```
using System;
using System.Collections.Generic;
using System.IO;
using System.Threading;
using System.Threading.Tasks;

namespace Stock
{
   public class StockBroker
   {
      public string BrokerName { get; set; }
      public List<Stock> stocks = new List<Stock>();

      private readonly string destPath =
Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "Lab1_output.txt");
      public static ReaderWriterLockSlim myLock = new ReaderWriterLockSlim();

      public StockBroker(string brokerName)
      {
         BrokerName = brokerName;

         // Write header to the log file
         using (StreamWriter outputFile = new StreamWriter(destPath, false))
         {
            string header = "Broker".PadRight(15) + "Stock".PadRight(15) + "Value".PadRight(10)
+ "Changes".PadRight(10) + "Date and Time";
            outputFile.WriteLine(header);
         }
      }

      public void AddStock(Stock stock)
      {
         stocks.Add(stock);
         stock.StockEvent += StockNotificationHandler;
      }

      private async void StockNotificationHandler(object sender, StockNotification e)
      {
         Stock stock = (Stock)sender;
         string message =
$"{BrokerName.PadRight(15)}{e.StockName.PadRight(15)}{e.CurrentValue.ToString().PadRight(
10)}{e.NumChanges.ToString().PadRight(10)}{DateTime.Now}";

         myLock.EnterWriteLock();
```

```
            try
            {

                using (StreamWriter outputFile = new StreamWriter(destPath, true))
                {
                    await outputFile.WriteLineAsync(message);
                }
                Console.WriteLine(message);
            }
            catch (IOException ex)
            {
                Console.WriteLine($"File writing error: {ex.Message}");
            }
            finally
            {
                myLock.ExitWriteLock();
            }
        }
    }
}
```