Mengenlehreuhr Clock Format Conversion

CAIT Web Developer Internship Test | Mayur Kabra

Executing & Testing

- The deliverable contains a compiled executable file by the name 'Mengenlehreuhr.exe'
- On executing this file, a cmd console should open up giving two options
    - Option 'a' corresponds to the first part of the given requirements in which a format is specified and an input in that format is expected.
        - On entering a valid time, an equivalent output of the Mengenlehreuhr clock format is displayed
        - Incorrect time entry will give handled exception for you to try again.
    - Option 'b' will demonstrate conversion from the in-built System.DateTime class
        - This example will use the current time for the purpose of example.
        - Expected output is the Mengenlehreuhr clock format equivalent of the current time

Code & Customizing

- The deliverable also contains the source code of the application which has been created in a modular pattern with the following features:
    - Any measure can be broken in 2 rows. Which, here, are referred to as 'Primary' & 'Secondary' rows.
      Eg: - The first row in clock denotes *primary row* for hour with unit of each lamp as 5 hours and the row following it, the *secondary row* with unit of each lamp as 1 hour
    - Any row can have *checkpoints* enabled which may have a different color of lamp
      Eg: - The *primary row* of minutes has a *checkpoint* enabled. This has every 3rd lamp of that row as a "R" unlike the rest being "Y"
- This gives the following customizing capabilities:
    - Lamp color of individual row
    - Unit of every *primary row*. This unit also implicitly defines the maximum amount of the measure that its *secondary row* will have.
      Eg: Primary row of hour has 5 hours as unit. So implicitly, the secondary row will have 4 lamps of 1 hour each
    - Ability to add a checkpoint lamp to any of the rows, with the repetitive value and its color being customizable
- Input Options
    - The function *ConvertTime(int hours, int minutes, int seconds)* takes the mentioned parameters as input and provides with the converted output
    - Now, as in the case of *ConvertTime(DateTime dateTime)*, if we want to add to the program functionality of being able to use any other object, we will have to create a similar function, which will extract hours, minutes and seconds from the given object, and pass it to the generic function. The function will then return the string it gets from the generic function.