TIC2002 Introduction to Software Engineering AY2021/22 Semester 1

Project report

Name: Chang Meng Yeong Brandon

Student Number: A0194283X

GitHub Username: https://github.com/Brandoncmy/duke

User stories Give some user stories (5-10) that match your project.

This project is building a "Reminder" application for the user.
The user is able upload new task to the application including setting date and time for the event or deadline.
The task can have a status of completion or not.
Lastly, the user is able to view the list of task uploaded to the application.

NFRs Give some non-functional requirements (3-5) for your project.

All the tasks will be saved in a document list as long as it is entered by the user.

The application is able to capture the capitalize of small letter of the wording. So it will not go into error message.

Level-1 Show the output Duke shows when you launch the program.

Under Duke.java

```
"C:\Program Files\Java\jdk-11.0.12\bin\java.exe" "-javaagent:C:\Program Files\JetB
    ----------------------------------------------------------
    Hello! I'm

     ----        _
    |  _ \ _   _| | _____
    | | | | | | | |/ / _ \
    | |_| | |_| |   < __/
    |____/ \__,_|_|\_\___|
    What can I do for you?
    Here are the list of menu:
     1. todo ...
     2. deadline ... by ...
     3. event ... at ...
     4. list = to show the item in the list
     5. done ...(number)
     6. delete ...(number)

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

Level-4 Give examples of command(s) and expected outputs for adding different types of tasks.

Under BeforeOOP.java

```
deadline return book /by: Sunday

--------------------------------------------------
Got it. I've added this task:
    [D][ ]  return book (by:: Sunday)
Now you have 2 tasks in the list.

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

Level-2 Give examples of command(s) and expected outputs for listing tasks.

Under BeforeOOP.java

```
list

--------------------------------------------------
Here are the tasks in your list:
1.[T][ ]  return book
2.[D][X]  return book (by:: Sunday)
3.[E][ ]  go to class (at:: 6pm)

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

Level-3 Give examples of command(s) and expected outputs for marking/unmarking tasks as done.

Under BeforeOOP.java

```
done 3

_____
Nice! I've marked this task as done:
[X]  go to class (at:: 6pm)

_____


list

_____
Here are the tasks in your list:
1.[T][ ]  return book
2.[D][X]  return book (by:: Sunday)
3.[E][X]  go to class (at:: 6pm)
4.[T][ ]  study exam for TIC2002

_____
```

Level-5 Describe what kind of errors Duke can handle. E.g., give different types of incorrect commands (and the expected outputs) it can handle

Basically, Duke can only recognize keyword like "todo", "deadline", "event" , "list" , "done" , "delete"

Todo error:

```
todo

_____
☹ OOPS!!! The description of a todo cannot be empty.

_____
```

Deadline error:

```
deadline

_____
☹ OOPS!!! The description of a deadline cannot be empty.

_____
```

Event error:

```
event

_____
☹ OOPS!!! The description of an event cannot be empty.

_____
```

Done error:

```
list

----------------------------------------------
Here are the tasks in your list:
1.[T][ ]  eat
2.[T][ ]  sleep
3.[T][ ]  study
4.[T][ ]  work
----------------------------------------------

done 5
Exception in thread "main" java.lang.IndexOutOfBoundsException Create breakpoint : Index 4 out of bounds for length 4 <3 internal lines>
    at java.base/java.util.Objects.checkIndex(Objects.java:372)
    at java.base/java.util.ArrayList.get(ArrayList.java:459)
    at duke.BeforeOOP.main(BeforeOOP.java:82)
```

Level-6 Give examples of command(s) and expected outputs for deleting tasks.

```
list

----------------------------------------------
Here are the tasks in your list:
1.[T][ ]  study TIC2002
2.[E][ ]  go to bed (at: 10pm)
3.[D][ ]  submit assignment (by: Sunday)

----------------------------------------------


delete 2

----------------------------------------------
Noted. I've removed this task:
    E[ ]  go to bed (at: 10pm)
Now you have 2 tasks in the list.

----------------------------------------------


list

----------------------------------------------
Here are the tasks in your list:
1.[T][ ]  study TIC2002
2.[D][ ]  submit assignment (by: Sunday)

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

Level-7 Give a sample of the tasks as they are stored in the hard disk.

```
   duke.txt - Notepad

 File  Edit  Format  View  Help
T | false |   study TIC2002
D | false |   submit assignment (by: Sunday)



e.g.,
T | 1 | read book
D | 0 | return book | June 6th
E | 0 | project meeting | Aug 6th 2-4pm
T | 1 | join sports club
```

Level-8 Explain how Duke uses dates/times i.e., what commands can use dates/times in meaningful ways?

Java has an inbuilt library (java.time.LocalDateTime).
Duke can this this function to combines data and time, but without any time-zone.

Level-9 Give examples of command(s) and expected outputs for searching for tasks.

My OOP is not working after several attempt and unable to debug…

So I will refer to "beforeOOP.java" to describe this function. Thus, user will need to input "list" to search the task.

Level-10/**individual feature** If you implemented a GUI, give some screenshots. If you implemented an individual feature, describe that feature.

No implemented.

Other features Describe other features you implement (i.e., not described above), if any e.g., optional increments.

Not available.

A-MoreOOP Give a class diagram to match your code.

Give at least one object diagram illustrating the state of your program at some point. It should include at least one object of each of the important classes in your project.

Give at least one sequence diagram illustrating an object interaction in your product.

A-JavaDoc: Give at least 2 javadoc comments from you code.

```
/**
 * Creates Duke with specified filepath and initialised with storage, taskList and ui.
 * @param filePath The file path of the storage file.
 */
```

```
/**
 * Runs Duke and show output .
 */
```

```
/**
 *   The main function is to run Duke.
 * @param    args       The argument values provide by the user to run Duke.
 */
```

A-JUnit: Give 2-3 JUnit test methods from your code.

```
ke.java ×    🅖 DukeTest.java ×    🅒 Deadline.java ×    🅒 AddEventCommand.java ×    🅒 Ui.java

   package duke;


   import org.junit.jupiter.api.Test;


   import static org.junit.jupiter.api.Assertions.assertEquals;
       💡

   |

   public class DukeTest {
       @Test
       public void dummyTest() { assertEquals( expected: 2,    actual: 2); }
   }
```

A-Assertions: Give at least 2 code segments that contain assertions you added to your code.
Suggested test commands Give a list of commands a tester can execute in sequence to examine your product. Cover all features in a reasonable order. E.g., list todo borrow book deadline return book /by Sunday done 2