

Changes To the Original Design

I made a variety of changes in regards to adding more functions to make the readability of my code easier. For example, if there was to be a process to display only the events in a particular month, I added a function that would handle that specific situation and so forth (displayYearly(), displayMonthly(), displayWeekly(), displayDaily()).

The primary change that I made was changing the way in which the software keeps track of the Date. The original design had a Month.java class that stored specific fields such as: name, maxDays, etc. However, being that there is a very useful java library called “java.util.Date” and other libraries, I tried to maneuver a way to interchange the design between the original and my new incorporation of the Date library. I ended up taking the original implementation (Month.java) and adding additional functions that would initiate Date objects based on the original String containing the startDate. The Month.java class is fully implemented, but the underlying work of the Timer.java class is driven by the use of the Date library. I decided this change would be necessary being that it is much more efficient to have an already existing library with working methods that get the job done easily and clearly. Moreover, it allowed for a more accurate implementation of a Timer for events.

The other minimal changes to the design that I made were primarily dealing with the addition of multiple functions in the main class, that allow the program to provide the user with instructions for input and limit the user to only being able to perform a certain function, one function at a time.

The features that I did not fully implement were the following: multiple users, adding the ability to repeat and event weekly, allowing users to view shared events, and the calendar being able to show only a weekly view of events. Time did not allow re-factoring existing code so that I could add multiple users to the program. However, it is a solution that with more time, could be added efficiently and easily. As for the further implementations that weren't fully finished, the same issue was present: lack of time. The features that are left to be implemented are inherently part of the existing design and must only be tweaked slightly to allow for a completely functioning system.