CODING EXERCISE

.NET test exercise

Context

You are working for a betting company that handles large volumes of real-time data. The betting company is developing a new system to process incoming bets and determine their results. The system must be capable of generating a sample data and processing with highest performance and ensuring data integrity.

Objective

Develop a .NET Core system that meets the following requirements:

1. **Data Input:**
    - Bets arrive continuously and must be processed in the order they are received. This sample date will be provided by the service itself. This data set could contain repeated Unique Bet Id with different statuses, representing a streaming of data.
    - Each bet is represented by a `Bet` object containing the following fields:
        - `int Id` (Unique ID of the bet)
        - `double Amount` (Amount bet)
        - `double Odds` (Odds of the bet)
        - `string Client` (Client placing the bet)
        - `string Event` (Event being bet on)
        - `string Market` (Market of the event)
        - `string Selection` (Selection associated with the bet)
        - `BetStatus Status` (Status of the bet: OPEN, WINNER, LOSER, VOID)

2. **Bet Processing:**
    - Bets should be processed by a set of workers to reach highest performance.
    - Processing bet will take 50ms, it will be mocked using Thread Sleep, Task.Delay or similar.
    - Each worker takes a bet, ensures that it transitions through the correct sequence of statuses (Open → Won/Lost/Void), calculates the profit or loss based on the bet status and odds, and stores the result.
        - Won: Bet will count as part of profits

        - Lost: Bet will count as part of looses
        - Void: Bet will be refunded to customer with original stake amount
    - If a bet arrives with a wrong status sequence, it should be marked for review.
    - Use a variable to count the total number of bets processed.

3. **System Shutdown:**
    - The system should be able to shut down gracefully, ensuring that all bets being processed are not discarded and displaying summary in that moment.

4. **Data Summary:**
    - Implement a method public string GetSummary() to provide a summary of the processed data. This summary should include:
        - The total number of bets processed.
        - The total amount bet.
        - The total profit or loss (calculated as the sum of all individual bet results).
        - The top 5 clients with the highest total profit.
        - The top 5 clients with the highest total losses.

Additional Requirements

- **Simulated User Interface:**
    - Implement a method public void AddBet(Bet bet) to simulate the arrival of new bets throught an API REST.
    - Implement a method public void ShutdownSystem() to initiate the graceful shutdown process throught an API REST.
    - Implement an initial data set with 100 bets to ensure System is properly working.
- **Testing**
    - Software must have a >95% test coverage