第二次作业

编写一个程序来读取日志文件

在任意 log.txt 文件中搜索，当找到以“ERROR”或“WARNING:”开头的行时，从该行中提取错误类型（“ERROR”或“WARNING”后面紧跟的单词），并使用字典dictionary计算每种错误类型出现的次数。读取 log.txt 中的所有数据后，输出出现次数最多的错误类型。

确保程序对日志文件中发现的所有错误类型进行排序并显示，忽略大小写。

代码

```python
"""
* coding=utf-8
* python=3.11
* File: Experiment2.py
* Author: donghy23@mails.tsinghua.edu.cn
* Created: 2024-7-12
* Repo: https://github.com/FHYQ-Dong/Tsinghua-Program-Design-Assignment-
3/blob/main/Experiment2/Experiment2.py
"""

from functools import cmp_to_key
import re
import typing

def parse_log_error_warning(logfile: typing.TextIO) -> dict[str, dict[str,
int]]:
    """文档中给出的指示与 log.txt 中的内容不一致，面向 log.txt 中内容编写

    Args:
        logfile (io.TextIO): 文件句柄

    Returns:
        dict: 返回一个字典，包含 ERROR 和 WARNING 两个 key，对应的值是一个字典，包含错误码
和错误数量
    """
    result = {}
    # ErrorType = re.compile(r'\[[(ERROR)(error)]\]\s(.*)')
    ErrorType = re.compile(r'(?P<type>\[ERROR\]) \[(?P<info>[\dA-Za-z]+)\]')
    WarningType = re.compile(r'(?P<type>\[WARNING\]) \[(?P<info>[\dA-Za-z]+)\]')
    logtext = logfile.readlines()
    for line in logtext:
        errors = ErrorType.findall(line)
        warnings = WarningType.findall(line)
        if errors:
            if 'ERROR' not in result:
                result['ERROR'] = {}
            if errors[0][1] not in result['ERROR']:
                result['ERROR'][errors[0][1]] = 0
            result['ERROR'][errors[0][1]] += 1
        elif warnings:
            if 'WARNING' not in result:
                result['WARNING'] = {}
            if warnings[0][1] not in result['WARNING']:
                result['WARNING'][warnings[0][1]] = 0
            result['WARNING'][warnings[0][1]] += 1

    return result


def cmp_error_item(item1: tuple[str, str, int], item2: tuple[str, str, int]) ->
int:
    """
    Compare two error items based on their attributes.

    ### Args:
        item1 (tuple[str, str, int]): The first error item to compare.
        item2 (tuple[str, str, int]): The second error item to compare.

    ### Returns:
        int: A negative value if item1 is less than item2, a positive value if
item1 is greater than item2,
             and 0 if item1 is equal to item2.

    """
    return \
        item1[2] - item2[2] if item1[2] != item2[2] \
            else (item1[0] > item2[0]) - (item1[0] < item2[0]) if item1[0] !=
item2[0] \
                else (item1[1] > item2[1]) - (item1[1] < item2[1])


def main():
    """
    This function reads a log file, parses the log for error and warning
details,
    and prints a sorted table of error types, error codes, and error counts.
    """
    with open('log.txt', 'r') as f:
        log_detail = parse_log_error_warning(f)

        log_detail_list = [
            (log_type, error_code, error_count)
            for log_type, error_item in log_detail.items()
            for error_code, error_count in error_item.items()
        ]
        log_detail_list.sort(key=cmp_to_key(cmp_error_item), reverse=True)
        col_width = [max(len(str(item[i])) for item in log_detail_list) for i in
range(3)]
        for i in range(3):
            col_width[i] = max(col_width[i], len(['ErrorType', 'ErrorCode',
'ErrorCount'][i])) + 1
        for i,item in enumerate(log_detail_list):
            print('{0:<{width0}}{1:<{width1}}{2:<{width2}}\n'.format(
                    "ErrorType", "ErrorCode", "ErrorCount", \
                    width0 = col_width[0], width1 = col_width[1], width2 =
col_width[2]
                ) \
                * (not bool(i)), end=''
            )
            print(('-' * (sum(col_width) + 5) + '\n') * (not bool(i)), end='')
            print('{0:<{width0}}{1:<{width1}}{2:<{width2}}'.format(
                    item[0], item[1], item[2], \
                    width0 = col_width[0], width1 = col_width[1], width2 =
col_width[2]
                )
            )


if __name__ == '__main__':
    main()
```

结果

输入1：

```
python ./Experiment2.py
```

输出1：

```
ErrorType ErrorCode          ErrorCount
---------------------------------------------
ERROR     DatabaseError      81
ERROR     FileNotFound       77
WARNING   CPUOverload        75
WARNING   HighMemoryUsage    71
ERROR     AccessDenied       70
ERROR     ConnectionTimeout  69
WARNING   DeprecatedFunction 67
WARNING   LowDiskSpace       66
ERROR     InvalidInput       63
WARNING   NetworkLatency     61
```