

Documentation for the Phonebook program

Description

The Phonebook program allows managing a list of contacts, which can be saved to a file or loaded from an existing file. This code is a command-line interface for managing a phonebook. It reads a JSON file containing phonebook entries, allowing users to perform CRUD operations on them.

The code uses a set of utility functions for input validation, print statements, and file operations.

The program provides the following functionalities:

Adding a new contact

Searching contacts by name or age

Finding an email address by name

Deleting a contact by name

Counting all contacts in the phone book

Sorting and displaying contacts by age

Updating ages of all contacts in the phone book

Calculating the average age of all contacts in the phone book

Functions

- `print_phonebook()`: Displays all contacts in the phonebook on the screen.
- `add_new_entry_to_phonebook()`: Asks the user for information about a new contact and adds it to the phonebook.
- `find_entry_by_name_in_phonebook()`: Asks for a name and displays the contact with that name on the screen if it exists in the phonebook.
- `find_entry_by_age_in_phonebook()`: Asks for an age and displays the contacts with that age on the screen if they exist in the phonebook.
- `find_email_by_name_in_phonebook()`: Asks for a name and displays the email of the contact with that name on the screen if it exists in the phonebook.
- `delete_entry_by_name_in_phonebook()`: Asks for a name and deletes the contact with that name from the phonebook.
- `count_all_entries_in_phonebook()`: Displays the total number of contacts in the phonebook.
- `print_phonebook_by_age()`: Displays all contacts in the phonebook, sorted by age.
- `update_contact_ages()`: Asks the user how many years to increase the age of each contact and updates them

in the phonebook.

- `calculate_average_age_of_all_persons_in_phonebook()`: Calculates the average age of all contacts in the phonebook and displays it on the screen.
- `save_phonebook_to_file()`: Saves the phonebook to a file.
- `load_from_file()`: Loads the phonebook from a file.

The functions are wrapped in the `print_start_end_process` decorator, which allows messages about the start and end of executing each function to be displayed on the screen.

Modules

The following modules are used to work with the program:

`copy` module from Python Standard Library: for working with object copies
`json` module from Python Standard Library: for working with json file
`format`
`argparse` module from Python Standard Library: for processing command-line arguments
`input` module from utilities package: for collecting user input data
`decorator` module from utilities package: for decorating functions
`print` module from utilities package: for outputting messages to the user
`save_file` module from utilities package: for saving the contact list to a file

Variables

`running`: a flag variable indicating the state of the program
`changes_made_to_phonebook`: a flag variable indicating whether changes have been made to the contact list
`file_name`: the name of the file with phone contacts
`phone_book`: the list of contacts

Command Line Parameters

The code has an `argparse` module that allows the user to pass a filename as an argument while running the code in the command-line interface. It also enables verbose mode using the `-v/--verbose` option.

`url` - The first argument, `url`, is a positional argument that specifies the path to the file that contains the phone book data. The `type` parameter specifies that this argument should be a string, and the `help` parameter provides a brief description of the argument for the user.

-v or --verbose - The second argument, is an optional argument that enables decorators when passed. The action parameter specifies that the argument should be a boolean flag, and the help parameter provides a brief description of the argument for the user.

How to run

The program can be run from the command line by navigating to the directory containing the script and typing the following command:

```
python phonebook.py <file_name> [-v]
```

where file_name is the name of the file to be used for storing contacts. The -v argument is optional and can be used to enable the display of process start and end messages.

Usage

After running the program, the user can choose from the following options:

1. Add a new contact: The user will be prompted to enter the surname, name, age, phone number, and email of the new contact. The new contact will be added to the phone book.
2. Search contacts by name: The user will be prompted to enter a name. The program will search the phone book for contacts with that name and display them.
3. Search contacts by age: The user will be prompted to enter an age. The program will search the phone book for contacts with that age and display them.
4. Find an email address by name: The user will be prompted to enter a name. The program will search the phone book for a contact with that name and display their email address.
5. Delete a contact by name: The user will be prompted to enter a name. The program will delete the contact with that name from the phone book.
6. Count all contacts in the phone book: The program will display the total number of contacts in the phone book.
7. Print the phone book sorted by age: The program will sort the contacts by age and display them.
8. Update ages of all contacts in the phone book: The user will be prompted to enter the number of

years to add to the ages of all contacts in the phone book. The program will update the ages and display a message.

9. Calculate the average age of all persons in the phone book: The program will calculate and display the average age of all contacts in the phone book.

The user can exit the program by typing exit or quit.