

Task 6.

Implement various text file operation

Aim:— To write a python program implement various text file operations.

Problem 6.1:-

You need to write the sentence "Error objects are thrown when runtime errors occur. The error object can also be used as a base object for user-defined exceptions" into a text file named log.txt. Implement a function that performs this task.

Algorithm:

1. write to a file
 - define write_file(filename) function.
 - open a file named "log.txt" in write mode.
 - write the following text to the file:
"Error objects are thrown when runtime errors occur. The error object can also be used as a base object for user-defined exceptions"
 - close the file.
2. Read the file:—
 - define read_file(filename) function.
 - open the specified by filename in read mode using a with statement.
 - Read the entire content of the file
 - print the content.
3. Execute the program:
 - call write_file("write") to write the predefined text to "log.txt".
 - call read_file("text") to attempt to read from a file named "text" and print its content

when a host error occurred.

The out put is nothing or System.out.println().

Errors objects are thrown when runtime errors occur.

The error object can also be used as a base object for user-defined exceptions.

Java has a class Exception which is the base class for all exceptions.

Java has a class Exception which is the base class for all exceptions.

interview

What is String?

String (String) is Java's class.

String is a class which is a representation of character arrays which are also called strings.

String is a class which is a representation of character arrays which are also called strings.

What is StringBuffer?

StringBuffer is a class which is a representation of character arrays which are also called strings.

Program 6.1

```
def write_file(filename):
    f = open("log.txt", "w")
    f.write("Error objects are thrown when runtime
errors occur. The errors object can also be used
as a base object for user-defined exceptions")
    f.close()

def read_file(filename):
    with open(filename, "r") as file:
        content = file.read()
        print(content)

write_file("write")
read_file("read")
```

Problem 6.2:-

You have a text file log.txt containing logs of a system, write a function that counts the number of lines containing the word "Error".

Algorithm:-

1. Initialize Error counter:
 - Define the function count_error_lines(filename):
 - Initialize error_count to 0.
2. Open and read file:
 - Open the file specified by filename in read mode using a with statement.
3. Check each line for "Error":
 - Loop through each line in the file:
 - If the line contains the word "Error"? increment error_count by 1.
4. Return error count:
 - After reading all the lines, return the values of error_count.

Output:  A stack of four cards representing a FIFO queue. The top card contains the value 10, the second card contains 20, the third contains 30, and the bottom contains 40.

5. execute the program:

- call `count_error_lines("log.txt")` to count the number of lines with the word "Error" in the file "log.txt".
- print the result with the message: "number of lines with 'Error': {error_lines}".

Program 6.2:

```
def count_error_lines(filename):  
    error_count = 0  
    with open(filename, "r") as file:  
        for line in file:  
            if "Error" in line:  
                error_count += 1  
    return error_count  
error_lines = count_error_lines("log.txt")  
print(f"Number of lines with 'Error': {error_lines}")
```

log.txt

"Error objects are known when runtime errors occur.
The Error object can also be used as a base object
for user-defined exceptions."

problem 6.3:

you need to write a report containing the details (name, departments) of the employee in list, write a python function that writes this report to a file named `employee-report.txt`.

Algorithm:

1. create employee data:
 - define the function `write_employee_report(filename)`:
 - create a list `employees` containing dictionaries, each with "name" and "department" keys for individual employees.

Output:-

Name: Alice, department: HR
Name: Bob, department: Engineering
Name: Charlie, department: Finance

- open file for writing:
 - open the file specified by filename in write mode using a with statement.
- write employee data to file:
 - loop through each employee in the employees list:
 - for each employee, format a string as "name: {employee['name']} , department: {employee['department']}".
 - write the formatted string to the file, followed by a new line character (\n).
- Execute the program:
 - call write_employee_report("employee-report.txt") to write the employee data to the file "employee-report.txt".

Program 6.3:-

```
def write_employee_report(filename):
    employees = [
        {"name": "Alice", "department": "HR"},
        {"name": "Bob", "department": "Engineering"},
        {"name": "Charlie", "department": "Finance"}]

    with open(filename, "w") as file:
        for employee in employees:
            line = f" name: {employee['name']}, department: {employee['department']}\n"
            file.write(line)
```

Example usage:

```
write_employee_report("employee-report.txt")
```

Result: Thus the python program implement various test file operations was successfully executed and the output

VEL TECH	
PERFORMANCE (5)	✓
RESULT AND ANALYSIS (5)	✓
VIVA VOCE (5)	✓
RECORD (5)	✗
TOTAL (20)	15
WITH DATE	10/10/2023