

There is a method `SendMessageAsync` in class `BusMessageWriter` for sending messages to Message Bus with bufferization, it sends messages when the buffer size reaches a threshold.

The code is not thread safe. Modify the code to make it thread safe. How to modify the code to make it better in a clean code way? You can change the code as you want, the main thing is that there must be the method `SendMessageAsync`, and messages must be sent to Message Broker with bufferization.

```
public class BusMessageWriter
{
    private readonly IBusConnection _connection;
    private readonly MemoryStream _buffer = new();

    // how to make this method thread safe?
    public async Task SendMessageAsync(byte[] nextMessage)
    {
        _buffer.Write(nextMessage, 0, nextMessage.Length);
        if (_buffer.Length > 1000)
        {
            await _connection.PublishAsync(_buffer.ToArray());
            _buffer.SetLength(0);
        }
    }
}
```

Detailed answers would be a plus:

1. What issue do asynchronous methods help to address? What is the difference between asynchronous and parallel execution?
2. Are there any nuances or issues arising from the use of `ThreadPool`, `TPL` and `async/await` that you can describe based on your experience?
3. Which approaches do you know to signal about an occurred event to another thread? Can you elaborate on their differences and specifics?
4. Can you describe your experience with applying common design and development principles and patterns? Which of them do you find most useful? Which tasks did you solve by following them and why did you choose them?
5. Can you list the mechanisms used to implement Dependency Injection and compare them in detail, with some examples from your own experience?