<Ziqi Yan>

<2022/3/13>

<IT FDN 110>

<Assignment\_07>

CD Inventory

# Introduction

The Assignment 07 is about pickling(serializing) a python object and error handling. Based on the starter code, we finished the IO class definition and safe the code with exception handling (try-except statement).

# Topic\_1 Pickle module

There are two methods used in the program. IO.read\_file and IO.write\_file. Based on how they were used, those are static methods. After adding the decorator @staticmethod to those two methods, we are ready to fill the function.

Pickle is a python module, we will use it by using “import pickle”. Then we can use it to load object from the file or write object into file. After searching online, we have known how to use “load” and “dump” and applied these into code.

When we are using Pickle to read objects from file, we need to open files by using the mode “rb”, which means “read binary”; when writing objects, the mode should be “ab”, means “appending binary”.

Graphical user interface, text, application

Description automatically generated

Figure read\_file implementation

Graphical user interface, text, application

Description automatically generated

Figure 2 write\_file implementation

# Topic\_2 Exception Handling

For some operation which we are not sure about the correctness of the execution. We can use Exception Handling to make sure the logic flow won’t get messed up. In our program, first we have to safe the file operations, then, in the math class, we need to keep the divide function safe by detecting if the val2 is zero.

Graphical user interface, application

Description automatically generated

Figure 2 check if read\_file has succeed

Text

Description automatically generated

Figure 3 check if dividing numbers has succeed

# Topic\_3 tester

I have to make a tester program to test if the main code works fine. It is just simply writing tuple into file and read list from file and print elements.

Text

Description automatically generated

Figure 4 implementation of tester

# Summery

In this assignment. I learned how to pickling/unpickling python objects and how to implement exception handling. The concept is important to learn and understand. When I understand it, I feel like I can make my code more reliable and readable.

# Appendix

SimpleMath

@staticmethod

def get\_quotient(val1 = 0.0, val2 = 0.0):

try:

\_ = val1/val2

except Exception as error:

print("divide error")

raise error

IO

class IO:

"""A collection of the Input / Output operations """

@staticmethod

def read\_file(fileName):

"""

function to read in two numbers from file fileName and return these

Args:

fileName (string): file name to read the numbers from

Returns:

numA (int): first number in file fileName.

numB (int): second number in file fileName.

"""

inputFile = open(fileName, "rb")

obj = pickle.load(inputFile)

inputFile.close()

return obj

@staticmethod

def write\_file(fileName, results):

"""

function to write the math results to file fileName

Args:

fileName (string): file Name to write the results to.

results (list): The results

Returns:

None.

"""

outputFile = open(fileName, "ab")

pickle.dump(results,outputFile)

outputFile.close()

Input/Output

print('Basic Math script. Calculating the Sum, Difference, Product and Quotient of two numbers.')

try:

intNumA, intNumB = IO.read\_file(strFileInput)

except Exception as error:

print("read file failed!")

raise error

lstResults = []

lstResults.append(SimpleMath.get\_sum(intNumA, intNumB))

lstResults.append(SimpleMath.get\_diffference(intNumA, intNumB))

lstResults.append(SimpleMath.get\_product(intNumA, intNumB))

lstResults.append(SimpleMath.get\_quotient(intNumA, intNumB))

try:

IO.write\_file(strFileOutput, lstResults)

except Exception as error:

print("write file failed!")

raise error