|  |
| --- |
| **파이썬프로그래밍및실습**  **Emotional**  **therapist**  **중간보고서 #1** |

|  |  |
| --- | --- |
|  | Date : 11/25  Name : 김세희  ID :233973 |

**1. Introduction (16 pt)**

**1) Background (14 pt)**

In these times of grayness, many people live without understanding their own emotions and feelings. If they continue without realizing their emotions, it could lead to significant issues for themselves in the future. To address this, I believe an emotional management program is necessary

**2) Project goal**

Analyzing the user's diary to rank emotions Aim to extract the first and second highest emotions and suggest solutions

**3) Differences from existing programs**

There is a difference from existing programs when it comes to ranking and recommending solutions to users rather than abstractly picking emotions by looking at the diary

**2. Functional Requirement**

**1) Analysis of the user's diary**

- Retrieve the user's diary and extract the user's emotional adjectives through various descriptive word

**(1) Detailed function**

- Rank the extracted emotions in order.

**2) Offer solutions for user emotions.**

- Provide solutions for the emotions ranked in the 1st and 2nd positions based on the ranked ranking

**(1) Detailed function 1**

- If the 1st and 2nd rankings are positive emotions, suggest the statement 'Today doesn't seem bad! How about trying healthy food and exercise for an even better day?

**(2) Detailed function 2**

- When there are many keywords related to death, a feature offering warm words and connecting to suicide prevention centers

**3. Progress (진척사항)**

**1) Function Implementation**

**(1) Analysis of the user's diary**

- Input: Enter the user's diary name.

- Output: X

- Description: Save the user's diary name as the "filename" variable, then retrieve the user's diary using the "load\_diary\_from\_file()" function. Divide the loaded diary into words, storing them in the "diary\_word" variable, and rank them using the "read\_diary\_word(diary\_list)" function.

- Applied Learned Content

Input function: Receives the user's diary name through the `input` function.

Functionization: Refactors complex program calculations into functions for improved readability.

Loop: Iterates through the user's diary sentences, extracting words one by one using a loop.

Conditional Statement: Checks whether the user's diary words are included in the emotion list using a conditional statement.

File I/O: Loads the user's file using the `readlines()` function and processes it line by line.

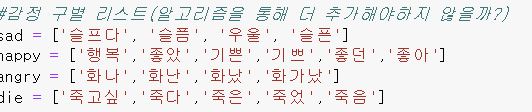
List: Creates a list of various emotions such as sad, happy, angry, and die.

Dictionary: Utilizes a dictionary to easily calculate the frequency of each emotion in the user's entries by associating emotions with their respective occurrence counts.

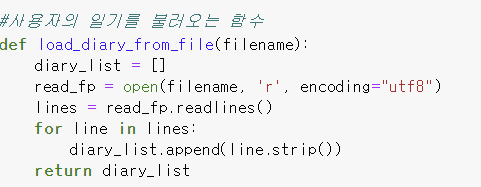
Diverse Function Usage: Employed various functions such as replace, split, sorted, and more.

- Code Screenshot

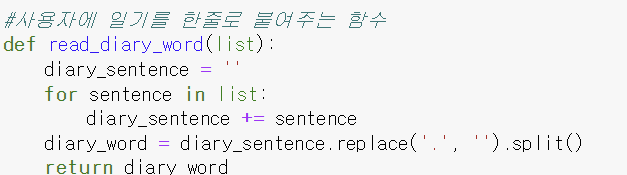
Emotion Discrimination List



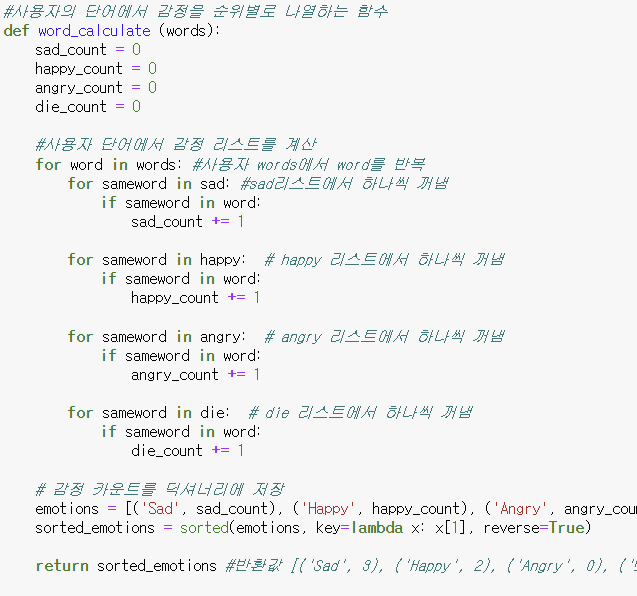
Function to Retrieve User's Diary



Function to Concatenate User's Diary into a Single Line and Tokenize into Words



Function to Rank Emotions in User's Words



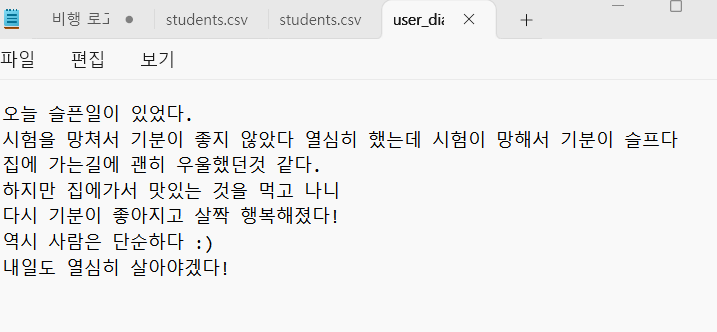
**2) Test Result**

**(1) Analysis of the user's diary**

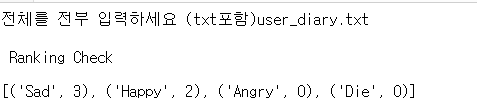
- Description: The received file from the user is loaded using the "load\_diary\_from\_file()" function. The "read\_diary\_word" function is then employed to read the file line by line, tokenize it into words, and finally, the "word\_calculate()" function is used to analyze emotions and rank them.

- Test Result Screenshot

Example User Diary



Ranked Results from User Diary

****

**4. Changes in Comparison to the Plan**

**1) Class Implementation and File Extension with VS Code**

- Before: Used Jupyter Notebook

- After: Expanded the program with VS Code

- Reason: Converted the program to a class-based structure for better organization and simplicity, especially with the emotion list and numerous functions.

**2)** **Afterward Features to Add**

1. Receive the user's location input and provide information on counseling center locations and contact numbers.

2. Enable automatic playback of meditation-related videos from websites.

3. Play music when the program is launched.

4. Implement a user feedback system.

5. Add features to enhance personal information protection.

**5. Schedule**

- 진행 상황 표기

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **업무** | **11/3** | **11/10** | **11/17** | **11/25** | **12/1** | **12/8** | **12/14** | **…** |
| **제안서 작성** | **제안서**  **제출** |  |  | **중간 보고서 제출** |  |  |  |  |
| **기능1** |  | **완료** | |  | **VS**  **Code**  **확장** |  |  |  |
| **기능2** |  |  | **진행 중** | |  |  |  |
| **추가기능** | **-** | | | | **추가**  **기능 1번** | **추가기능**  **2,3번** |  |