Jacob Madsen

Programming Journal

11/05/2019

Created BitBucket repository

Started Employee Clock project and linked it to git Repository

Added an employee class with the parameters: employeeName, employeeID and hourlyPay

Customized a GUI swing interface with the app name, a textfield for input and a submit button for clockin\clockout

Implemented functionality for clockin button where when clicked if the inputed ID is valid a clockin message displayed if invalid an error message is Displayed

11/06/2019

Added a clock class with a LocalDateTime, day of the week, and the employee clocking in\out

Added more functionality to the submit button where the employee can clock in or out

When the employee clocks in\out, their information and clock information is now saved to a txt file called ClockTimes.txt

11/07/2019

Added functionality where the clockin information is loaded on startup

The program currently can’t determine if the employee has already clocked in based on the txt file.

11/08/2019

Fixed error where the date and time fail to parse from strings to DateTime

Fixed bug where employee can only clockin once

Added functionality where the previous clockins\outs are loaded and if the employee clocked in then they can clock out

Changed GUI layout to have a clockin button and a clockout button

11/12/2019

Fixed bug where on startup, only the last line of the file is added to both clock in and out due to having the previous line overwritten when the next line is assigned to the clock.

Began to add a Boolean method to handle the conditions for if the employee clocked in.

11/13/2019

Fixed conditionals to determine if clocked in or out

Finished Boolean method to return true or false for if the employee clocked in

Changed the GUI to have one button for clockins\outs

Created Flowchart to help map out the conditionals for clockins\outs

11/14/2019

Added enum called ClockType with IN or OUT

If even the enum is set to IN else OUT

Changed condtionals to check for if the enum is IN

Added parameter called lastNonNull to hold the last clock in\out

Added a parameter in clock for the clock type with the appropriate getter and setter methods

Notes and restrictions:

* Del IV took longer than the others due to having to revert to previous commits to rework the conditionals for clocking in\out. Some major roadblocks were to refactor the conditionals to be contained inside a Boolean method and making use of an enum, and a second clock to determine if the most recent clock time is a clock in or out

BitBucket: <https://bitbucket.org/Jacob_Madsen/employeeclock/commits/968415b1b3f724e508c2b2b215b7317bfd2cc939>

Trello: <https://trello.com/b/t2kHi2vl/employeeclock>