

11.0-Introduction to Enumerations

Program

using System;

// Enumeration for Light Modes

public enum LightMode

{

Day,

Night

}

// Street Light Control Class

public class StreetLight

{

private LightMode currentMode;

public StreetLight()

{

// Initialize with Day as the default mode

currentMode = LightMode.Day;

}

// Method to toggle the light mode between Day and Night

public void ToggleMode()

{

currentMode = (currentMode == LightMode.Day) ? LightMode.Night : LightMode.Day;

```
}

// Method to display the current light mode

public void DisplayMode()

{
    Console.WriteLine($"Street Light is currently in {currentMode} mode.");
}

}
```

```
// Main Program

class Program

{
    static void Main()

    {
        StreetLight streetLight = new StreetLight();

        // Simulate mode changes for demonstration purposes

        for (int i = 0; i < 5; i++)
        {
            streetLight.DisplayMode();

            Console.WriteLine("Press Enter to toggle the light mode...");

            Console.ReadLine();

            streetLight.ToggleMode();
        }
    }
}
```

Exercise

Write a C# program that simulate a street light control system that toggles between day and night modes.

The program defines an enum named `LightMode` with two values: `Day` and `Night`.

It also defines a class named `StreetLight` which represents the street light control system.

StreetLight Class:

It has a private attribute `currentMode` of type `LightMode` to store the current mode of the street light.

Constructor:

The constructor initializes the `currentMode` attribute with the default value `Day`.

This method toggles the light mode between `Day` and `Night`. If the current mode is `Day`, it switches to `Night`, and vice versa.

This method displays the current mode of the street light.

Main Program:

The Main method creates an instance of the `StreetLight` class.

It then simulates mode changes for demonstration purposes using a loop.

In each iteration of the loop:

It displays the current mode of the street light.

Waits for the user to press Enter.

Toggles the mode of the street light.

During execution, the program simulates changes in the street light mode.

It displays the current mode and waits for the user to toggle the mode by pressing Enter.

This process repeats for a predefined number of iterations (5 in this case) to demonstrate the mode toggle functionality.

Output:

The program outputs the current mode of the street light system, indicating whether it's in "Day" or "Night" mode.

It prompts the user to press Enter to toggle the mode.

The user interacts with the program by observing the mode changes displayed on the console and pressing Enter to toggle the mode.

The program terminates after completing the predefined number of mode toggles, allowing the user to observe the simulation results.

Hint

It has a private attribute `currentMode` to store the current mode of the street light.

The constructor initializes the `currentMode` attribute with the default value `Day`.

The `ToggleMode` method toggles the light mode between `Day` and `Night`.

The `DisplayMode` method displays the current mode of the street light.

It creates an instance of the `StreetLight` class.

It simulates mode changes for demonstration purposes using a loop.

In each iteration of the loop:

It displays the current mode of the street light.

It prompts the user to press Enter to toggle the mode.

It toggles the mode of the street light based on user input.

Explanation

This program simulates the control of a street light system, where the light can be toggled between two modes: "Day" and "Night".

In the `StreetLight` class, an enumeration named `LightMode` is defined with two values: `Day` and `Night`, representing the possible modes of the street light. The class itself maintains a private attribute `currentMode` to store the current mode of the street light, initialized to `Day` by default in the constructor.

The `ToggleMode` method is responsible for changing the light mode between `Day` and `Night`. It utilizes a conditional ternary operator to toggle the mode: if the current mode is `Day`, it switches to

Night, and vice versa.

The `DisplayMode` method simply outputs the current mode of the street light to the console using `Console.WriteLine`, providing feedback to the user about the current state of the light.

In the `Main` method of the `Program` class, an instance of `StreetLight` is created. It then enters a loop that iterates five times, each time displaying the current mode of the street light using `DisplayMode`. After each iteration, the program prompts the user to press `Enter` to toggle the light mode. Upon receiving input, it toggles the mode using `ToggleMode`. This loop simulates mode changes for demonstration purposes.