Daniel Monge
Howard Moon
Daniel Ramirez
Tyler Thorin

Assignment 1 - Stocks

Program.cs

```
using System;
using System.IO;

namespace Stock
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Directory.GetCurrentDirectory());
            Console.WriteLine(StockBroker.titles);
            Stock stock1 = new Stock("Technology", 160, 5, 15);
            Stock stock2 = new Stock("Retail", 30, 2, 6);
            Stock stock3 = new Stock("Banking", 90, 4, 10);
            Stock stock4 = new Stock("Commodity", 500, 20, 50);
            StockBroker b1 = new StockBroker("Broker 1");
            b1.AddStock(stock1);
            b1.AddStock(stock2);
            StockBroker b2 = new StockBroker("Broker 2");
            b2.AddStock(stock1);
            b2.AddStock(stock3);
            b2.AddStock(stock4);
            StockBroker b3 = new StockBroker("Broker 3");
            b3.AddStock(stock1);
            b3.AddStock(stock3);
            StockBroker b4 = new StockBroker("Broker 4");
            b4.AddStock(stock1);
            b4.AddStock(stock2);
            b4.AddStock(stock3);
            b4.AddStock(stock4);
        }
    }
}
```

Stock.cs

```
using System;
using System.Threading;

namespace Stock
{
    public class Stock
    {
        public event EventHandler<StockNotification> StockEvent;
        public string StockName { get; set; }
        public int InitialValue { get; set; }
        public int CurrentValue { get; set; }
        public int MaxChange { get; set; }
        public int Threshold { get; set; }
        public int NumChanges { get; set; }

        private Thread stockThread;

        /// <summary>
        /// Stock class that contains all the information and changes of
the stock
        /// </summary>
        /// <param name="name">Stock name</param>
        /// <param name="startingValue">Starting stock value</param>
        /// <param name="maxChange">The max value change of the
stock</param>
        /// <param name="threshold">The range for the stock</param>
        public Stock(string name, int startingValue, int maxChange, int
threshold)
        {
            this.StockName = name;
            this.InitialValue = startingValue;
            this.CurrentValue = startingValue;
            this.MaxChange = maxChange;
            this.Threshold = threshold;

            this.StartThread();
        }

        public void StartThread()
        {
            ThreadStart stockRef = new ThreadStart(Activate);
            stockThread = new Thread(stockRef);
            stockThread.Start();
        }
```

```
        public void StopThread()
        {
            stockThread.Abort();
        }

        /// <summary>
        /// Activates the threads synchronizations
        /// </summary>
        public void Activate()
        {
            for (int i = 0; i < 25; i++)
            {
                Thread.Sleep(500); // 1/2 second
                this.ChangeStockValue();
            }
            this.StopThread();
        }

        /// <summary>
        /// Changes the stock value and also raising the event of stock
value changes
        /// returns true if the stock goes above or below the threshold
        /// </summary>
        public void ChangeStockValue()
        {
            var rand = new Random();
            this.CurrentValue += rand.Next(-this.MaxChange,
this.MaxChange);
            this.NumChanges++;
            if (Math.Abs(this.CurrentValue - this.InitialValue) >
this.Threshold)
            {
                StockNotification a = new StockNotification(this.StockName,
this.CurrentValue, this.NumChanges);
                if (StockEvent != null)
                {
                    StockEvent(this, a);
                }
                this.InitialValue = this.CurrentValue;
            }
        }
    }
}
```

StockBroker.cs

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

namespace Stock
{
    public class StockBroker
    {
        public string BrokerName { get; set; }
        public List<Stock> stocks = new List<Stock>();
        readonly string docPath = Directory.GetCurrentDirectory();
        public static string titles = "Broker".PadRight(10) +
"Stock".PadRight(15) + "Value".PadRight(10) + "Changes".PadRight(10);

        /// <summary>
        /// The stockbroker object
        /// </summary>
        /// <param name="brokerName">The stockbroker's name</param>
        public StockBroker(string brokerName)
        {
            this.BrokerName = brokerName;
            docPath += "\\" + BrokerName + ".txt";
            File.WriteAllText(docPath, titles + "\n");
        }

        /// <summary>
        /// Adds stock objects to the stock list
        /// </summary>
        /// <param name="stock">Stock object</param>
        public void AddStock(Stock stock)
        {
            stocks.Add(stock);
            stock.StockEvent += new
EventHandler<StockNotification>(EventHandler);
        }

        /// <summary>
        /// The eventhandler that raises the event of a change
        /// </summary>
        /// <param name="sender">The sender that indicated a change</param>
        /// <param name="e">Event arguments</param>
        void EventHandler(Object sender, EventArgs e)
        {
            StockNotification data = (StockNotification)e;
            String statement = this.BrokerName.PadRight(10) +
```

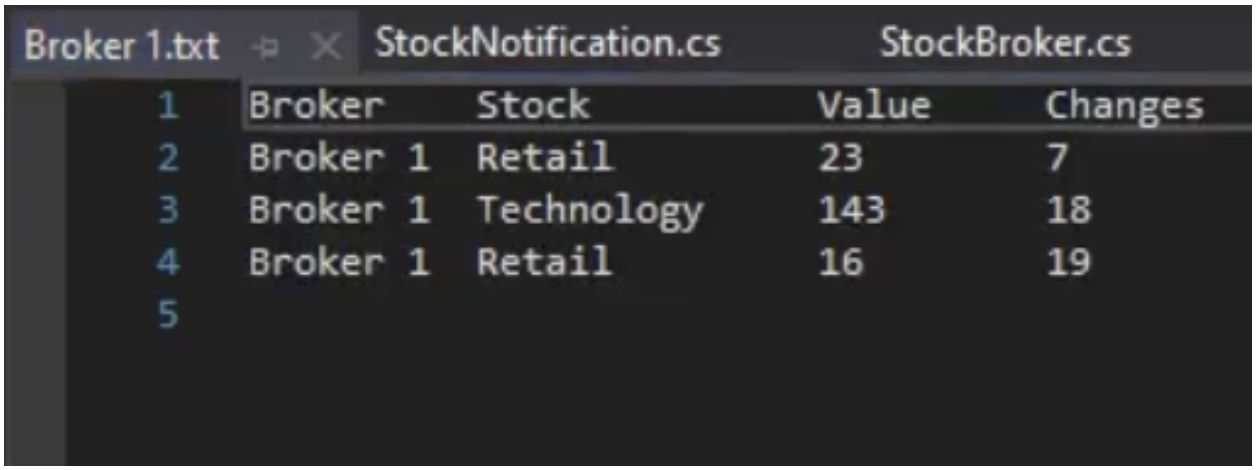
```
data.StockName.PadRight(15) +
                data.CurrentValue.ToString().PadRight(10) +
data.NumChanges.ToString().PadRight(10);
            Console.WriteLine(statement);
            LogEvent(statement);
        }

        void LogEvent(string line)
        {
            StreamWriter sw = File.AppendText(docPath);
            sw.WriteLine(line);
            sw.Close();
        }
    }
}
```

StockNotification.cs

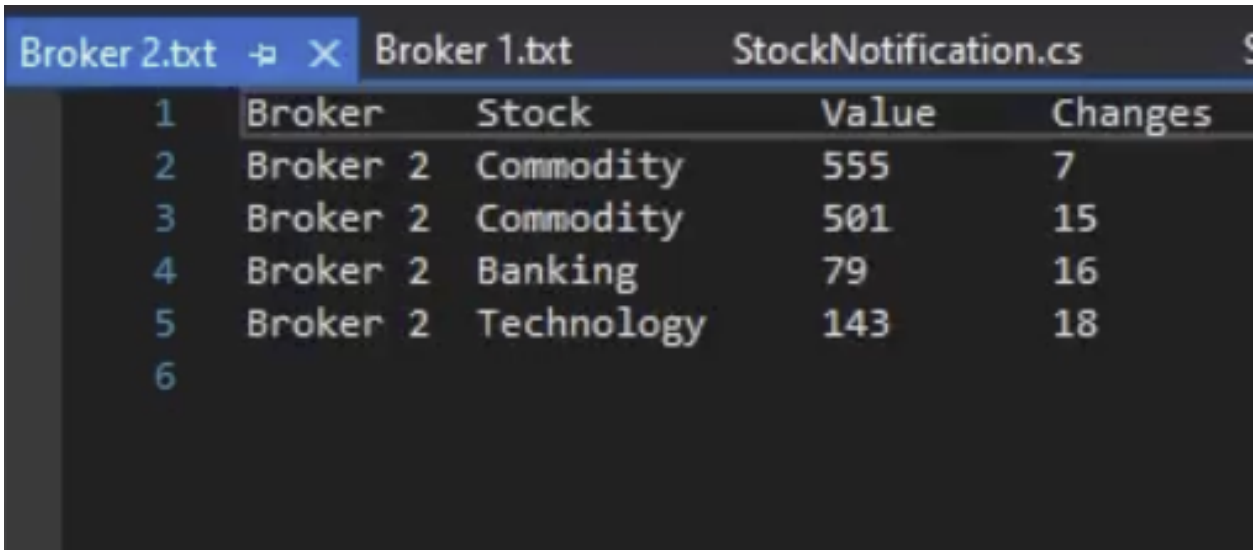
```
using System;

namespace Stock
{
    public class StockNotification : EventArgs
    {
        public string StockName { get; set; }
        public int CurrentValue { get; set; }
        public int NumChanges { get; set; }

        /// <summary>
        /// Stock notification attributes that are set and changed
        /// </summary>
        /// <param name="stockName">Name of stock</param>
        /// <param name="currentValue">Current value of the stock</param>
        /// <param name="numChanges">Number of changes the stock goes
through</param>
        public StockNotification(string stockName, int currentValue, int
numChanges)
        {
            this.StockName = stockName;
            this.CurrentValue = currentValue;
            this.NumChanges = numChanges;
        }
    }
}
```

Output of text files

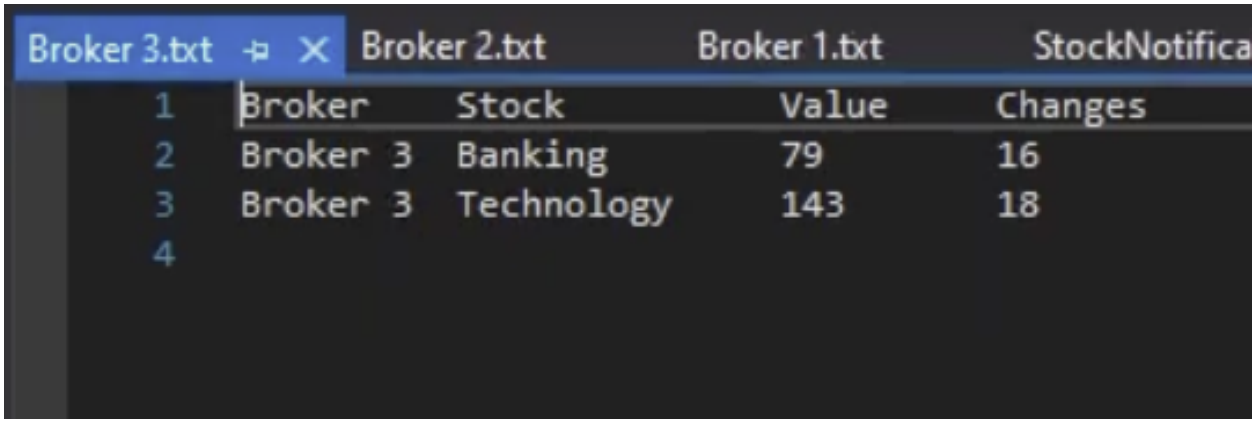
Broker1.txt



Broker2.txt



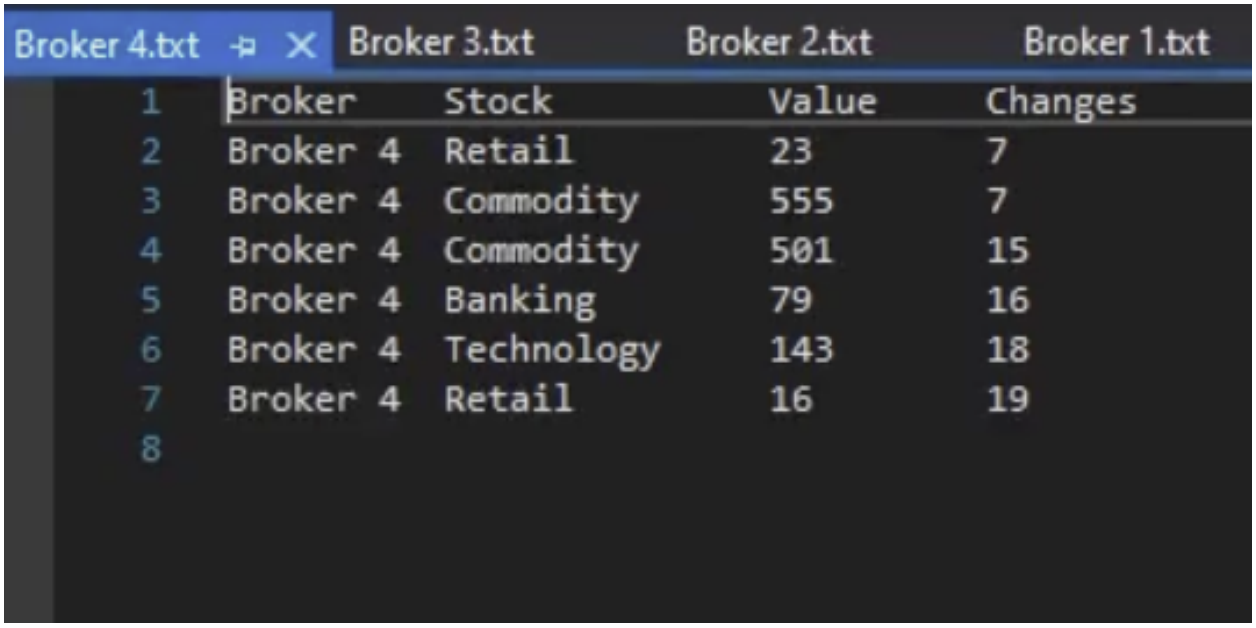
Broker3.txt



Broker 3.txt | Broker 2.txt | Broker 1.txt | StockNotifica

	Broker	Stock	Value	Changes
1				
2	Broker 3	Banking	79	16
3	Broker 3	Technology	143	18
4				

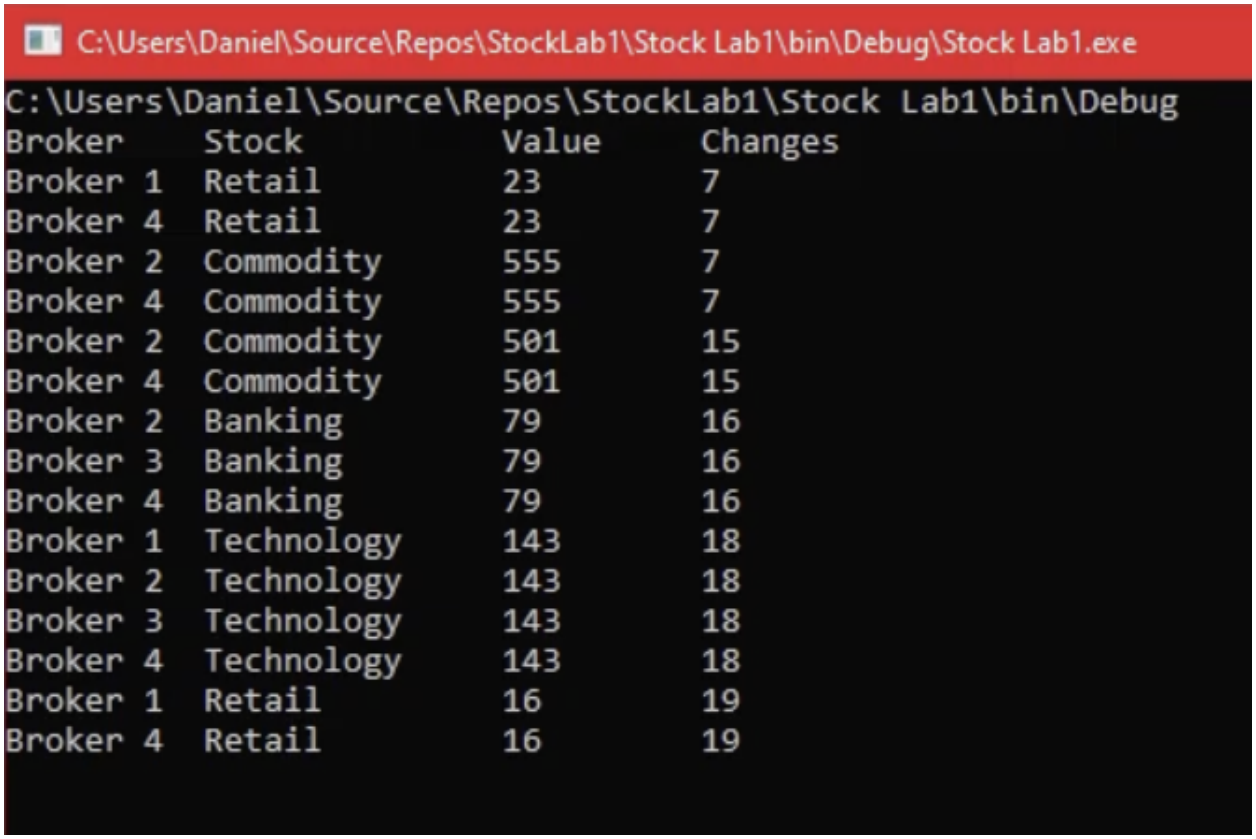
Broker4.txt



Broker 4.txt | Broker 3.txt | Broker 2.txt | Broker 1.txt

	Broker	Stock	Value	Changes
1				
2	Broker 4	Retail	23	7
3	Broker 4	Commodity	555	7
4	Broker 4	Commodity	501	15
5	Broker 4	Banking	79	16
6	Broker 4	Technology	143	18
7	Broker 4	Retail	16	19
8				

Console Output:



Demonstration:
https://www.youtube.com/watch?v=O4t50Im3IY8