Python Project

Programming Assignment

**Title**: Creating a Note Using Python Class and Object

**Author**: Maha Vajeeshwaran Navaneethan

Completed on 27/10/2021

Table of Contents

1. Introduction........................................................................................................................3

2. Description………………..........................................................................................................3

3. User Guide...........................................................................................................................5

**Introduction:**

Designed an object-oriented program which allows to keep track of all notes that the user makes by using the class ” Note”. CRUD operations are created to help the user to create, read, update, and delete the notes. With the help of pickle module user shall save the created notes and read it from the text file.

This application's additional functionality allows the user to designate notes as completed or uncompleted based on his actions. User also could able to count the number of notes created, status of pending notes, number of days user took to complete the notes, and menu option to change the completion date.

**Description:**

A) Class Note:

Note class is created to keep the Notes information such as id, title, text, date, completed, datecomp. The datetime module is imported to set the date of note created.

* An \_\_int\_\_ method accepts arguments for a notes id, title, text, date, completed, datecomp.
* A set\_id method that sets the id attribute.
* A set\_ title method that sets the title attribute.
* A set\_ text method that sets the text attribute.
* A set\_ date method that sets the date attribute.
* A set\_ completed method that sets the completed attribute.
* A set\_ datecomp method that sets the datecomp attribute.
* A get\_id method that gets the id attribute.
* A get\_ title method that gets the title attribute.
* A get\_ text method that gets the text attribute.
* A get\_ date method that gets the date attribute.
* A get\_ completed method that gets the completed attribute.
* A get\_ datecomp method that gets the datecomp attribute.
* A note method that prints the objects.

B) Class Interface:

In Interface python file from Note.py file module class Note is imported, pickle module and from datetime module date is imported. Initially global constants for the menu choices are created. Now class Interface is created. Created a empty list ‘Note\_list’ to store all the notes information and can be retrieved for the later use. Menu function is created to print the menu option. It is set to accept the user input choice and action is performed based on the choice.

To save and read the notes created pickle module is used. In text document all the notes are saved in a binary. A while true loop is used to executes the menu choice till the user selects the Save and Exit. For each ‘select’ choice appropriate function is called. If the user enters the choice less than 1 or greater than 10 program will ask the user to enter the valid choice. To handle the exception, try and except block is used and executes the print statement to enter the correct integer value.

CREATE\_THE\_NOTE() function is created to create the notes object instance. In this unique id is created automatically, title, text, date, completed status, completion date will be created based on the user input values this is done by INPUT\_VALUES(notebook) function. This values will be append to Note\_list and saved in the text file using SAVEANDEXIT() function.

READ\_THE\_NOTE() function displays the note based on the user selected id.

UPDATE\_THE\_NOTE() function update all the attributes created in the Note class by using user selected id.

DELETE\_THE\_NOTE() deletes the selected id notes.

Count\_date() function counts the number of days it took for the selected id notes to complete.

CHANGE\_COMPDATE() function changes the completed date for the user selected note id. If user selects each menu option and perform the operation it should be saved using the Save and Exit menu. Try and except block created to handle error.

User could also able to check the number of notes created in a database using NO\_OF\_NOTES() function by calculating the list length . And also can see all the notes created using SCREEN\_ALL\_NOTES() function. User shall able to check the completed status and percentage of pending notes and total notes created using NOTES\_COMPLETED\_STATUS() function. Finally, menu function is called inside the main function.

C) Class Main:

Interface class is imported from Interface module. Main class is created and interface menu function is called inside the main method and main function is called. User should run the Main class first to execute the menu option in command shell.

**User Guide**:

To start the program run the main.py then program will import the data from the notes.txt file. The menu option will ask the user a option between 1 and 10.

The following options are displayed in the menu:

1) **CREATE THE NOTE**: Select menu 1. The program will ask the user to enter the title, text, completion status Y or y for completed else any other value as not completed and completion date will be printed if completion status is Y or y. Id will be automatically created after each note created. Once note is created save the note using Save and Exit menu the note created will save in the notes.txt document.

2) **READ THE NOTE**: Select menu 2 to read the notes from the text file by using the user input id.

3) **UPDATE THE NOTE**: Select menu 3 and enter the note id that need to be updated. Then user should update the all attributes and save the file

4) **DELETE THE NOTE**: Select menu 4 and enter the note id that need to be removed from the database.

5) **NO OF NOTES**: Select menu 5 to print the total number of notes created in the database.

6) **NOTES COMPLETED STATUS**: Select menu 6 to print the total number of notes created in the database, number of uncompleted notes and percentage of notes yet to complete.

7) **SCREEN ALL NOTES**: Select menu 7 to display all notes created in the data base.

8) **COUNT DATE**: Select menu 8 and enter the note id to find the number of days user took to complete the note.

9) **CHANGE COMPLETED DATE**: Select menu 9 and enter the id to change the note completed date for the selected note. (it can be created in the update menu itself but it is requested to create a separate menu option so this menu option is created).

10) **SAVE & EXIT**: Select menu 10 to save after making any changes in the database. Please select this option after creating, updating, and deleting note.