Le Duc - User Guide

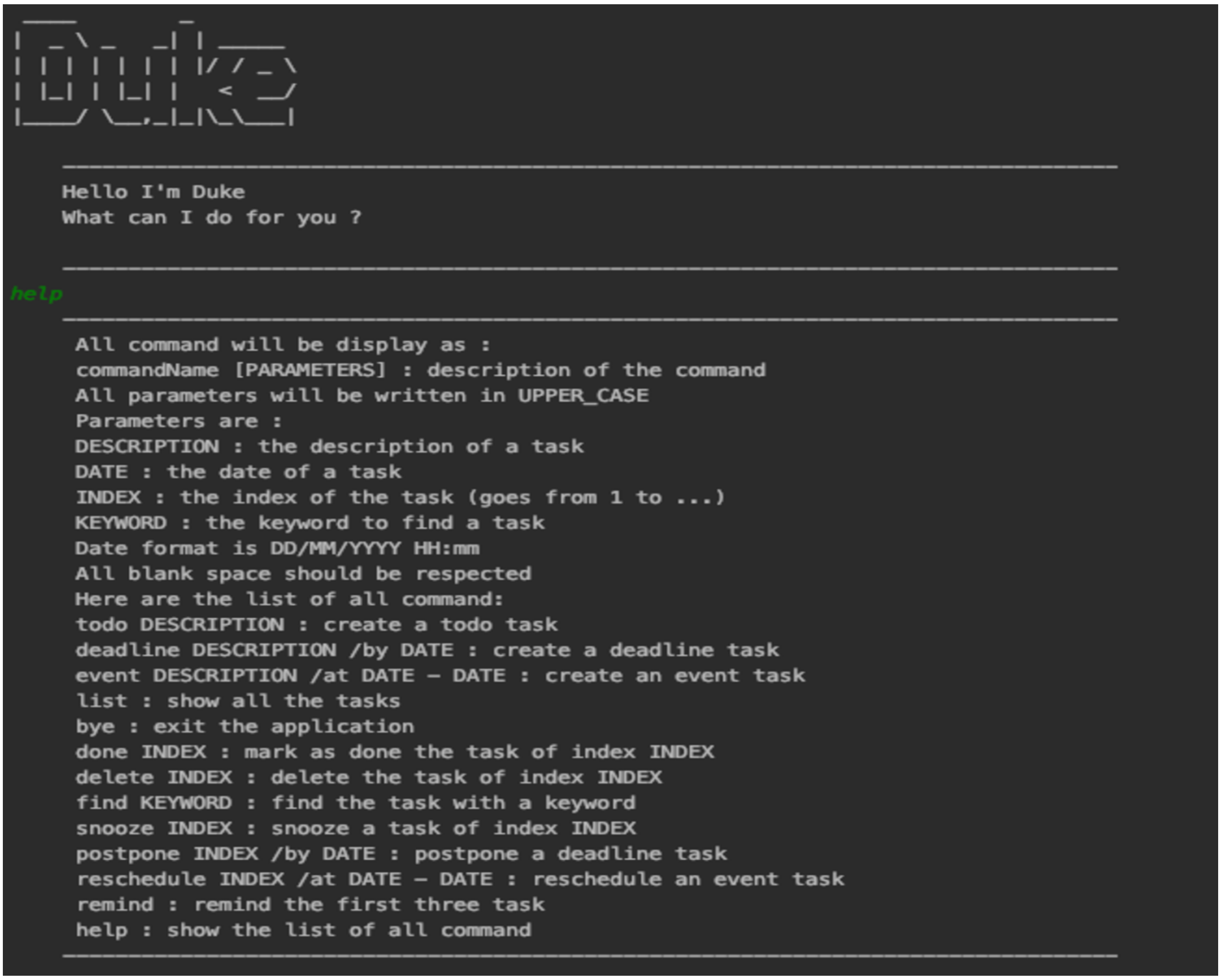
By: `[AY1920S1-CS2113-T16-1] Le Duc`

1. Introduction

Le Duc ("The Duke" in French) is for students who have both academic and extracurricular lives and have to manage them both. Outside, they have to go to school, work part-time, attend club and have fun at some party. At home, they have to do homework, undertake a project, do some chores, cook some nice food. Pretty busy, right? With this new tool, a student can manage and organise his time more efficiently.

2. Quick Start

1. Please ensure that you have Java 11 or above installed.
2. Download the latest leduc jar file.
3. Execute the jar file.
4. Type the command in the shell.
5. Please refer to Section 3, “Features” part for details of each command.



```
 ____        _
|  _ \ _   _| | _____
| | | | | | | |/ / _ \
| |_| | |_| |   <  __/
|____/ \__,_|_|\_\___|

    ____________________________________________________________
    Hello I'm Duke
    What can I do for you ?

    ____________________________________________________________
help
    ____________________________________________________________
     All command will be display as :
     commandName [PARAMETERS] : description of the command
     All parameters will be written in UPPER_CASE
     Parameters are :
     DESCRIPTION : the description of a task
     DATE : the date of a task
     INDEX : the index of the task (goes from 1 to ...)
     KEYWORD : the keyword to find a task
     Date format is DD/MM/YYYY HH:mm
     All blank space should be respected
     Here are the list of all command:
     todo DESCRIPTION : create a todo task
     deadline DESCRIPTION /by DATE : create a deadline task
     event DESCRIPTION /at DATE - DATE : create an event task
     list : show all the tasks
     bye : exit the application
     done INDEX : mark as done the task of index INDEX
     delete INDEX : delete the task of index INDEX
     find KEYWORD : find the task with a keyword
     snooze INDEX : snooze a task of index INDEX
     postpone INDEX /by DATE : postpone a deadline task
     reschedule INDEX /at DATE - DATE : reschedule an event task
     remind : remind the first three task
     help : show the list of all command
    ____________________________________________________________
```

3. Features

Command Format

- All whitespaces must be respected.
- Words in UPPER_CASE are parameters the users have to write:
    - DATE : The date format must be dd/MM/yyyy hh:mm where dd is the days, MM is the months, yyyy is the years, hh is the hours and mm is the minutes.
    - DESCRIPTION: The description of the task
    - WELCOME: the welcome message
    - INDEX is the index of the task (goes from 1 to the number of tasks). You can find the index of the task by showing all the task with the command list of by searching the task with the command find.
    - KEYWORD: the keyword to find a task.
    - SORTTYPE: date, description, priority, type or done

3.1. Viewing all the command: `help`

To show the list of all command : `help`

Example:

- `help`

3.2. Adding a todo task: `todo`

To add a todo task : `todo DESCRIPTION` An option could be added to give a priority to the task : `todo DESCRIPTION prio INDEX`

Be careful:

- The second INDEX cannot be less than 0 or greater than 9.
- 0 is the less urgent, 9 is the most urgent

Examples:

- `todo read book`
- `todo buy book`
- `todo sell book prio 3`

3.3. Adding a homework task: `homework`

To add a homework task : `homework DESCRIPTION /by DATE` An option could be added to give a priority to the task : `homework DESCRIPTION /by DATE prio INDEX`

Be careful:

- The second INDEX can’t be less than 0 or greater than 9.
- 0 is the less urgent, 9 is the most urgent

Example:

- `homework CS4239: lab3 /by 21/09/2019 23:59`
- `homework CS4239: lab4 /by 25/09/2019 23:59 prio 7`

3.4. Adding a event task: `event`

To add an event task : `event DESCRIPTION /at DATE - DATE` An option could be add to give a priority to the task : `event DESCRIPTION /at DATE - DATE prio INDEX`

Be careful:

- The second INDEX can’t be less than 0 or greater than 9.
- 0 is the less urgent, 9 is the most urgent

Example:

- `event CS4211: mid-term /at 07/10/2019 09:00 - 07/10/2019 11:00`
- `event CS4211: final exam /at 07/12/2019 09:00 - 07/12/2019 11:00 prio 8`

Be careful : when creating an event, two dates can’t clash

3.5. Find the task by relevancy : `find`

To find a task by character relevancy : `find QUERY`

The find command allows the user to search for tasks via character matching (NOT keyword matching). The command will return tasks ordered by ascending relevancy.

Typos in the user query will not affect performance.

Examples:

Partial word query

```
find sci

--------------------------------------------------------------------------------
         Here are the most relevant tasks in your list:
         7. [H][▯] science by: 05/05/2005 05:05 [Priority: 2]
         8. [H][▯] test by: 01/01/2001 01:01 [Priority: 5]
         6. [H][▯] homeworktest by: 04/04/2004 04:04 [Priority: 5]


--------------------------------------------------------------------------------
```

Typo in Query

```
find homewqeuktest

----------------------------------------------------------------------------------
         Here are the most relevant tasks in your list:
         6. [H][□] homeworktest by: 04/04/2004 04:04 [Priority: 5]
         5. [H][□] math by: 11/11/2011 01:01 [Priority: 5]
         7. [H][□] science by: 05/05/2005 05:05 [Priority: 2]
         8. [H][□] test by: 01/01/2001 01:01 [Priority: 5]


----------------------------------------------------------------------------------
```

If all tasks have no matching characters with query

```

----------------------------------------------------------------------------------
         There is no matching tasks in your list

----------------------------------------------------------------------------------
```

3.6. Delete a task : `delete`

To delete a task : `delete INDEX`

Example:

- `delete 1`

3.7. Show all the tasks: `list`

To show all the tasks: `list`

Example:

- `list`

3.8. Mark a task as done : `done`

To mark a task as done: `done INDEX`

Example:

- `done 2`

3.9. Snooze a homework task : `snooze`

To snooze a homework task: `snooze INDEX` The snooze time is fixed at 30 minutes.

Example:

- `snooze 2`

3.10. Postpone a homework task : `postpone`

To postpone a homework task: `postpone INDEX /by DATE`.

DATE is the new date of the homework task. The new date should be after the old one.

Example:

- `postpone 2 /by 12/12/2020 22:10`

3.11. Reschedule an event task : `reschedule`

To reschedule an event task: `reschedule INDEX /at DATE - DATE`.

Be careful : when rescheduling an event, two dates can’t clash

Example:

- `reschedule 3 /at 12/12/2020 10:00 - 12/12/2020 12:00`

3.12. Reminder for upcoming tasks : `remind`

Reminds the user of the 3 most upcoming tasks : `remind`

Example:

- `remind`

```
remind
         1. [H][□] test by: 01/01/2001 01:01 [Priority: 5]
         4. [H][□] math by: 11/11/2011 01:01 [Priority: 5]
         5. [E][□] e at: 21/09/2019 00:00 - 28/10/2019 22:22 [Priority: 5]
```

3.13. Edit a task : `edit`

- Multi-steps command: to edit a task, follow these instructions:
  1. `edit`
  2. All of the tasks will be displayed, you have to choose a task INDEX

    3. Depending on the type of task:
        - If it is a todo task, you have to enter the new DESCRIPTION
        - If it is not a todo task, you have to choose 1) if you want to edit the description or 2) if you want to edit the date
            - Then, enter the new DESCRIPTION or the new DATE of the task

- For one shot command:
    - edit the description: `edit INDEX description DESCRIPTION`
    - edit the date of an homework task: `edit INDEX /by DATE`
    - edit the period of an event task: `edit INDEX /at DATE - DATE`

Examples:

- Edit the description of the task 2 (todo task)
    - Multi-steps command
        - `edit`
        - `2`
        - `DESCRIPTION`
    - One shot command
        - `edit 2 description DESCRIPTION`
- The task 2 is a homework task:
    - Multi-steps command
        - `edit`
        - `2`
        - `2`
        - `DATE`
    - One shot command
        - `edit 2 /by DATE`
- The task 3 is a event task:
    - Multi-steps command
        - `edit`
        - `3`
        - `2`
        - `DATE - DATE`
    - One shot command
        - `edit 3 /at DATE - DATE`

3.14. Customize the welcome message : `setwelcome`

To customize the welcome message: `setwelcome WELCOME`

Example:

- setwelcome `hello World`

Be careful:

*Reverting to the previous welcome message is not possible once a new message is set. *Ensure the directory data is in the correct location

3.15. Exit the program : `bye`

To exit the program : `bye`

Example :

- `bye`

3.16. Get some statistics: `stats`

Display statistics : `stats`

Display statistics about your tasklist.

Enter command `stats` to view general statistics, `stats -p` to view detailed priority statistics, or `stats -c` to view detailed completion statistics.

Example:

- `stats`

General Statistics Example:

```
stats

----------------------------------------------------------------------------------
Here are some general statistics about your task list:
Number of tasks: 8.0
Number of Todo's : 3
Number of Events: 1
Number of Homeworks: 4
Number of Uncompleted Tasks: 5
Number of Completed Tasks: 3
Percent Complete: 37.5%

----------------------------------------------------------------------------------
```

- `stats -c`

Completion Statistics Example

```
stats -c

_________________________________________________________________________________
Here are some completion statistics about your task list:
----COMPLETION COUNTS----
Number of incomplete Homeworks remaining: 2
Number of incomplete Todos remaining: 2
Number of incomplete Events  remaining: 1
----COMPLETION PERCENTAGES----
Percent of incomplete Homework: 50.0%
Percent of incomplete Todo: 66.66666666666666%
Percent of incomplete Events: 100.0%

_________________________________________________________________________________
```

- `stats -p`

Priority Statistics Example:

```
stats -p

_________________________________________________________________________________
Here are some priority statistics about your task list:
----PRIORITY COUNTS----
Number of tasks with priority 5: 7
Number of tasks with priority 4: 0
Number of tasks with priority 3: 0
Number of tasks with priority 2: 1
Number of tasks with priority 1: 0
----PRIORITY PERCENTAGES----
Percent of tasks with priority 5: 87.5%
Percent of tasks with priority 4: 0.0%
Percent of tasks with priority 3: 0.0%
Percent of tasks with priority 2: 12.5%
Percent of tasks with priority 1: 0.0%

_________________________________________________________________________________
```

3.17. Sort by: `sort`

Sort all task by date/description/priority: `sort SORTTYPE`

SORTTYPE is either date, description, priority

Be careful:

- Sorting by date will sort tasks in chronological order
- Sorting by description will sort the descriptions in alphabetical order
- Sorting by priority will sort tasks in ascending urgency
- Sorting by type will sort tasks depending on its task type (event, homework, todo)
- Sorting by done will sort tasks depending on it the task is done or not

Examples:

- `sort date`
- `sort priority`
- `sort description`
- `sort type`
- `sort done`

3.18. Prioritize: `prioritize`

Giving priority to task: `prioritize INDEX prio INDEX`

The first INDEX is the task index

The second INDEX is the priority (goes from 0 to 9)

Be careful:

- The second INDEX can't be less than 0 nor greater than 9.
- 0 is the less urgent, 9 is the most urgent
- When creating a task, specifying the priority is optional. When the priority is not specified, the task will automatically have a priority of 5.

Example:

- `prioritize 4 prio 2`
- `prioritize 5 prio 8`

3.19. Display by day/week/month: `show`

Display all the task for one particular day/week/month: `show DATETYPE DATE`

DATETYPE is day, dayofweek, today, week, month, year.

The DATE argument depends on the DATETYPE

- day :

    - shows all the tasks for the given date
    - DATE : DD/MM/YYYY
- dayofweek :
    - shows all the tasks for the given day of week
    - DATE : monday, tuesday, wednesday, thursday, friday, saturday, sunday
- today :
    - shows all the tasks for the user’s today
    - DATE : nothing should be written !!
- week :
    - shows all the tasks, starting from user’s today to 7 days later (the last days is not included)
    - DATE : nothing should be written !!
- month :
    - shows all the tasks for the given month
    - DATE : MM/YYYY
- year
    - shows all the tasks for the given year
    - DATE : YYYY

There are two behaviour, one line command and multi-step command

Example (one line command) :

- `show day 29/10/2019`
- `show dayofweek monday`
- `show today`
- `show week`
- `show month 10/2019`
- `show year 2019`

Example (multi-step command) :

- `show day`
- The console ask to enter the day :
- `29/10/2019`

or

- `show dayofweek`
- The program ask to enter the day :
- `monday`

3.20. Shortcut : `shortcut`

Give shortcut to command : `shortcut COMMANDNAME SHORTCUTNAME`

COMMANDNAME is the name of the command (like todo, sort, show ...)

SHORTCUTNAME is the new shortcut name for the command

There are three behaviour : One line command, multi-step command, multi-step customize all command

Be careful :

- There can't be 2 same shortcut name.
- The shortcut name can't be the same as one of the default command name (for example, the shortcut name can't be todo because it is a default command name).

Example (One line command) :

- `shortcut todo t`
- `shortcut prioritize prio`

Example (multi-step command) :

- `shortcut todo`
- The program ask to enter the shortcut
- `t`

Example (multi-step customize all command)

- `shortcut`
- The program enter in customize shortcut mode
- The program display the first command with his shortcut and ask to enter for a new shortcut name
- `shortcutname`
- The program display the first command with his shortcut and ask to enter for a new shortcut name
- `shortcutname2`
- ...
- The program display all the shortcut

Be careful :

- There can't be 2 same shortcut name. If we enter a shortcut that already exists, we are in one line command or multi-step command behaviour, it will show an error, and if we are in multi-step customize all command, it will ask again.

- The shortcut name can't be the same as one of the default command name (for example, the shortcut name can't be todo because it is a default command name).

For each command, the console will output the command name.

The user (you) have to input the shortcut you want.

If the shortcut already exists, the console will tell you to assign another shortcut

3.21. View Unfinished tasks : `unfinished`

Find and display all unfinished tasks : `unfinished`

Example:

- `unfinished`
- Output:

```
Here are the unfinished tasks in your list:
        1. [T][□] td1 [Priority: 5]
        2. [H][□] h1 by: 14/09/2019 22:33 [Priority: 5]
        3. [E][□] e1 at: 21/09/2019 00:00 - 28/10/2019 22:22 [Priority: 5]
        4. [T][□] td4 [Priority: 5]
        5. [T][□] td5 [Priority: 5]

--------------------------------------------------------------------------------
```

3.22. Change language : `language`

Change the language for all the display message : `language LANGUAGE`

`LANGUAGE` can be en or fr (only two language is available). The language will be set after the program is closed and open again. Example :

- `language fr`
- `language en`

3.23. Create task list (not implemented yet)

Create a different task list: `tasklist DESCRIPTION`

DESCRIPTION is the name of the new tasklist

Be careful:

- The first tasklist will be name main task list, except if it is edited (can't be done for the moment)
- Each new tasklist will have an index automatically

3.24. Display different task list (not implemented yet)

Display different task list, for example one task list for school, one task list for work: `display tasklist INDEX`.

INDEX is the task list index (if exist)

3.25. Import other task list (not implemented yet)

If the user want to import another task list: `Import FILENAME`.

The FILENAME is path to the file

Be careful:

- The file must be in the same format as every file in the project
- The file must be in the directory data

3.26. Subtask (not implemented yet)

You can create a subtask of another task: when creating the task, just add `sub INDEX` at the end.

INDEX is the index of the task

4. FAQ

Q) Will there be more feature ?

A) Yes, if we have enough imagination and time and energy.

5. Command Summary

- **help** `help`
- **todo** : `todo DESCRIPTION`
- **homework** : `homework DESCRIPTION /by DATE`
- **event** : `event DESCRIPTION /at DATE - DATE`
- **find** : `find KEYWORD`
- **delete** : `delete INDEX`
- **list** : `list`
- **done** : `done INDEX`
- **snooze** : `snooze INDEX`
- **postpone** : `postpone INDEX /by DATE`
- **reschedule** : `reschedule INDEX /at DATE - DATE`

- **remind** : `remind`
- **edit** : `edit` and follow the instructions or `edit INDEX description DESCRIPTION` or `edit INDEX /by DATE` or `edit INDEX /at DATE - DATE`
- **setwelcome** : `setwelcome WELCOME`
- **bye** : `bye`
- **stats** : `stats`
- **sort** : `sort SORTTYPE`
- **prioritize** : `prioritize INDEX prio INDEX`
- **display** : `display DATETYPE DATE`
- **shortcut** : `shortcut COMMANDNAME SHORTCUTNAME`
- **unfinished** : `unfinished`
- **language** : `language LANGUAGE`

Not implemented yet

- **create a tasklist** : `tasklist DESCRIPTION`
- **display tasklist** : `display tasklist INDEX`
- **import file** : `Import FILENAME`
- **subtask** : when creating the task, just add `sub INDEX` at the end