. When we call from other modules, we need to insert the module alias before any line of code that calls from the designated module.

Designing Applications

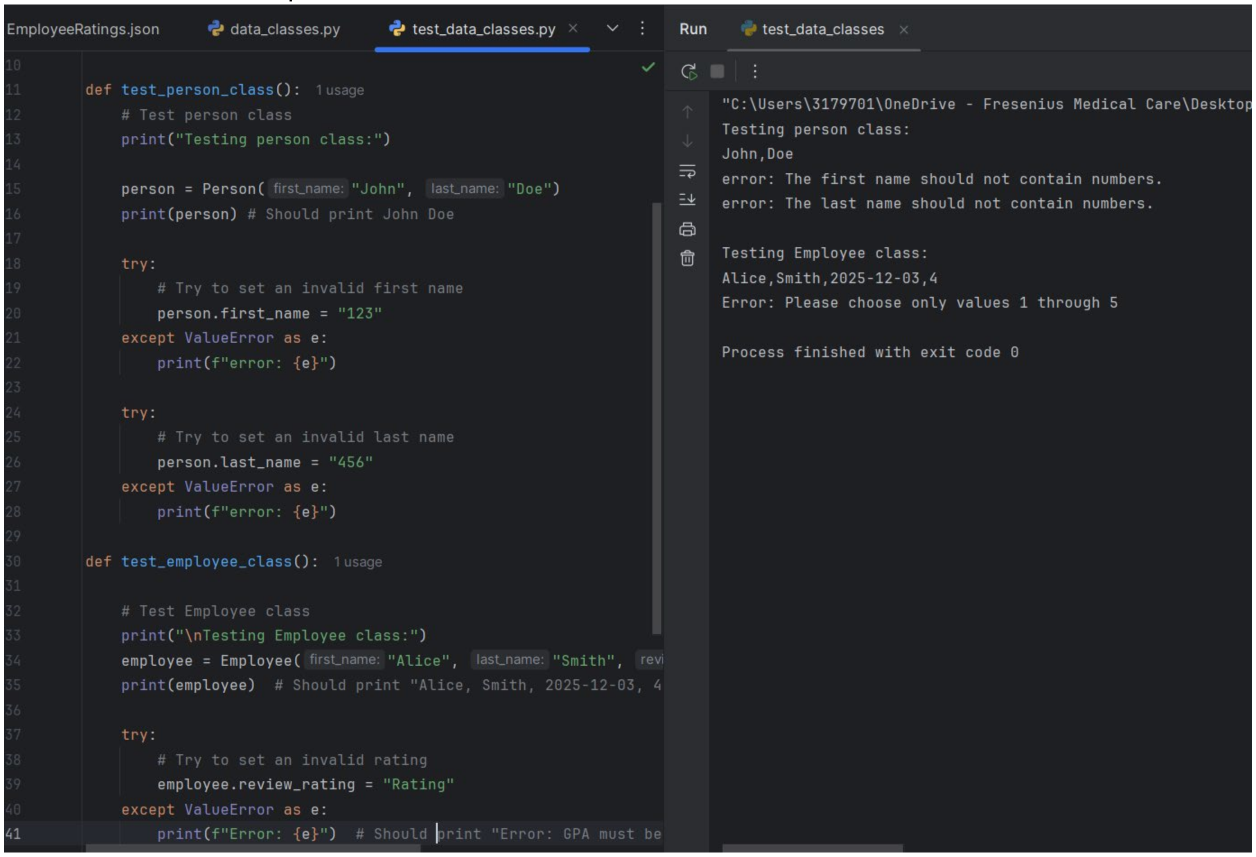
To properly design the employee reviewing application, I determined that the program needs an existing JSON file that can be read from and written to and that there are four properties that need defining (first name, last name, review date that defaults to "1900-01-01" and review rating that defaults to 3). Next. the architectural design, mentioned above, is where I took the assignment template and broke it into separate modules.

The following steps I took were to assess the relationships between all of the classes and objects created from classes. The employee class exists independently of the review rating, but the employee list object cannot exist independently of the employee object where it requires at least one.

Testing and Quality Assurance

During office hours, Becky and I were unit testing my code to figure out why one of my parameters was not callable and used breakpoints to step into the code. Through testing my code, Becky offered some guidance on another way to design the application where most of the heavy lifting comes from the main module.

See below for an example of data class test harness:



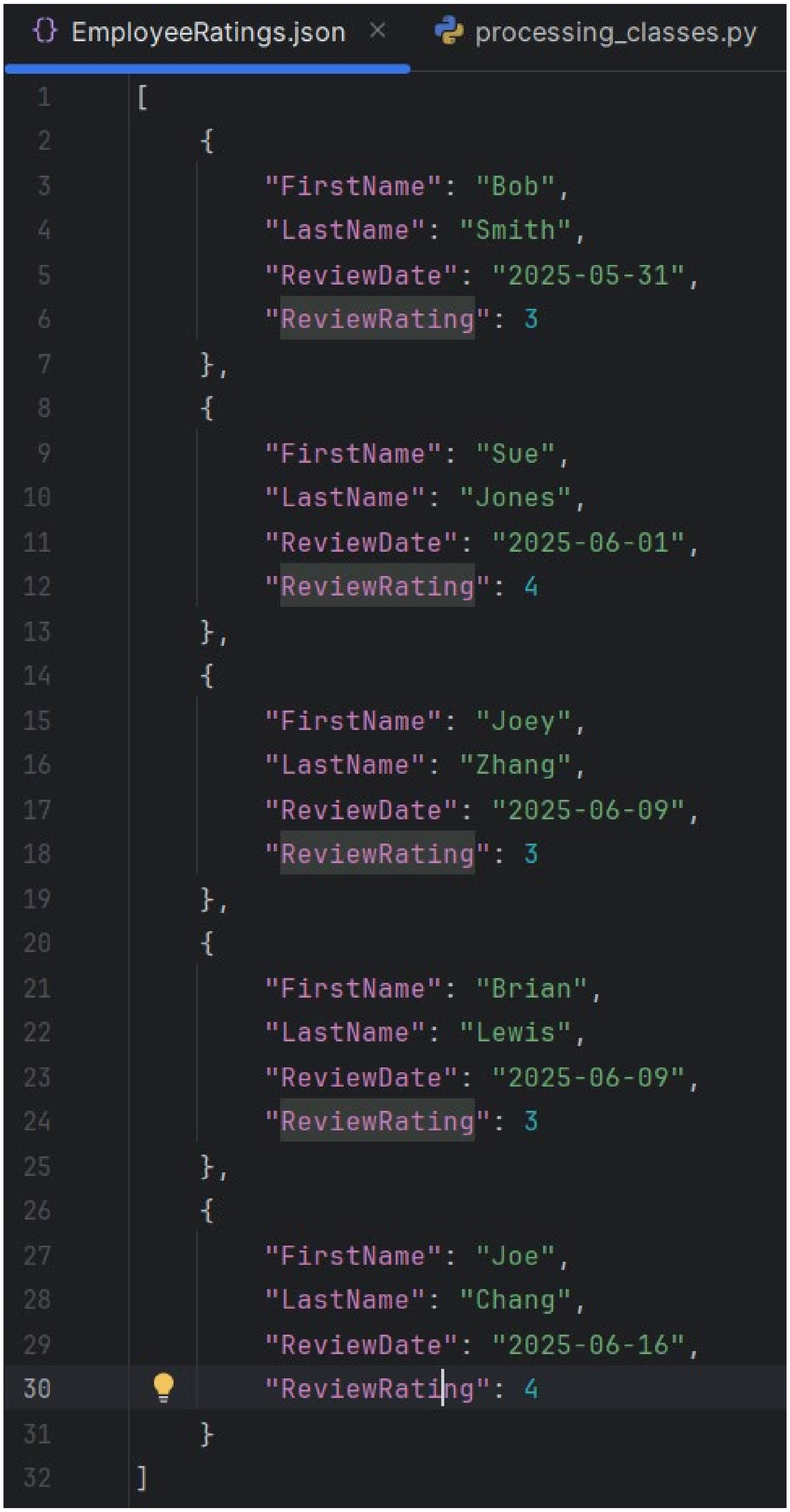
```
def test_person_class():  1 usage
    # Test person class
    print("Testing person class:")

    person = Person( first_name: "John",  last_name: "Doe")
    print(person) # Should print John Doe

    try:
        # Try to set an invalid first name
        person.first_name = "123"
    except ValueError as e:
        print(f"error: {e}")

    try:
        # Try to set an invalid last name
        person.last_name = "456"
    except ValueError as e:
        print(f"error: {e}")

def test_employee_class():  1 usage

    # Test Employee class
    print("\nTesting Employee class:")
    employee = Employee( first_name: "Alice",  last_name: "Smith",  revi
    print(employee)  # Should print "Alice, Smith, 2025-12-03, 4

    try:
        # Try to set an invalid rating
        employee.review_rating = "Rating"
    except ValueError as e:
        print(f"Error: {e}")  # Should print "Error: GPA must be
```

```
"C:\Users\3179701\OneDrive - Fresenius Medical Care\Desktop
Testing person class:
John,Doe
error: The first name should not contain numbers.
error: The last name should not contain numbers.

Testing Employee class:
Alice,Smith,2025-12-03,4
Error: Please choose only values 1 through 5

Process finished with exit code 0
```

Running the Application

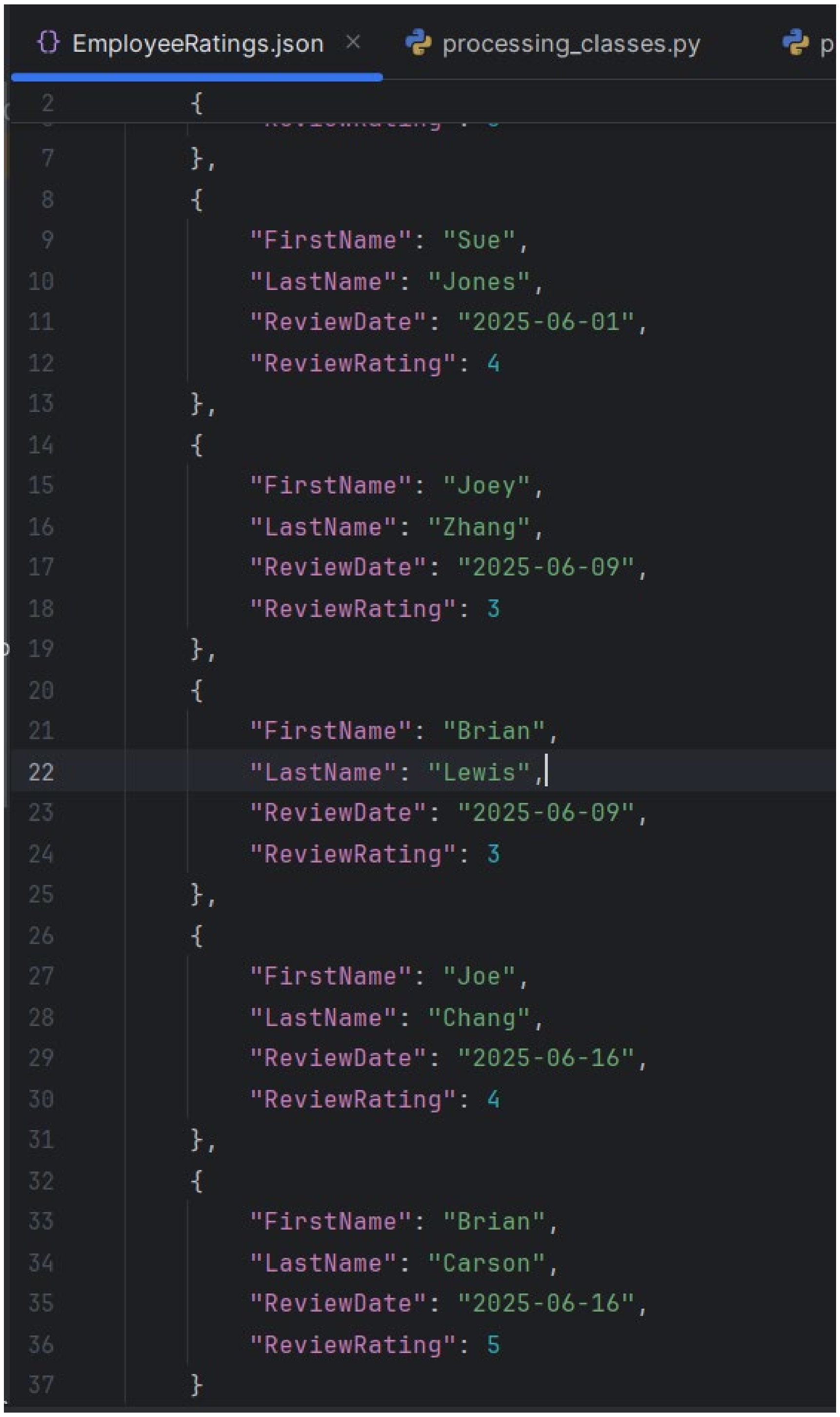
PyCharm Test with JSON output

Before running application:



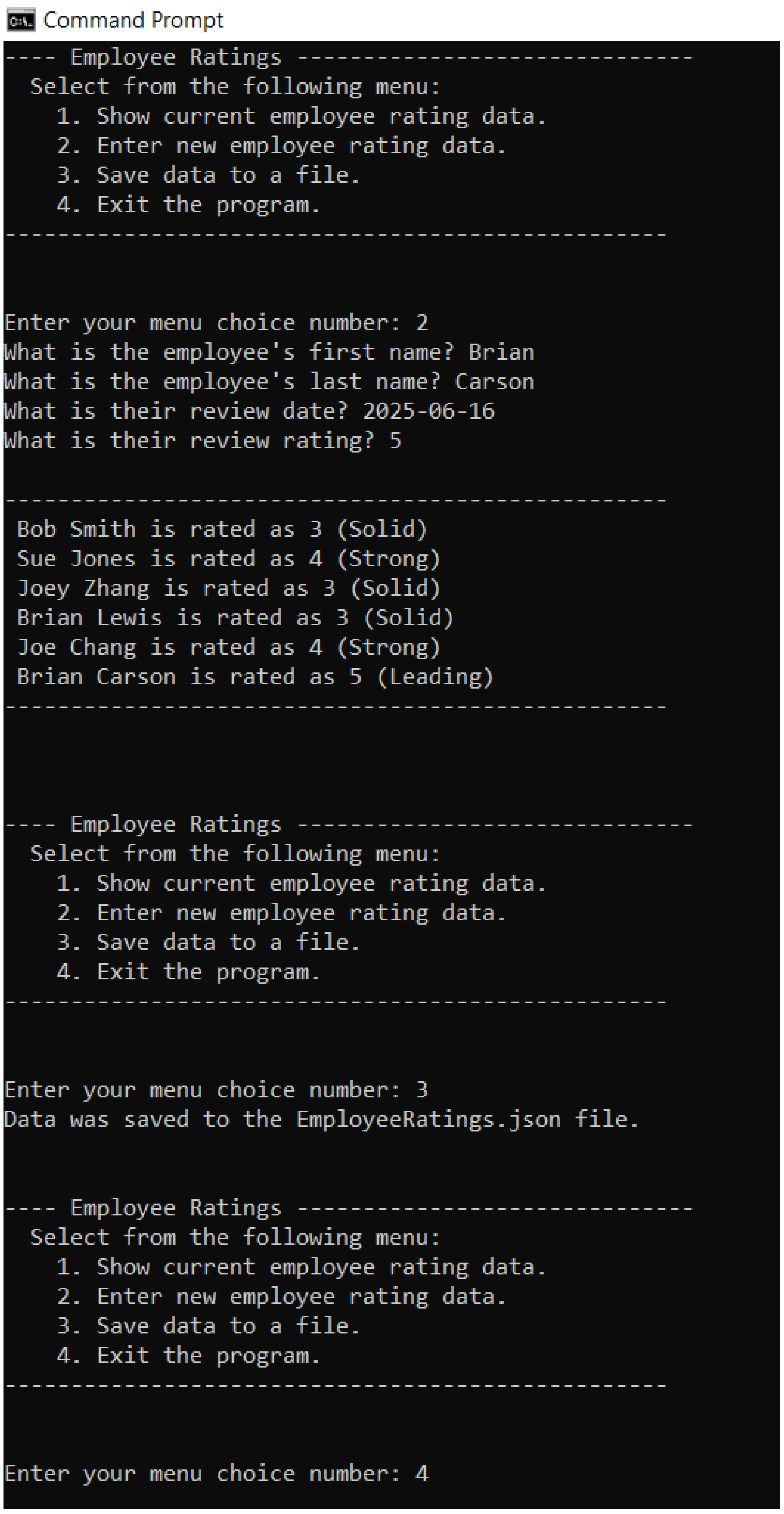
```json
[
    {
        "FirstName": "Bob",
        "LastName": "Smith",
        "ReviewDate": "2025-05-31",
        "ReviewRating": 3
    },
    {
        "FirstName": "Sue",
        "LastName": "Jones",
        "ReviewDate": "2025-06-01",
        "ReviewRating": 4
    },
    {
        "FirstName": "Joey",
        "LastName": "Zhang",
        "ReviewDate": "2025-06-09",
        "ReviewRating": 3
    },
    {
        "FirstName": "Brian",
        "LastName": "Lewis",
        "ReviewDate": "2025-06-09",
        "ReviewRating": 3
    },
    {
        "FirstName": "Joe",
        "LastName": "Chang",
        "ReviewDate": "2025-06-16",
        "ReviewRating": 4
    }
]
```

After running application:



```
        {
        },
        {
            "FirstName": "Sue",
            "LastName": "Jones",
            "ReviewDate": "2025-06-01",
            "ReviewRating": 4
        },
        {
            "FirstName": "Joey",
            "LastName": "Zhang",
            "ReviewDate": "2025-06-09",
            "ReviewRating": 3
        },
        {
            "FirstName": "Brian",
            "LastName": "Lewis",
            "ReviewDate": "2025-06-09",
            "ReviewRating": 3
        },
        {
            "FirstName": "Joe",
            "LastName": "Chang",
            "ReviewDate": "2025-06-16",
            "ReviewRating": 4
        },
        {
            "FirstName": "Brian",
            "LastName": "Carson",
            "ReviewDate": "2025-06-16",
            "ReviewRating": 5
        }
```

Command Terminal Test (see next page):


Command Prompt

```
---- Employee Ratings ------------------------------
  Select from the following menu:
    1. Show current employee rating data.
    2. Enter new employee rating data.
    3. Save data to a file.
    4. Exit the program.
--------------------------------------------------



Enter your menu choice number: 2
What is the employee's first name? Brian
What is the employee's last name? Carson
What is their review date? 2025-06-16
What is their review rating? 5

--------------------------------------------------
 Bob Smith is rated as 3 (Solid)
 Sue Jones is rated as 4 (Strong)
 Joey Zhang is rated as 3 (Solid)
 Brian Lewis is rated as 3 (Solid)
 Joe Chang is rated as 4 (Strong)
 Brian Carson is rated as 5 (Leading)
--------------------------------------------------




---- Employee Ratings ------------------------------
  Select from the following menu:
    1. Show current employee rating data.
    2. Enter new employee rating data.
    3. Save data to a file.
    4. Exit the program.
--------------------------------------------------



Enter your menu choice number: 3
Data was saved to the EmployeeRatings.json file.


---- Employee Ratings ------------------------------
  Select from the following menu:
    1. Show current employee rating data.
    2. Enter new employee rating data.
    3. Save data to a file.
    4. Exit the program.
--------------------------------------------------



Enter your menu choice number: 4
```

Summary

Module 8 merges all the previous concepts together and transforms the file into separate modules called an application (or program) to enhance flexibility and reusability. It also teaches how to do unit testing and use a test harness. All in all, this module ties together the core concepts of Python programming and defines a solid foundation for intermediate and eventually advanced learning of the programming language.