

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**  
**(Университет ИТМО)**

Кафедра Систем Управления и Информатики

**Лабораторная работа №3**

**Вариант №6**

Выполнил:

Таякин Д.Р.

Проверил:

Мусаев А.А.

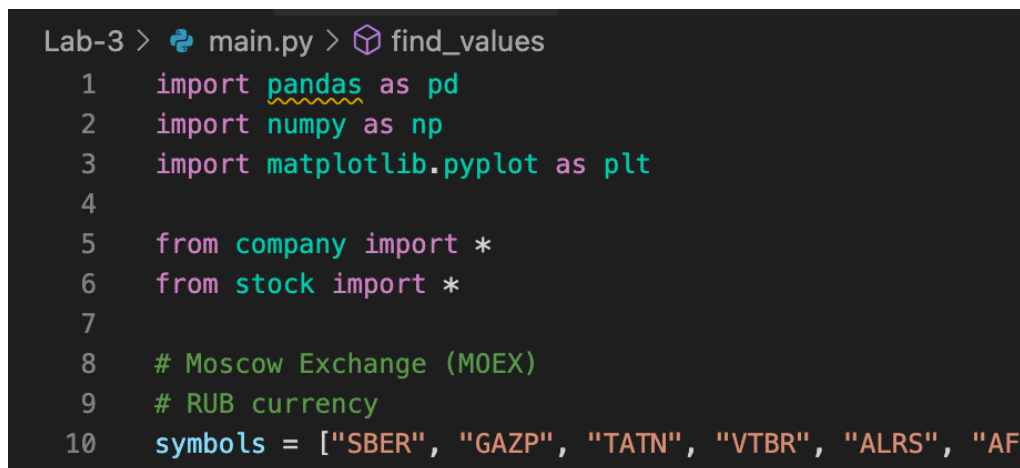
Санкт-Петербург, 2022

## 1 Задание

В первом и единственном задании нужно оценить вложения каждого человека в разные компании в зависимости от разных условий вложения. Анатолий вкладывал поровну в 3 пары компании с наименьшими (близкими к 0) коэффициентами парной корреляции. Борис - поровну в 3 пары компании так же как и Анатолий, только с условием того, что парные коэффициенты корреляции должны быть наибольшими. А Евгений делал вложения во все компании пропорционально их капитализации. Период от покупки до продажи составлял 6 месяцев.

Перечень исследуемых компаний: Газпром, Татнефть, Сбербанк, ВТБ, Алроса, Аэрофлот, РусГидро, Московская Биржа, НЛМК, Северсталь, Детский Мир, Полиметалл, Яндекс, АФК Система, Группа ЛСР, Ленэнерго, Лукойл, МТС, Новатэк и ПИК.

Для реализации поставленной задачи я решил найти исходные данные на просторах интернета.

A screenshot of a code editor window with a dark background. The title bar shows 'Lab-3 > main.py > find\_values'. The code is as follows:

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 from company import *
6 from stock import *
7
8 # Moscow Exchange (MOEX)
9 # RUB currency
10 symbols = ["SBER", "GAZP", "TATN", "VTBR", "ALRS", "AFK"]
```

Рисунок 1 – Импортируемые модули и перечисленные акции

На рисунке 1 представлены импортируемые модули, которые дальше будут использоваться в программе. Также в переменной `symbols` перечислены все необходимые акции (20 штук).

На рисунке 2 изображена часть кода основной логики программы. В данной части объявлены основные переменные программы и представлен

```

29
30 if __name__ == "__main__":
31     total_money = 10_000_000
32     total_months = 3 * 12 # years * months
33
34     people = {
35         "Anatoliy": {"money": total_money // 3, "stocks": [], "money_log": []},
36         "Boris": {"money": total_money // 3, "stocks": [], "money_log": []},
37         "Evgeniy": {"money": total_money // 3, "stocks": [], "money_log": []}
38     }
39
40     # starts from 6 – previously we analyzed stocks for 6 months (2016)
41     for i in range(6, total_months+1, 6):
42
43         sum_market_cap = 0
44         all_prices = []
45
46         # data of companies for period
47         companies = []
48
49         for symbol in symbols:
50             df = pd.read_csv(f"./data/{symbol}.csv", sep=';', names=['date', 'price', 'change', 'cap'])
51             all_prices.append(df['price'][i:i+6])
52             sum_market_cap += df['cap'][i]
53
54             companies.append(
55                 Company(symbol, list(df['price'][i:i+6]), df['cap'][i])
56             )
57

```

Рисунок 2 – Объявление основных переменных и считывание данных из csv файлов

алгоритм считывания данных из csv файлов, используя библиотеку pandas. Написан цикл обработки данных за 6-месячные периоды от покупки до продажи акций.

Также на рисунке 2 можно встретить объекты Company и Stock для удобного хранения и обработки данных. На рисунке 3 и 4 представлены классы этих объектов.

```

1 class Company:
2     def __init__(self, name, prices, cap):
3         # stock name = company name
4         self.name = name
5         self.prices = prices
6         self.cap = cap

```

Рисунок 3 – Класс Company

```

1 class Stock:
2     def __init__(self, name, number, price):
3         # stock name = company name
4         self.name = name
5         self.number = number
6         self.price = price

```

Рисунок 4 – Класс Stock

```

58         for _, assets in people.items():
59             for stock in assets["stocks"].copy():
60                 company = companies[symbols.index(stock.name)]
61                 assets["money"] += stock.number * company.prices[len(company.prices) - 1]
62                 assets["stocks"].pop()
63
64             assets["money_log"].append(assets["money"])
65

```

Рисунок 5 – Продажа акций в каждой итерации цикла для 6-месячных периодов

На рисунке 5 представлен код отвечающий за продажу акций всеми 3 участниками инвестирования за 6-месячный период. Также записываем в конце каждой продажи акций сумму денег, которые получили от продажи участники для последующего анализа данных.

```

66     if i != total_months:
67         # Evgeniy case
68         temp_money = 0
69         for company in companies:
70             stock_price = company.prices[len(company.prices) - 1]
71             stocks_number = people['Evgeniy']['money'] * (company.cap / sum_market_cap) // stock_price
72             people['Evgeniy']['stocks'].append(Stock(company.name, stocks_number, stock_price))
73             temp_money += stocks_number * stock_price
74         people['Evgeniy']['money'] -= temp_money
75
76
77         # finds Pearson's coefficient matrix
78         coef = np.corrcoef(all_prices)
79
80         min_values = find_values(coef, which="min", k=3)
81         max_values = find_values(coef, which="max", k=3)
82
83         def buy_stocks(who, stocks_names):
84             temp_money = 0
85             for stock_name in stocks_names:
86                 company = companies[symbols.index(stock_name)]
87                 stock_price = company.prices[len(company.prices) - 1]
88                 stocks_number = people[who]['money'] / 3 * 0.5 // stock_price
89                 people[who]['stocks'].append(Stock(company.name, stocks_number, stock_price))
90                 temp_money += stocks_number * stock_price
91             people[who]['money'] -= temp_money
92
93         anatoliy_stocks = []
94         for min_value in min_values:
95             i, j = np.where(coef == min_value)
96             anatoliy_stocks.append(symbols[i[0]])
97             anatoliy_stocks.append(symbols[j[0]])
98         buy_stocks("Anatoliy", anatoliy_stocks)
99
100         boris_stocks = []
101         for max_value in max_values:
102             i, j = np.where(coef == max_value)
103             boris_stocks.append(symbols[i[0]])
104             boris_stocks.append(symbols[j[0]])
105         buy_stocks("Boris", boris_stocks)

```

Рисунок 6 – Основная бизнес-логика покупки акиций участниками при разных условиях инвестирования

На рисунке 6 представлена основная бизнес-логика вложения участниками денег при разных условиях инвестирования. Для Евгения это просто пропорциональное вложение денег в акции всех компаний относительно капитализации. Для Анатолия и Бориса высчитываются коэффициенты корреляции Пирсона, которые, в итоге, представлены в квадратичной матрицы. После выполнения некоторых действий с матрицей мы находим в какие парные компании лучше всего вкладывать Анатолию и Борису относительно наименьших парных корреляций и наибольших парных корреляций соответственно.

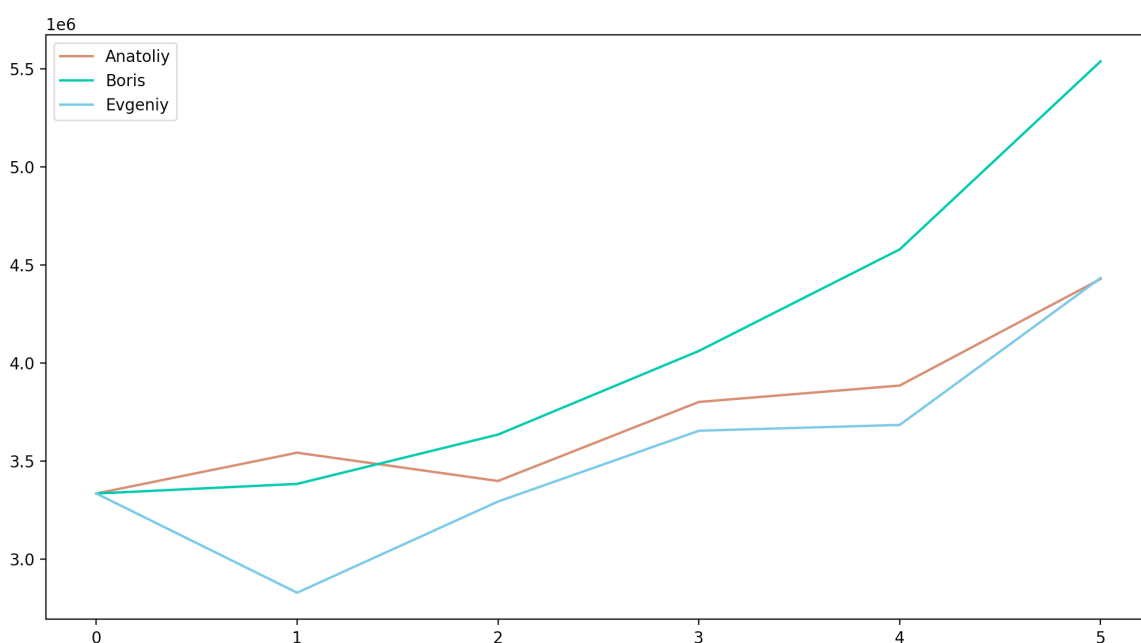


Рисунок 7 – График роста денежных средств (млн. рублей) по оси абсцисс

На рисунке 7 представлен график роста денежных средств каждого инвестора в зависимости от того, как было сделаны вложения с начала 2017 года до конца 2019 года. Как можно заметить, на графике Борис показал стабильный и наибольший рост денежных средств из-за своего портфеля вложений, логика которого была основана на вложении средств в 3 пары компаний каждые 6 месяцев с наибольшей корреляцией и растущим трендом.

У Евгения, по большей части, все завичит от рынка в целом. Из-за хорошей диверсификации акции будут тоже постоянно расти, уравнивая друг

друга. Но можно подметить, что на второй итерации продаж акций, денежные средства Евгения понизились. Опять таки, все зависит от рынка в целом. Но Анатолий, у которого средства были вложены в 3 парные компании с наименьшими коэффициентами корреляции показали хороший рост.