World Clock Documentation

How it works

The program starts by prompting the user to enter a list of cities separated by a comma.

The user's input is stored in a list and split into separate city names.

The program then defines TimeZoneInfo objects for different cities and retrieves their time zone information using the FindSystemTimeZoneById method.

A while loop is then entered that continues to run until the user presses the ESC key.

The loop first clears the console and then iterates through the cities list.

For each city, the program compares the lowercase version of the city name with predefined city names and displays the corresponding time using the ConvertTimeFromUtc method.

If the user's input does not match any of the predefined city names, the program will display a message indicating that either the city name is not in the list or there is a spelling mistake.

The loop then waits for 1 second using the Thread.Sleep method and repeats itself to show the most recent time.

Supported Cities

The following cities are supported by the World Clock program:

|  |  |
| --- | --- |
| Lahore | Pakistan Standard Time |
| Paris | Romance Standard Time |
| London | GMT Standard Time |
| New York | Eastern Standard Time |
| Perth | W. Australia Standard Time |
| Santiago | Central America Standard Time |
| Salt Lake City | Mountain Standard Time |
| Copenhagen | Central European Standard Time |

Note: The city names are case-insensitive, but the spelling should match the predefined city names.

Conclusion

The World Clock program is a simple but useful tool for quickly checking the current time in different cities. With its user-friendly interface and continuously updating display, it can be a valuable resource for individuals or organizations that need to keep track of time across multiple time zones.