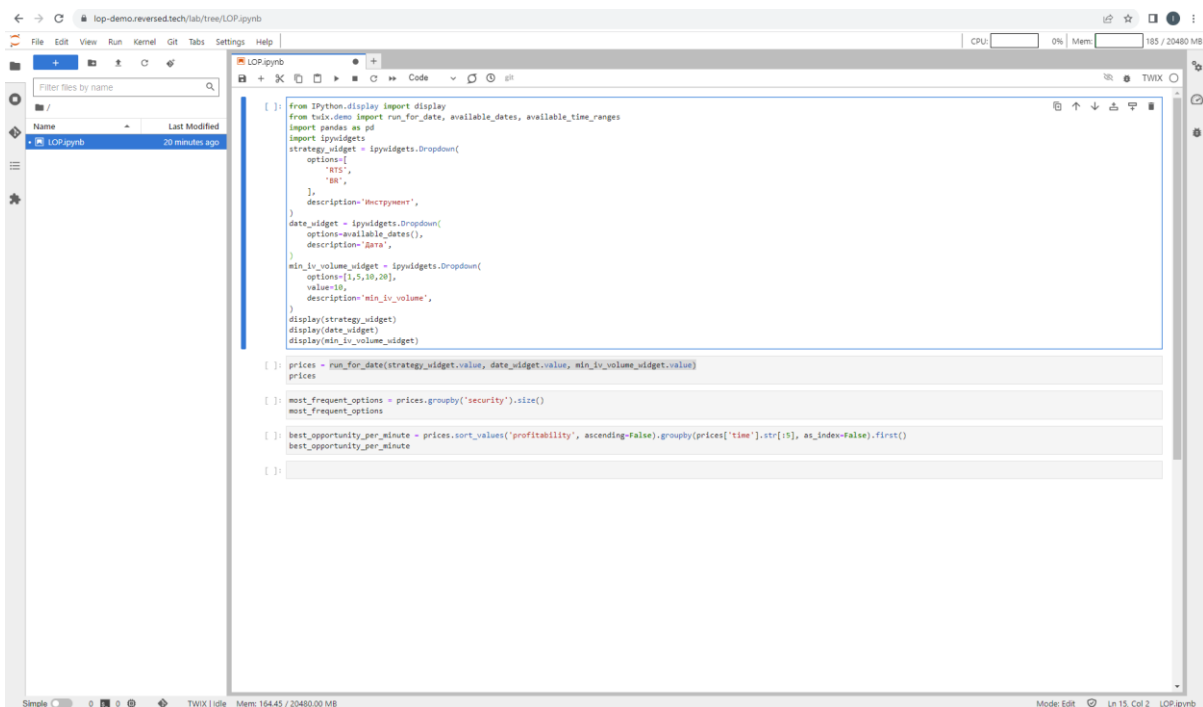


## Инструкция по эксплуатации экземпляра ПО «LOR / ЛОП», предоставленного для проведения экспертной проверки

Программа «LOR / ЛОП» (далее Программа) представляет собой дистрибутив (файл `tar.gz`), внутри которого находится код, написанный на языке C++. Данная программа должна быть встроена в ПО клиента для анализа биржевых данных или торговли на Московской Бирже. Для эксплуатации ПО требуется иметь доступ до данных с биржевой информацией. Для проведения экспертной проверки предоставлен удалённый доступ до демонстрационной платформы, на которой развёрнута скомпилированная версия Программы и реализован доступ до биржевых данных за июль 2023 года. Таким образом, демонстрируется возможная работа Программы в комплексе с клиентским ПО для анализа биржевых данных.

Эксперт получает возможность запуска ПО на различных торговых инструментах, с различными наборами параметров на выбранных датах. Интерфейс демонстрационной платформы является стандартным Jupyter Notebook под названием `LOR.ipynb`. Подробнее о функционале Jupyter Notebook можно прочесть, например, [здесь](#).

Демонстрационная платформа перед стартом работы выглядит следующим образом:



```
[ ]: from IPython.display import display
from twix_demo import run_for_date, available_dates, available_time_ranges
import pandas as pd
import ipynbwidgets
strategy_widget = ipynbwidgets.Dropdown(
    options=[
        'RTS',
        'BB',
    ],
    description='Инструмент',
)
date_widget = ipynbwidgets.Dropdown(
    options=available_dates(),
    description='Дата',
)
min_iv_volume_widget = ipynbwidgets.Dropdown(
    options=[1,5,10,20],
    value=10,
    description='min_iv_volume',
)
display(strategy_widget)
display(date_widget)
display(min_iv_volume_widget)

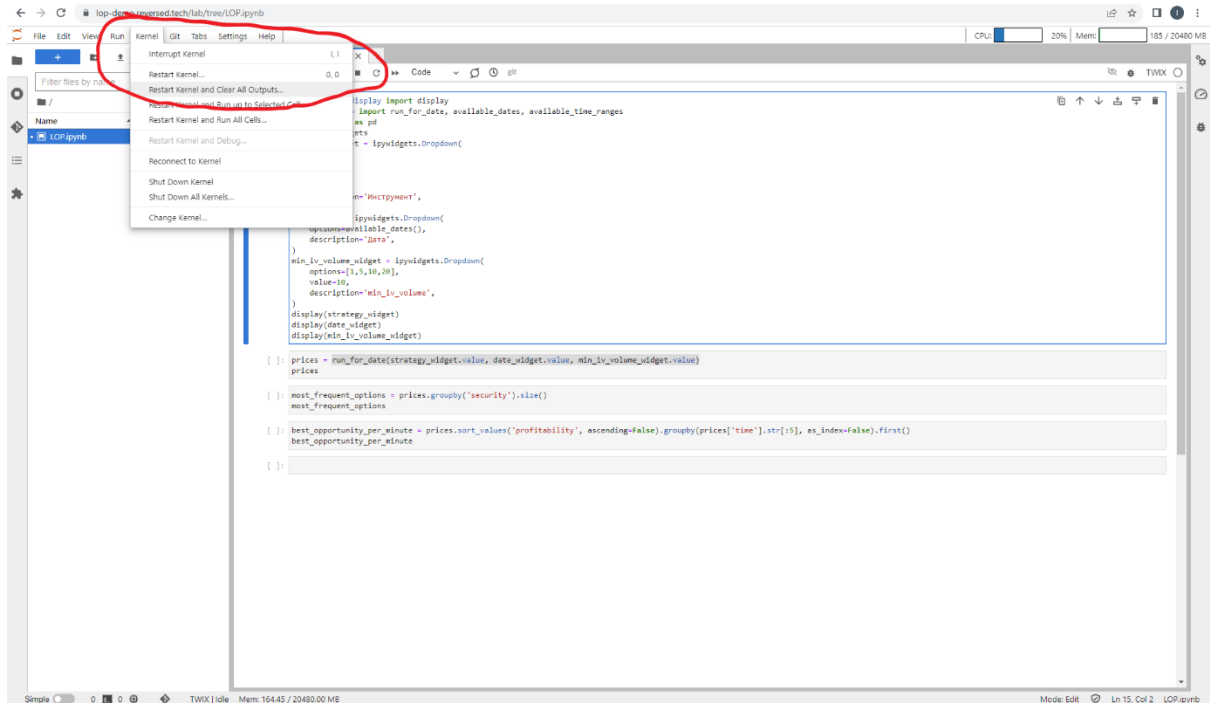
[ ]: prices = run_for_date(strategy_widget.value, date_widget.value, min_iv_volume_widget.value)
prices

[ ]: most_frequent_options = prices.groupby('security').size()
most_frequent_options

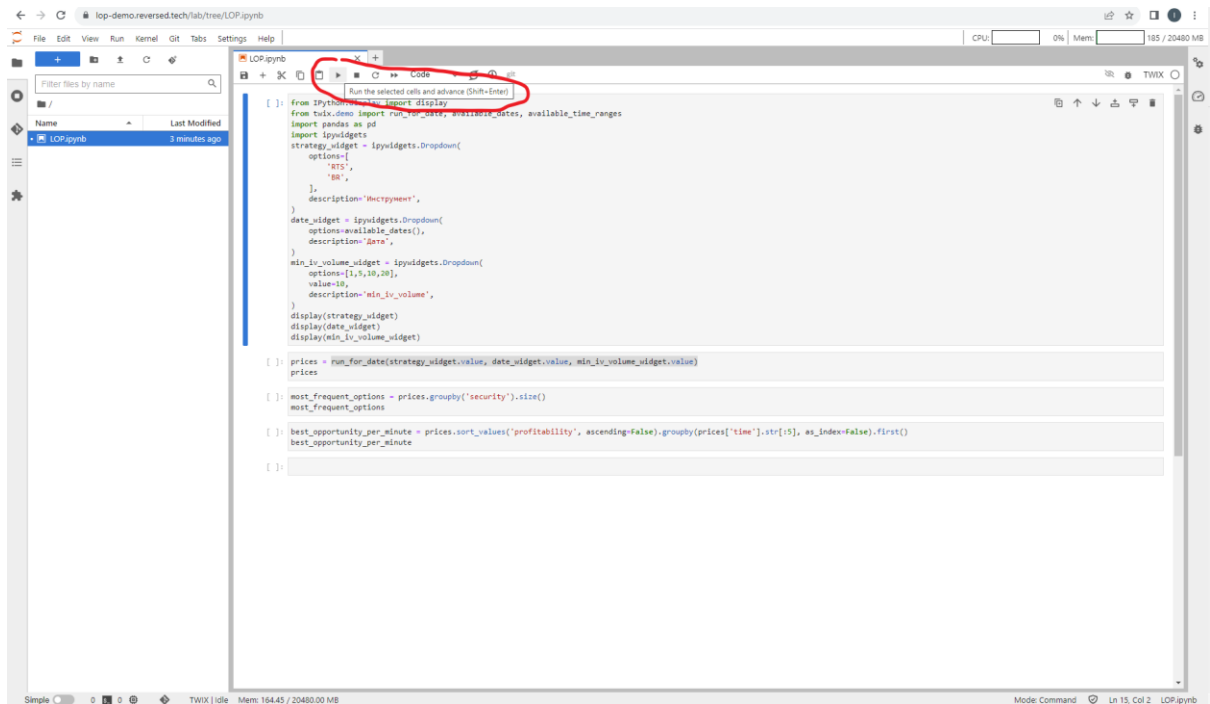
[ ]: best_opportunity_per_minute = prices.sort_values('profitability', ascending=False).groupby(prices['time'].str[1:5], as_index=False).first()
best_opportunity_per_minute

[ ]:
```

В случае, если доступ до демонстрационной платформы с использованием заданного пароля будет осуществляться несколько экспертов, при очередном запуске платформы могут быть видны результаты предыдущих запусков. В таком случае, требуется перед стартом работы очистить ноутбук командой Kernel → Restart Kernel and Clear All Outputs...



Первая клетка Jupyter ноутбука LOP.ipynb задаёт параметры для запуска ПО. Для запуска каждой клетки Jupyter ноутбука требуется нажать Shift+Enter или кнопку Play.



Перед запуском ПО требуется заполнить несколько форм, определяющих параметры, с которыми будет осуществляться запуск.

- **Инструмент.** Эксперту предлагается совершить выбор одного из доступных на платформе инструментов срочного рынка Московской Биржи. ПО будет получать биржевую информацию по ближайшему фьючерсу на выбранный актив, а также ближайшей недельной или месячной опционной серии. Доступен выбор одной из двух опций: 1) сентябрьский фьючерс на индекс RTS и ближайшие недельные опционные серии; 2) августовский фьючерс на нефть Brent и месячная опционная серия на него.
- **Дата.** Доступен список дней из июля 2023 года, когда осуществлялась биржевая торговля выбранными инструментами на срочном рынке ПАО «Московская биржа».
- **min\_iv\_volume.** Параметр ПО, отвечающий за то, опционные котировки какого объема ПО считает значимыми при расчёте вменяемой волатильности.

По умолчанию формы заполнены следующим образом:

```
[1]: from IPython.display import display
from tsix_demo import run_for_date, available_dates, available_time_ranges
import pandas as pd
import ipywidgets
strategy_widget = ipywidgets.Dropdown(
    options=[
        'rts',
        'нефть',
    ],
    description='Инструмент',
)
date_widget = ipywidgets.Dropdown(
    options=available_dates(),
    description='Дата',
)
min_iv_volume_widget = ipywidgets.Dropdown(
    options=[1,10,20],
    value=10,
    description='min_iv_volume',
)
display(strategy_widget)
display(date_widget)
display(min_iv_volume_widget)

Инструмент RTS
Дата 2023-07-03
min_iv_volu 10

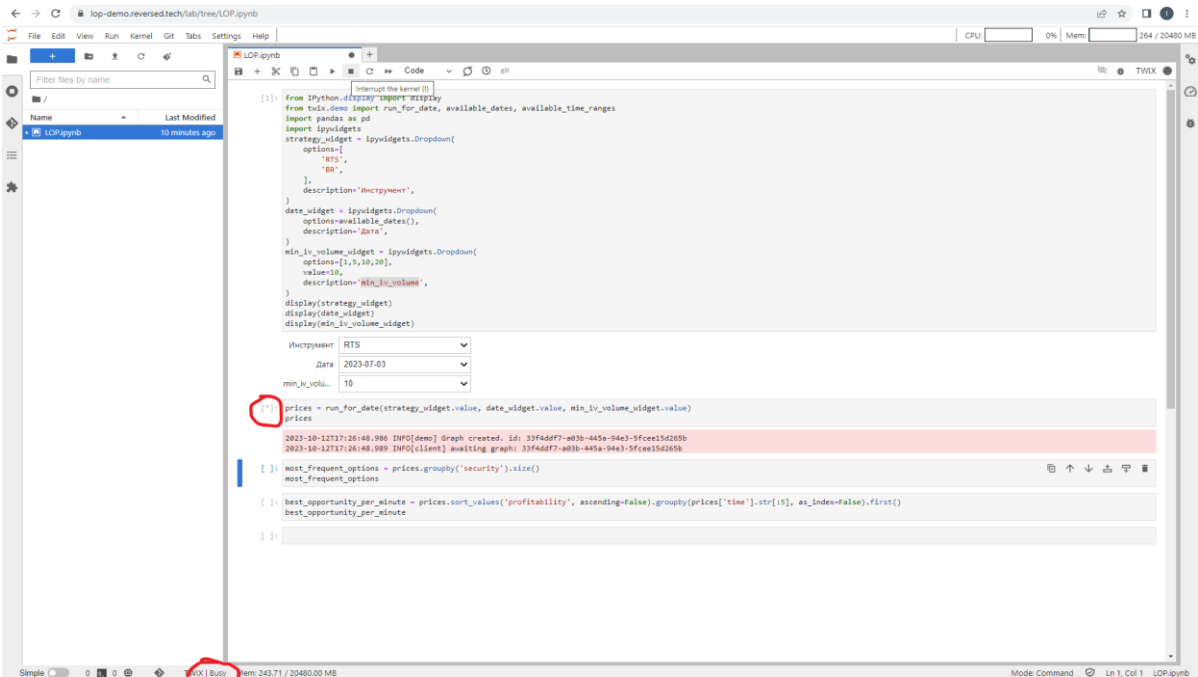
[ ]: prices = run_for_date(strategy_widget.value, date_widget.value, min_iv_volume_widget.value)
prices

[ ]: most_frequent_options = prices.groupby('security').size()
most_frequent_options

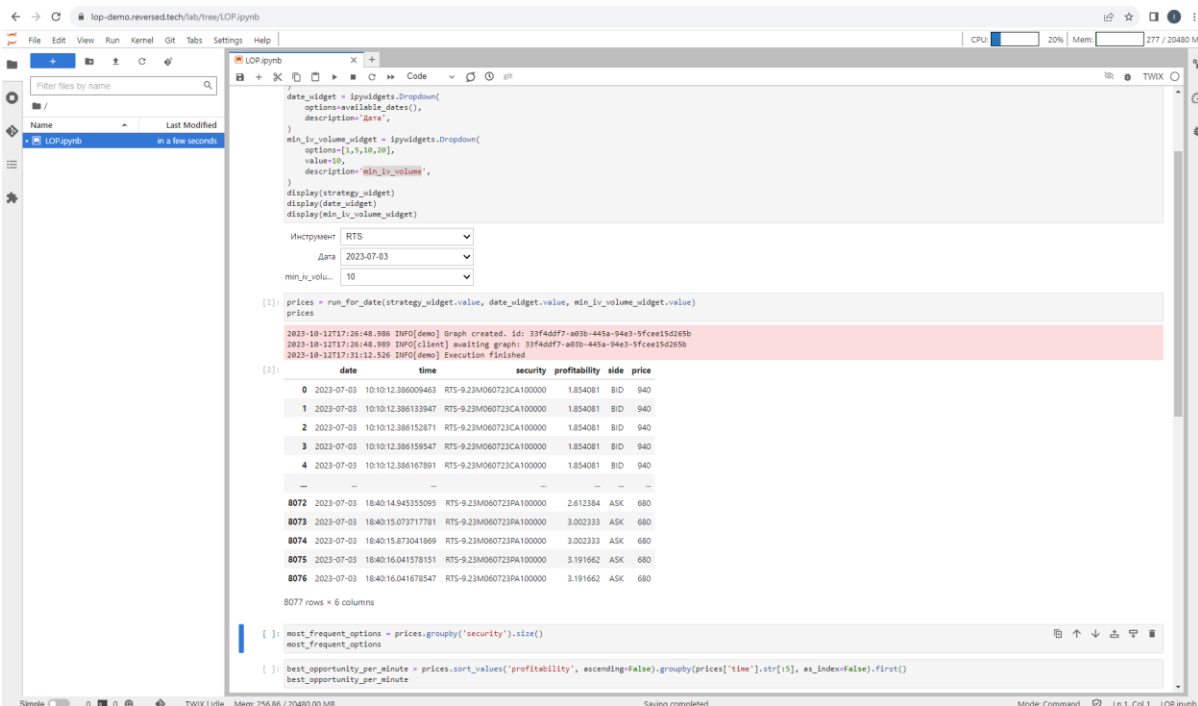
[ ]: best_opportunity_per_minute = prices.sort_values('profitability', ascending=False).groupby(prices['time'].str[:1], as_index=False).first()
best_opportunity_per_minute

[ ]:
```

Вторая клетка Jupyter ноутбука LOP.ipynb запускает ПО, которое получает биржевую информацию за указанный промежуток времени и производит расчёт оценок прибыльности котировок выбранной опционной серии после каждого обновления биржевой информации. Расчёт оценок занимает несколько минут, т. к. в течение торгового дня происходят миллионы обновлений биржевой информации. При проведении расчётов демонстрационная платформа выглядит следующим образом:



Результатом работы ПО является таблица со значениями оценок прибыльности котировок выбранной опционной серии. В данную таблицу попадают только котировки с ненулевой прибылью. Полученная таблица это pandas.DataFrame, подробнее о данном функционале можно прочесть [здесь](#).



В третьей и четвертой клетках Jupyter ноутбука LOP.ipynb добавлены примеры возможного дальнейшего использования полученного набора оценок. Так, в клетке 3 показана информация о том сколько апдейтов в течение дня в стакане соответствующих опционов возникали котировки с положительной профитностью, а в клетке 4 -- котировка с наивысшей оценкой профитности за каждую минуту. При желании эксперты имеют возможность провести собственный анализ таблицы со значениями предсказаний, используя стандартный инструментарий Jupyter notebook.

```

[1]: prices = run_for_date(strategy_widget.value, date_widget.value, min_iv_volume_widget.value)
prices
2023-10-12T17:36:48.986 INFO[demo] Graph created. Id: 33f4ddf7-a03b-445a-94e3-5fcee15d265b
2023-10-12T17:26:48.989 INFO[client] awaiting graph: 33f4ddf7-a03b-445a-94e3-5fcee15d265b
2023-10-12T17:31:12.528 INFO[demo] execution finished

[2]:
   date      time      security  profitability  side  price
0  2023-07-03  10:10:12.386009463  RTS-9.23M060723CA100000  1.854081  BID  940
1  2023-07-03  10:10:12.386133947  RTS-9.23M060723CA100000  1.854081  BID  940
2  2023-07-03  10:10:12.386152871  RTS-9.23M060723CA100000  1.854081  BID  940
3  2023-07-03  10:10:12.386159547  RTS-9.23M060723CA100000  1.854081  BID  940
4  2023-07-03  10:10:12.386167891  RTS-9.23M060723CA100000  1.854081  BID  940
--  --  --  --  --  --
8072 2023-07-03  18:40:14.945355095  RTS-9.23M060723PA100000  2.612304  ASK  680
8073 2023-07-03  18:40:15.073717781  RTS-9.23M060723PA100000  3.002333  ASK  680
8074 2023-07-03  18:40:15.873041869  RTS-9.23M060723PA100000  3.002333  ASK  680
8075 2023-07-03  18:40:16.041578151  RTS-9.23M060723PA100000  3.191662  ASK  680
8076 2023-07-03  18:40:16.041678547  RTS-9.23M060723PA100000  3.191662  ASK  680

8077 rows x 6 columns

[3]: most_frequent_options = prices.groupby('security').size()
most_frequent_options

[3]: security
RTS-9.23M060723CA100000    1175
RTS-9.23M060723CA102500     574
RTS-9.23M060723CA105000     41
RTS-9.23M060723CA180000      6
RTS-9.23M060723CA97500     132
RTS-9.23M060723PA100000    4756
RTS-9.23M060723PA102500    1177
RTS-9.23M060723PA105000     18
RTS-9.23M060723PA95000     13
RTS-9.23M060723PA97500     185
dtype: Int64

[4]: best_opportunity_per_minute = prices.sort_values('profitability', ascending=False).groupby(prices['time'].str[:5], as_index=False).first()
best_opportunity_per_minute

```

```

[4]:
   date      time      security  profitability  side  price
0  2023-07-03  10:10:11.681563533  RTS-9.23M060723CA100000  3.322179  BID  940
1  2023-07-03  10:11:49.855739273  RTS-9.23M060723PA100000  8.569213  BID  1290
2  2023-07-03  10:12:48.175783671  RTS-9.23M060723PA100000  17.081423  BID  1400
3  2023-07-03  10:14:02.569236845  RTS-9.23M060723PA102500  44.465516  BID  3080
4  2023-07-03  10:18:45.658183105  RTS-9.23M060723CA97500   4.852366  BID  2580
--  --  --  --  --  --
79  2023-07-03  18:16:00.165076629  RTS-9.23M060723CA102500  1.038642  ASK  320
80  2023-07-03  18:30:32.219957157  RTS-9.23M060723PA102500  6.935617  BID  2020
81  2023-07-03  18:31:37.679088175  RTS-9.23M060723PA102500  21.326783  BID  2000
82  2023-07-03  18:35:16.648786523  RTS-9.23M060723CA100000  1.007956  BID  1530
83  2023-07-03  18:40:16.041578151  RTS-9.23M060723PA100000  3.191662  ASK  680

84 rows x 6 columns

[ ]:

```

Напоминаем, что в случае, если доступ до демонстрационной платформы с использованием заданного пароля будут осуществлять несколько экспертов, то каждому из экспертов после завершения работы требуется сбросить все результаты запросов с помощью команды Kernel → Restart Kernel and Clear All Outputs...:

