Shuhua Liang

2022-11-30

IT FDN 110 A

Assignment07

https://github.com/sliang2022/IntroToProg-Python-Mod07

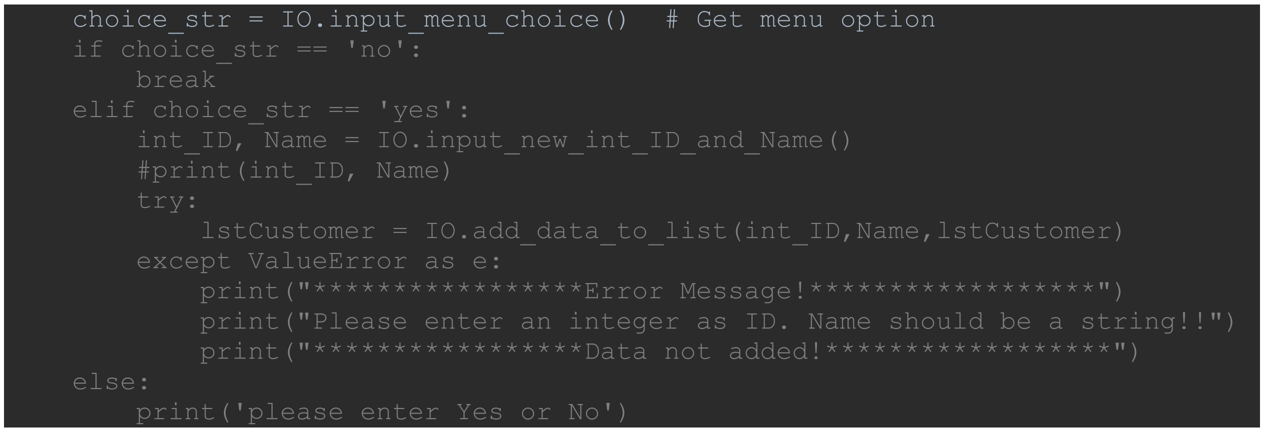
Exception handling and Pickling

Introduction

From this lecture, I learned (1) How to handle exception (2) How to use pickling to access data.

Exception handling

The homework task is to create a project and demonstrate how to handle exceptions. I decided to start my work based upon the template in module07 created by the professor. In the code I need to get an input ID and name from user, then store it in a list object (see fig. 1). There are two possible types of errors. (1) When I ask if the user need to add more data. The user needs to enter either “yes” or "no". If the user entered something else, I used an if else statement to ask the user try again. This is the common method I used to avoid errors before. (2) From this class, now I learned a second method, which is the “try exception” method. Using this method, a backup plan is used to handle an error without showing red colored message in the screen. This is a more direct way than the “if else” method I used before.



```
    choice_str = IO.input_menu_choice()   # Get menu option
    if choice_str == 'no':
        break
    elif choice_str == 'yes':
        int_ID, Name = IO.input_new_int_ID_and_Name()
        #print(int_ID, Name)
        try:
            lstCustomer = IO.add_data_to_list(int_ID,Name,lstCustomer)
        except ValueError as e:
            print("****************Error Message!*****************")
            print("Please enter an integer as ID. Name should be a string!!")
            print("****************Data not added!*****************")
    else:
        print('please enter Yes or No')
```

Figure 1, two ways to handle input errors: (1) When input is neither “Yes” or “No”. (2) When input ID is not an integer, or name not as a string. This is caught as an “ValueError”.

Pickling

In R, there is a method you can create a list and assign various typle of data into a datafile without worrying about redefine them when you read the saved data from csv file. In Python, it seems Pickling is a similar approach which can significantly simplify the read-write process. For example, in Fig. 2, I used pickle.dump to write a list into a dat file without needing to worry about how each line will be written in the dat file. As for read from the dat data file, I used pickle.load, it automatically reconstructed the list without reading line by line and reconstruct the data structure. I find this methodology extremely useful and efficient. One

```python
class Processor:
    def save_data_to_file(file_name, list_of_data):
        with open(file_name, 'wb') as f:
            pickle.dump(list_of_data, f)
        return list_of_data


    def read_data_from_file(file_name):
        with open(file_name, 'rb') as file_obj:
            list_of_data = pickle.load(file_obj)
        return list_of_data
```

Figure 2, Pickle.dump and Pickle.load can process data in a packaged way. It reduced the complexity of processing data.

Results

As shown in Fig. 3, this code asks the user to enter ID and Name, which are saved into a list. Then the list is stored into a binary file. In the end the code read the list from the dat file and print it out.

```
C:\_PythonClass\Assignment07\Assignment07_Shuhua_liang.py
Do you want to add new data? [Yes or No] - Yes
Add ID: 3872
Please enter Name: Shuhua
Do you want to add new data? [Yes or No] - Yes
Add ID: 9837
Please enter Name: Jason
Do you want to add new data? [Yes or No] - Yes
Add ID: James
Please enter Name: 3817
*****************Error Message!******************
Please enter an integer as ID. Name should be a string!!
*****************Data not added!******************
Do you want to add new data? [Yes or No] - no
******* The current ID and Name list: *******
ID: 3872  Name: Shuhua
ID: 9837  Name: Jason
*******************************************


Process finished with exit code 0
```

Figure 3: Results: It asks you to enter an ID and a Name which are added to the list. And print them out in the end. If you do not enter the ID or Name in the proper format, the exception handling will ask you to try again.

Summary

I encountered some troubles implementing the pickling. But in the end, I was able to make it work. I find the pickling methodology extremely elegant. It prevents me from making mistakes when read/write data from/into data files. During my current job, we tried to standardize our workflow. However, most of the errors happened during reading data from csv/excel files due to formatting issues. I think pickling is a very useful tool in avoiding making mistakes during formatting data files.