

# Разработка Web-приложения для анализа данных биржи

Гурин В.В., Смуглиенко А.С., Павлов А.Н., Колпаков Б.П.  
10И класс

Куратор: Гареев Артур Маратович

ОАНО «Первый Лобачевского»

Усть-Лабинск, 2023

# Проблема

Инвесторы часто сталкиваются с трудностью в анализе и отслеживании большого количества финансовых данных акций, которые могут повлиять на их решения по инвестированию.

# Цель

Создать приложение для удобного сбора и анализа финансовых данных акций, которое поможет начинающим инвесторам.

# Постановка задач

1. Собрать информацию, интересную целевой аудитории.
2. Определить функционал приложения и используемые технологии.
3. Написать программу приложения, собрать отдельные элементы и концепций в единый продукт.
4. Протестировать приложение на предмет недочётов, неисправностей.
5. Показать приложение аудитории с целью получения пользовательской информации.

# Определения

**Web-приложение** – клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи **браузера**. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, **на сервере**, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты **не зависят** от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными службами.

# Определения

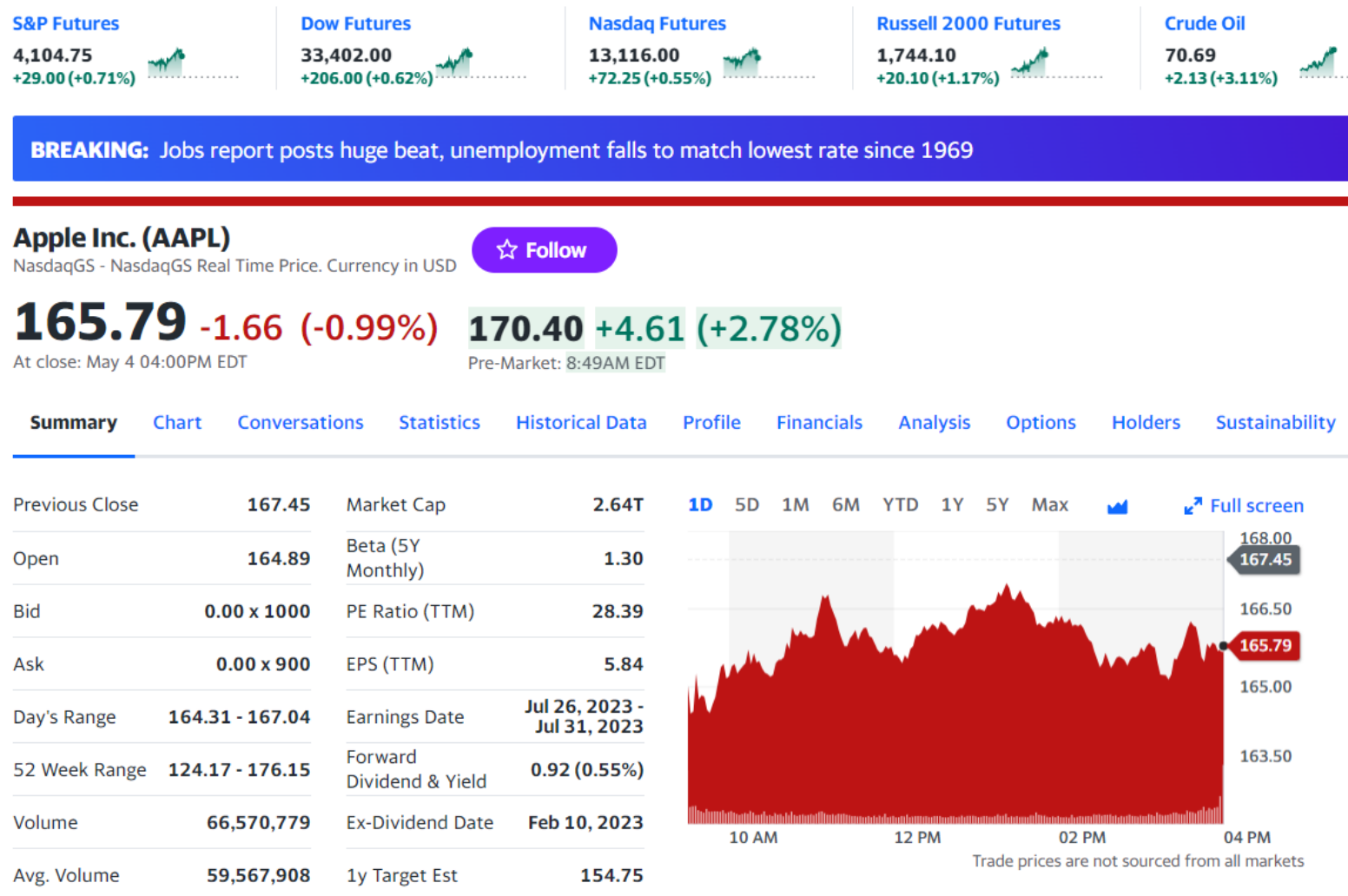
**Тикер** – краткое название в биржевой информации котируемых инструментов (акций, облигаций, индексов). Является уникальным идентификатором в рамках одной биржи или информационной системы.

# Используемые в работе сокращения

**Интегрированная среда разработки (IDE)** – это программное приложение, которое помогает программистам эффективно разрабатывать программный код

# Решение задачи 1

## Задача 1. Собрать информацию, интересную целевой аудитории.





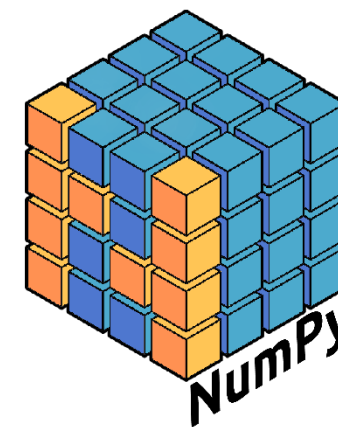
# Решение задачи 2.1. Структура

1. Общая информация о компании.
2. Отображение статистических данных:
  - Мультипликаторы
  - Дивиденды
3. Отображение графиков:
  - скользящая средняя
  - изменение биржевых котировок,
  - изменение объема торгов
4. Гибкое формирование аналитики  
по мультипликаторам **P/E, P/S, PEG, P/B**

# Решение задачи 2.2. Технологии

**IDE** – PyCharm Community.

**Библиотеки** – streamlit, pandas, numpy, requests.



# Решение задачи 3. Реализация

**Задача 3.** Написать программу приложения, собрать отдельные элементы и концепций в единый продукт

## **Подзадачи.**

Написание программного кода для таких разделов, как:

- Общая информация о компании
- Статистика
- Историческая информация
- Финальная аналитика

# Решение задачи 3.1

```
def load_summary_data(ticker_url):  
    summary_url = ticker_url  
    summary_data =  
load_data(summary_url)  
    data = [summary_data[0],  
summary_data[1]]  
    data = pd.concat(summary_data)  
    data.reset_index(drop=True,  
inplace=True)  
    return data[:8:], data[8::]
```

Загрузка общей информации

## Общая информация о компании

	0	1
0	Previous Close	167.45
1	Open	164.89
2	Bid	0.00 x 1000
3	Ask	0.00 x 900
4	Day's Range	164.31 - 167.04
5	52 Week Range	124.17 - 176.15
6	Volume	66570779
7	Avg. Volume	59567908

	0	1
8	Market Cap	2.64T
9	Beta (5Y Monthly)	1.30
10	PE Ratio (TTM)	28.39
11	EPS (TTM)	5.84
12	Earnings Date	Jul 26, 2023 - .
13	Forward Dividend & Yield	0.92 (0.55%)
14	Ex-Dividend Date	Feb 10, 2023
15	1y Target Est	171.29

# Решение задачи 3.2. Статистика

```
def load_statistics_mult(ticker_url):
    statistics_data = load_data(ticker_url.replace("?",
"/key-statistics?"))
    data = statistics_data[0]
    data = data.iloc[:-1, :]
    data.reset_index(drop=True, inplace=True)
    df = data.transpose()
    df.columns = df.iloc[0]
    df = df.drop(0, errors='ignore')
    df = df.reset_index(drop=True)
    return data, dict(df.to_dict())

def load_statistics_div(ticker_url):
    try:
        statistics_data = load_data(ticker_url.replace("?",
"/key-statistics?"))
        data = statistics_data[2]
        data = data.iloc[:-1, :]
        data.reset_index(drop=True, inplace=True)
        return data
    except:
        return "Информации о дивидендах компании нет"
```

Статистика		
Мультипликаторы		
	0	1
0	Market Cap (intraday)	40.68B
1	Enterprise Value	42.40B
2	Trailing P/E	30.97
3	Forward P/E	24.04
4	PEG Ratio (5 yr expected)	2.66
5	Price/Sales (ttm)	5.92
6	Price/Book (mrq)	7.25
7	Enterprise Value/Revenue	6.12
Дивиденды		
	0	
0	Avg Vol (3 month)	3
1	Avg Vol (10 day)	3
2	Shares Outstanding	5
3	Implied Shares Outstanding	6
4	Float	8
5	% Held by Insiders	1
6	% Held by Institutions	1
7	Shares Short (Feb 14, 2023)	4
8	Short Ratio (Feb 14, 2023)	4
9	Short % of Float (Feb 14, 2023)	4

Загрузка мультипликаторов и дивидендов

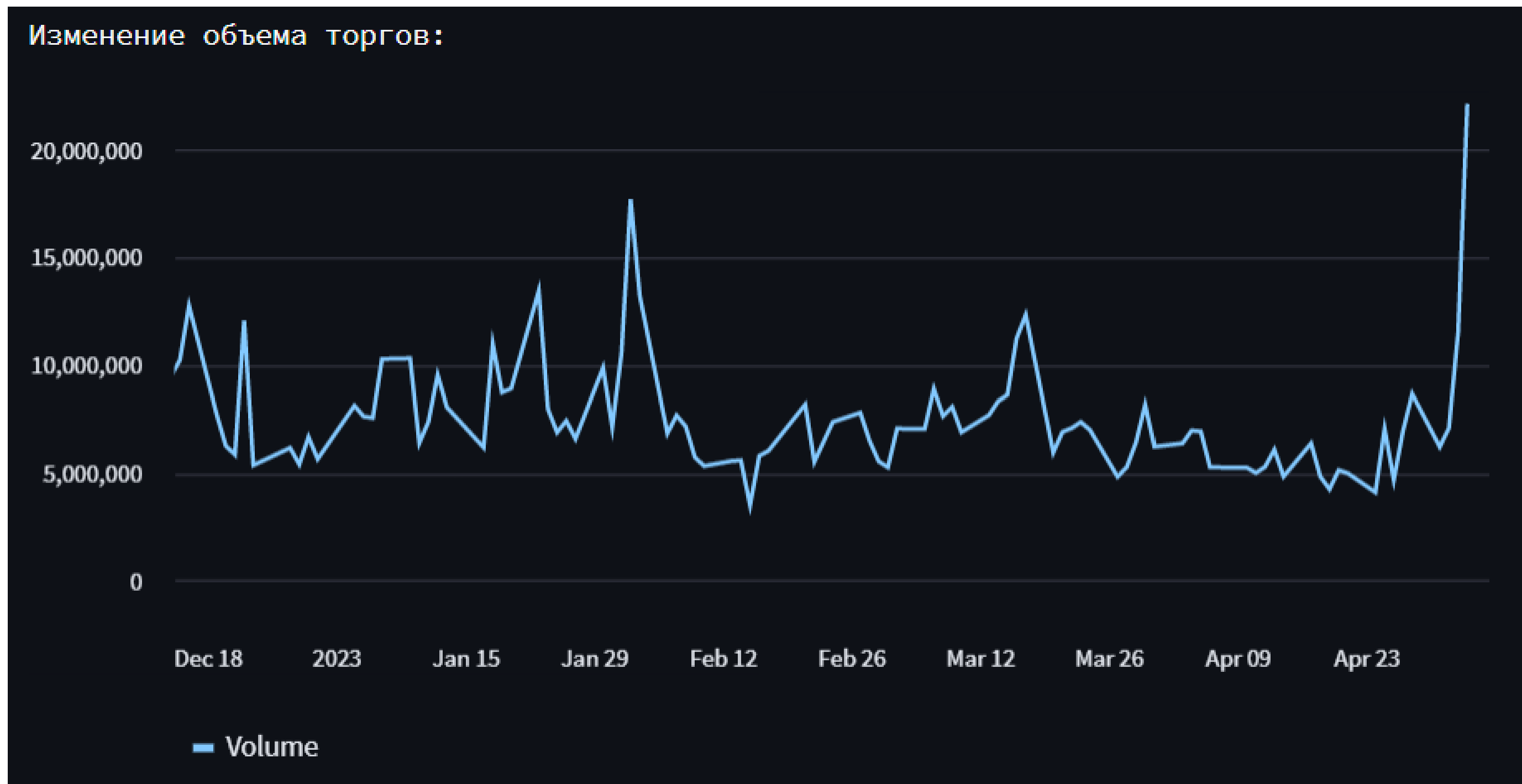
# Решение задачи 3.3. Графики

```
def load_historical_data(ticker_url):
    historical_data =
load_data(ticker_url.replace("?", "/history?"))
    data = historical_data[0]
    data = data.iloc[:-1, :]
    data =
data[~data["Open"].str.contains("Dividend")]
    data['Date'] = pd.to_datetime(data['Date'],
format='%b %d, %Y')
    data = data.dropna(subset=['Date'])
    numeric_columns = list(data.columns)[1:]
    data[numeric_columns] =
data[numeric_columns].replace(',', '', \
regex=True).astype(np.float64)
    data.set_index('Date', inplace=True)
    data = data.iloc[:::-1]
    return data.dropna()
```

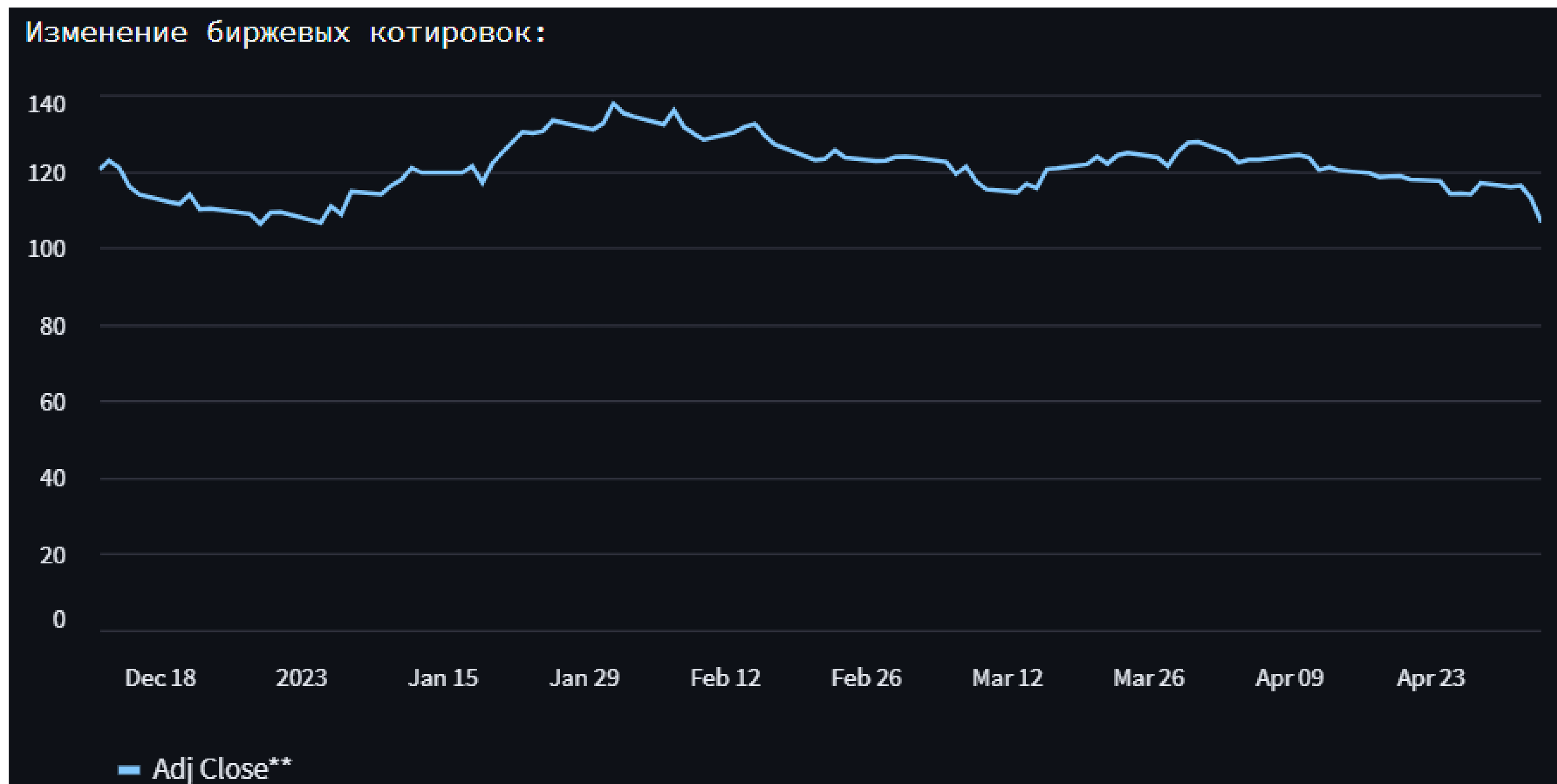


Загрузка данных для графиков

# Решение задачи 3.3. Графики



# Решение задачи 3.3. Графики





# Решение задачи 3.4. Аналитика

```
def load_analysys_data(df):
    analys = []
    if df.get('Trailing P/E'):
        if float(df['Trailing P/E'][1]) > 16:
            analys += ['По маркеру P/E (Цена (Price)/Чистая прибыль (Earnings Ratio)) компания переоценена.']
        elif 12 <= df.get(float(df['Trailing P/E'][1])) <= 15:
            analys += ['По маркеру P/E (Цена (Price)/Чистая прибыль (Earnings Ratio)) компания оценена справедливо.']
        elif df.get(float(df['Trailing P/E'][1])) < 12:
            analys += ['По маркеру P/E (Цена (Price)/Чистая прибыль (Earnings Ratio)) компания недооценена.']
    if df.get('Price/Sales (ttm)'):
        if 1 < float(df['Price/Sales (ttm)'][1]) <= 2:
            analys += ['По маркеру P/S (Рыночная стоимость компании (Price)/Объем продаж (Sales)) компания быстро окупается и соответствует своей цене.']
        elif float(df['Price/Sales (ttm)'][1]) >= 2:
            analys += ['По маркеру P/S (Рыночная стоимость компании (Price)/Объем продаж (Sales)) компания медленно окупается и переоценена.']
        elif float(df['Price/Sales (ttm)'][1]) <= 1:
            analys += ['По маркеру P/S (Рыночная стоимость компании (Price)/Объем продаж (Sales)) компания очень быстро окупается и недооценена.']
    if df.get('Price/Book (mrq)'):
        if float(df['Price/Book (mrq)'][1]) < 1:
            analys += ['По маркеру P/B (Рыночная стоимость компании (Price)/Балансовая стоимость активов компании (Book)) компания недооценена.']
        elif float(df['Price/Book (mrq)'][1]) > 5:
            analys += ['По маркеру P/B (Рыночная стоимость компании (Price)/Балансовая стоимость активов компании (Book)) компания переоценена.']
        elif 1 < float(df['Price/Book (mrq)'][1]) < 5:
            analys += ['По маркеру P/B (Рыночная стоимость компании (Price)/Балансовая стоимость активов компании (Book)) компания соответствует своей цене.']
    if df.get('Enterprise Value/Revenue'):
        if float(df['Enterprise Value/Revenue'][1]) < 0:
            analys += ['По маркеру PEG P(Market Cap (Капитализация))/E (Earnings(Чистая прибыль))/EGR (Годовой прогноз EPS(Earnings Per Share(ожидаемый рост прибыли на акцию)) компания имеет отрицательную чистую прибыль. Критерий не может адекватно оценить ее потенциал.']
        elif 0 < float(df['Enterprise Value/Revenue'][1]) < 1:
            analys += ['По маркеру PEG P(Market Cap (Капитализация))/E (Earnings(Чистая прибыль))/EGR (Годовой прогноз EPS(Earnings Per Share(ожидаемый рост прибыли на акцию)) компания недооценена инвестиционно, это привлекательно для инвестора.']
        elif 3 > float(df['Enterprise Value/Revenue'][1]) > 1:
            analys += ['По маркеру PEG P(Market Cap (Капитализация))/E (Earnings(Чистая прибыль))/EGR (Годовой прогноз EPS(Earnings Per Share(ожидаемый рост прибыли на акцию)) компания оценена оптимально.']
        elif float(df['Enterprise Value/Revenue'][1]) > 3:
            analys += ['По маркеру PEG P(Market Cap (Капитализация))/E (Earnings(Чистая прибыль))/EGR (Годовой прогноз EPS(Earnings Per Share(ожидаемый рост прибыли на акцию)) компания не привлекательная для инвестора из-за высокой перекупленности.']
    return analys
```

## Финальная аналитика компании

По маркеру P/E (Цена (Price)/Чистая прибыль (Earnings Ratio)) компания переоценена.

По маркеру P/S (Рыночная стоимость компании (Price)/Объем продаж (Sales)) компания медленно окупается и переоценена.

По маркеру P/B (Рыночная стоимость компании (Price)/Балансовая стоимость активов компании (Book)) компания переоценена.

По маркеру PEG P(Market Cap (Капитализация))/E (Earnings(Чистая прибыль))/EGR (Годовой прогноз EPS(Earnings Per Share(ожидаемый рост прибыли на акцию)) компания не привлекательная для инвестора из-за высокой перекупленности.

Анализ полученной информации

# Решение задачи 4. Тестирование

```
IndexError: list index out of range
```

```
Traceback:
```

```
File "C:\Users\SmuglienkoAS\AppData\Roaming\Python\Python311\site-packages\str
exec(code, module.__dict__)
File "C:\Users\SmuglienkoAS\PycharmProjects\pythonProject1\main.py", line 142,
df1, df2 = load_summary_data(ticker_url)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "C:\Users\SmuglienkoAS\PycharmProjects\pythonProject1\main.py", line 53,
data = [summary_data[0], summary_data[1]]
~~~~~
```

```
"streamlit" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.
```

```
2023-03-05 16:13:06.221
```

```
Warning: to view this Streamlit app on a browser, run it with the following
command:
```

```
streamlit run C:\Users\GurinVV\PycharmProjects\hackathon\app.py [ARGUMENTS]
```



## stackoverflow

# Заключение

В ходе разработки приложения мы освоили множество актуальных технологий и сфер проектирования веб-приложения, а также разобрались в устройстве биржевых котировок и современных способах их анализа через графики и мультипликаторы.

# Приложение



# Список использованной литературы

1. [aws.amazon.com/ru/what-is](https://aws.amazon.com/ru/what-is)
2. <https://ru.wikipedia.org/>
3. [finance.yahoo.com](https://finance.yahoo.com)
4. [ru.tradingview.com](https://ru.tradingview.com)
5. [stackoverflow.com](https://stackoverflow.com)