

EWolf – Пояснительная записка

1. Название программы

EWolf (Exchange Wolf)

2. Аннотация

Программа позволяет протестировать стратегию алгоритмической торговли на бирже. Тестирование реализовано в режиме оффлайн с использованием исторических данных, загруженных из сети интернет. Пользователю доступна детальная статистика по каждой сделке и график изменения цены в соответствующий период. Визуальное представление (график и статистика) позволяют эффективно выявить сильные и слабые стороны стратегии и внести соответствующие правки.

3. Адрес репозитория

<https://github.com/KEF-Caesar/EWolf>

4. Участники команды и их роли

❖ Федоров Кирилл ББИ156(2)

- Проектирование структуры хранения данных
- Разработка методологии действия торгового алгоритма
- Реализация простого торгового алгоритма
- Реализация возможности тестирования оффлайн (на исторических данных)

❖ Марченко Екатерина ББИ152(2)

- Формирование запросов для получения необходимых данных из интернет-ресурса
- Загрузка, обработка и сохранение данных из интернет-ресурса
- Проектирование и дизайн графиков

❖ Савостьянова Дарья ББИ152(2)

- Проектирование и дизайн пользовательского интерфейса
- Реализация взаимодействия логики программы и ее интерфейса

5. Список классов и их краткое описание

EWolf Data

❖ Candle

Японская свеча (максимальная и минимальная цены, цены открытия и закрытия), объем сделок за этот период, момент времени, соответствующий концу периода

❖ Company

Компания, цены на её акции в виде японских свечей разных временных интервалов. Загрузка в оперативную память данных из соответствующего файла.

❖ Data

Список голубых фишек (компаний, акции которые наиболее ликвидны) и информация, соответствующая этим компаниям.

EWolf Trading Algorithms

❖ Order

Открытие/закрытие сделки (объем, момент времени)

- ❖ Deal
Сделка («от нуля до нуля»). Ордера с момента открытия позиции (когда текущий объем был равен нулю и совершается покупка/продажа) до момента полного закрытия позиции (обращения текущего объема в нуль). Статистика (моменты открытия и закрытия, цены входа и выхода, прибыль/убыток)
- ❖ Stock
Акция какой-либо компании, текущий объем и сделки, совершенные с ней.
- ❖ Main
Основная логика работы торгового алгоритма. Запросы на обновление данных и обработка данных (определение моментов открытия и закрытия сделок). Заполнение статистических данных и вызов события, означающего закрытие сделки.

EWolf_Test

- ❖ Offline_Test
Оффлайн эмулирование получения данных в режиме реального времени. Исторические данные находятся в «секретном» месте (о котором «не знает» торговый алгоритм). Один из методов класса, считывая их, создает файлы нужной версии (увеличивая количество информации с каждым шагом) в директории, с которой работает торговый алгоритм.
- ❖ Magic
Промежуточное звено между пользовательским интерфейсом и оффлайн тестированием. Его основная функция заключается в запуске оффлайн теста вместо работы в режиме реального времени.
- ❖ Program
Класс использовался для тестирования программы в консоли

EWolf_Input

- ❖ Repository_For_Candles
Содержит адрес интернет-ресурса, с которого загружаются данные. Параметры для универсального идентификатора ресурса (URI). Метод GetCandles для скачивания данных (строка) с помощью WebClient. Когда происходит событие завершения загрузки, вызывается метод для загрузки в данных в файл.
Метод GetCandles вызывается при создании объекта класса, данные для работы метода передаются в конструкторе.

EWolf_GUI

- ❖ MainWindow
Код данного окна (MainWindow.xaml.cs) создает экземпляр класса Magic, который запускает оффлайн тестирование. Экземпляр класса Magic содержит объект класса Main, на событие которого происходит подписание в данном окне. Данное окно берет информацию о сделке и название фирмы из объекта класса Main. Непосредственно окну (listBoxDeals) при этом передается информация о компании, с которой была заключена сделка, и Profit_or_loss - полученные в

итоге прибыль или потеря. Далее при выборе определенной ячейки листа активируется кнопка «Get more information», с помощью которой можно вызвать окно (Information) с более полной информацией о сделке. Это окно получает полную информацию о сделке, название компании и экземпляр класса Main (основной логики программы Trading_Algorithm).

❖ Information

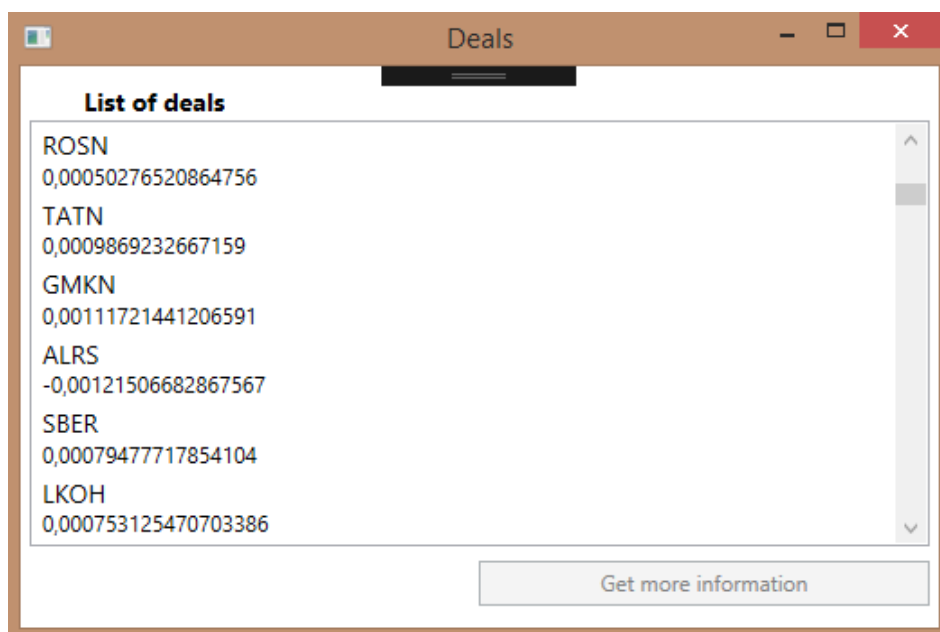
Information.xaml.cs при инициализации берет данные о конкретной выбранной сделке и выводит их на экран. Также передает эти данные следующему окну для того, чтобы можно было отобразить график по заданным свечам.

❖ Chart

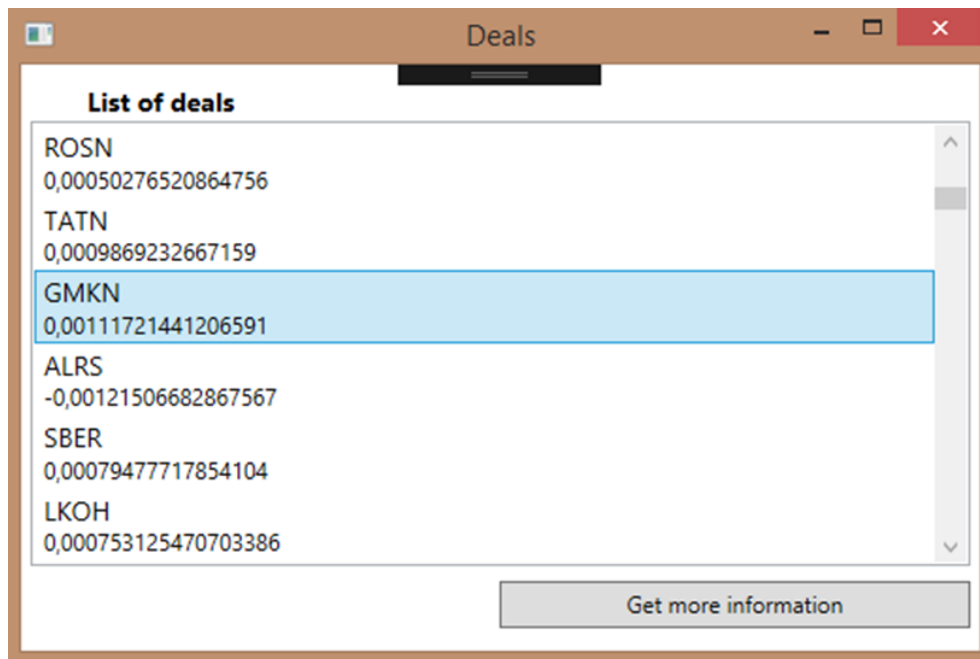
Chart.xaml.cs при инициализации создает график свечей, который строится на основе данных о выбранной сделке и компании из Information.xaml.cs. График использует данные даты и времени открытия и закрытия сделки (отображены на графике), данные свечи. С их помощью выбирается интервал для рисования графика. Реализовано рисование свечей на графике (стандартный класс Chart, область графика ChartAreas, элементы графика Series(Candlestick – Японская свеча)), изменение размеров графика относительно заданному интервалу свечей.

6. Интерфейс программы

Главное окно – MainWindow – открывается с началом работы программы:



Класс MainWindow.xaml описывает дизайн окна: содержит Label с обозначением данных, отображаемых в окне; listBoxDeals, показывающий все данные, передаваемые главной программой, и позволяющий выбрать определенную сделку и с помощью кнопки GetInfo открыть следующее окно (Information) с точными данными по конкретной сделке.



Класс Information.xaml имеет два listBox, которые отображают информацию о выбранной сделке, передаваемую от главного окна, и кнопку (Show chart) перехода к третьему окну (Chart), показывающему график свечей для данной сделки.

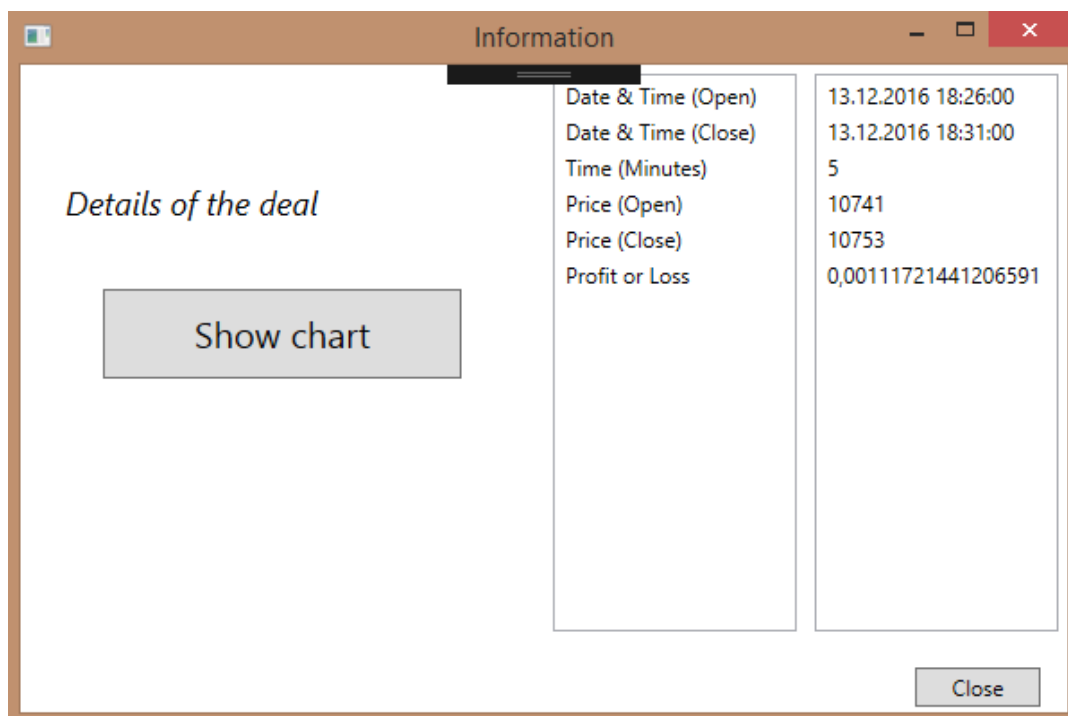


Chart.xaml содержит элемент WindowsFormsHost для иллюстрирования графика свечей для выбранной сделки.

Подключены ссылки System.Windows.Forms.DataVisualization, WindowsFormsIntegration и System.Windows.Forms

