

Работу выполнили: Исоков А.А., Квашнин К.А., Манин Д.И

Вариант А.

1. Используя Pandas, проанализировать данные из объявлений о продажах автомобилей на сайте Avito.

```
import pandas as pd
from bs4 import BeautifulSoup
import requests as req
import matplotlib.pyplot as plt
```

```
def is_car(title):
    s = "цена"
    if (str(title).rfind(s) != -1): # если найдена подстрока "цена"
        return title # возвращаем всю строку
```

```
def get_car(title):
    k = str(title).find(',') # позиция первого вхождения запятой
    s = title[0:k] # выделение подстроки с названием
    return s # возвращение этой подстроки
```

```
def get_price(title):
    k = title.rfind('цена') # позиция вхождения подстроки "цена"
    l = title.rfind('руб') - 1 # позиция вхождения подстроки "руб"
    s1 = title[k + 5:l] # выделение подстроки с ценой с учётом пробелов
    return s1 # возвращение этой подстроки
```

```
def get_year(title):
    k = title.find(',') # позиция первого вхождения запятой
    s = title[k:len(title)] # выделение всей подстроки после запятой
    s1 = s[2:6] # выделение подстроки длиной 4 символа (год)
    return s1 # возвращение этой подстроки
```

```
def get_city(title):
    k = title.rfind('Автомобили в') # позиция вхождения подстроки "Автомобили в"
    s = title[k + 11:len(title)] # выделение подстроки со сдвигом на 11 до конца (город)
    return s # возвращение этой подстроки
```

```
df = pd.DataFrame({'марка': [], 'цена': [], 'год выпуска': [], 'город': []}) # создание таблицы pandas
pages = 3 # количество анализируемых страниц
base_url = 'https://www.avito.ru/mordoviya/avtomobili?cd=1'
for i in range(pages): # цикл по просматриваемым страницам
    url = base_url.format(str(i)) # форматирование ссылки
    resp = req.get(url) # передача ссылки
    soup = BeautifulSoup(resp.text, "html.parser") # парсинг
    links = soup.findAll('a') #заголовки
    #print(links) # тест
```

```

for link in links:
    title = link.get('title') # все элементы
    # print(title) # вывод элементов
    if is_car(title): # если найдена подстрока "цена"
        df1 = pd.DataFrame( # добавляем в датафрейм
            {'марка': [get_car(title)], 'цена': [get_price(title)], 'год выпуска': [get_year(title)],
             'город': [get_city(title)]}) # распределение по столбцам
        df = df.append(df1, ignore_index=True) # добавить строку df1 в df

print(df) # вывод датафрейма
df.to_csv("avito_cars.csv") # создаем csv из Dataframe
df_by_city = df.groupby('город') # сортировка по городам
print(df_by_city.get_group("в Саранске")) # Машины в Саранске
df_by_model = df.groupby('марка') # сортировка по моделям
print(df_by_model.get_group("LADA Granta")) # Lada granta

```

Результат выполнения программы:

1. Общий вывод датафрейма:

	марка	цена	год выпуска	город
0	LADA Largus	500 000	2018	в Тургенево
1	Skoda Rapid	340 000	2014	в Саранске
2	BMW X3	1 200 000	2013	в Саранске
3	LADA Priora	277 000	2012	в Саранске
4	Toyota Land Cruiser Prado	1 250 000	2010	в Рузаевке
..	...	...	...	...
175	KIA Rio	765 000	2019	в Краснослободске
176	Renault Logan	510 000	2016	в Ардатове
177	LADA Granta	170 000	2015	в Саранске
178	LADA Kalina	130 000	2011	в Тургенево
179	Новый ГАЗ ГАЗель Next	1 670 000	2020	в Саранске

## 2. Вывод автомобилей в городе Саранск

```
[180 rows x 4 columns]
      марка      цена  год выпуска      город
1      Skoda Rapid  340 000      2014  в Саранске
2      BMW X3 1  200 000      2013  в Саранске
3      LADA Priora  277 000      2012  в Саранске
6      LADA Priora  395 000      2015  в Саранске
7  Renault Sandero Stepway  385 000      2012  в Саранске
..      ...      ...      ...      ...
170     Audi 80      70 000      1990  в Саранске
171     LADA Kalina  165 000      2010  в Саранске
174     BMW X1      800 000      2013  в Саранске
177     LADA Granta  170 000      2015  в Саранске
179  Новый ГАЗ ГАЗель Next  1 670 000      2020  в Саранске

[108 rows x 4 columns]
```

## 3. Вывод автомобилей по модели (LADA Granta)

```
[107 rows x 4 columns]
      марка      цена  год выпуска      город
44  LADA Granta  170 000      2015  в Саранске
116 LADA Granta  170 000      2015  в Саранске
176 LADA Granta  170 000      2015  в Саранске

Process finished with exit code 0
```

2. Проанализировать данные о курсах акций Amazon, Facebook, Google, Netflix и Apple за период с 2015 по 2020 год.

Код программы:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn
import random

seaborn.set_context('poster') # общий стиль графика (шрифты и т.д.)
#print(plt.style.available) # вывод доступных стилей графика
plt.style.use('fivethirtyeight') # Стиль графика
plt.rcParams['figure.figsize'] = (25, 15) # Размер изображения (*100
пикселей)

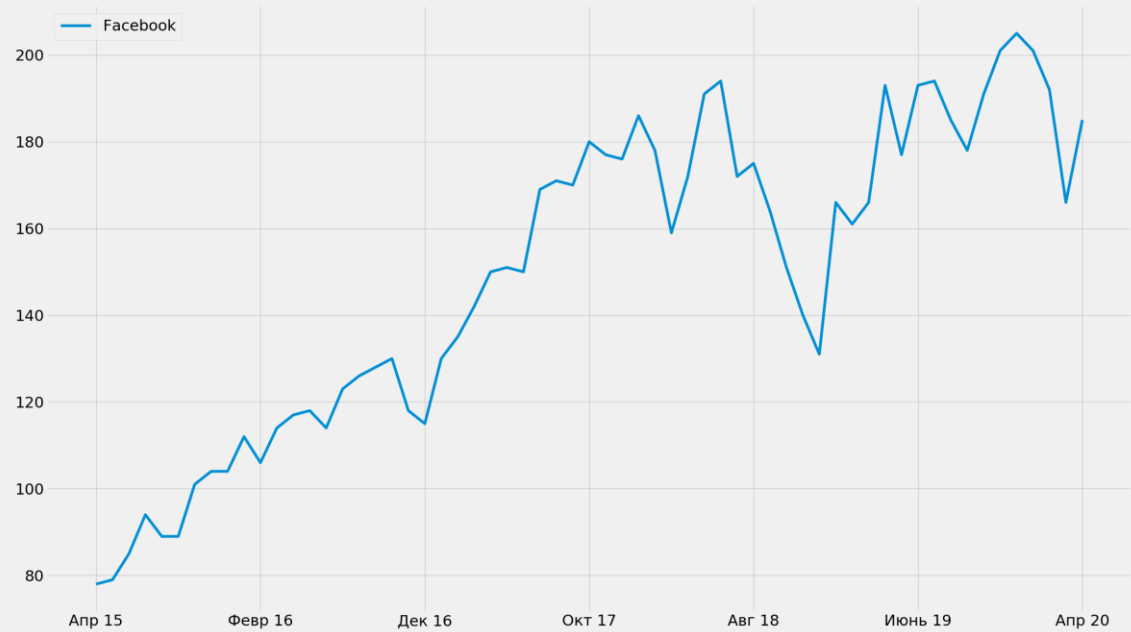
def graph(company):
    df = pd.read_csv(company+'.csv', sep=',') # считывание csv файла
    df = df.iloc[::-1] # обратный порядок
    #print(df) # вывод считанных данных
    df['Date'].plot(x="Date", y="Price", label=company,
color='C'+str(random.randint(0,50))) # построение графика: по оси x -- дата,
по оси y -- цена, Название -- Компания, Цвет -- CN палитра (где N
определяется случайным образом
    plt.legend(loc='upper left') # Легенда в левом верхнем углу
    #plt.show() # построение графика в отдельном окне
    path = 'Graphs' + '/' + company + '.png' # путь сохранения изображений
    plt.savefig(path) # сохранение
    plt.close('all') # Закрытие файла

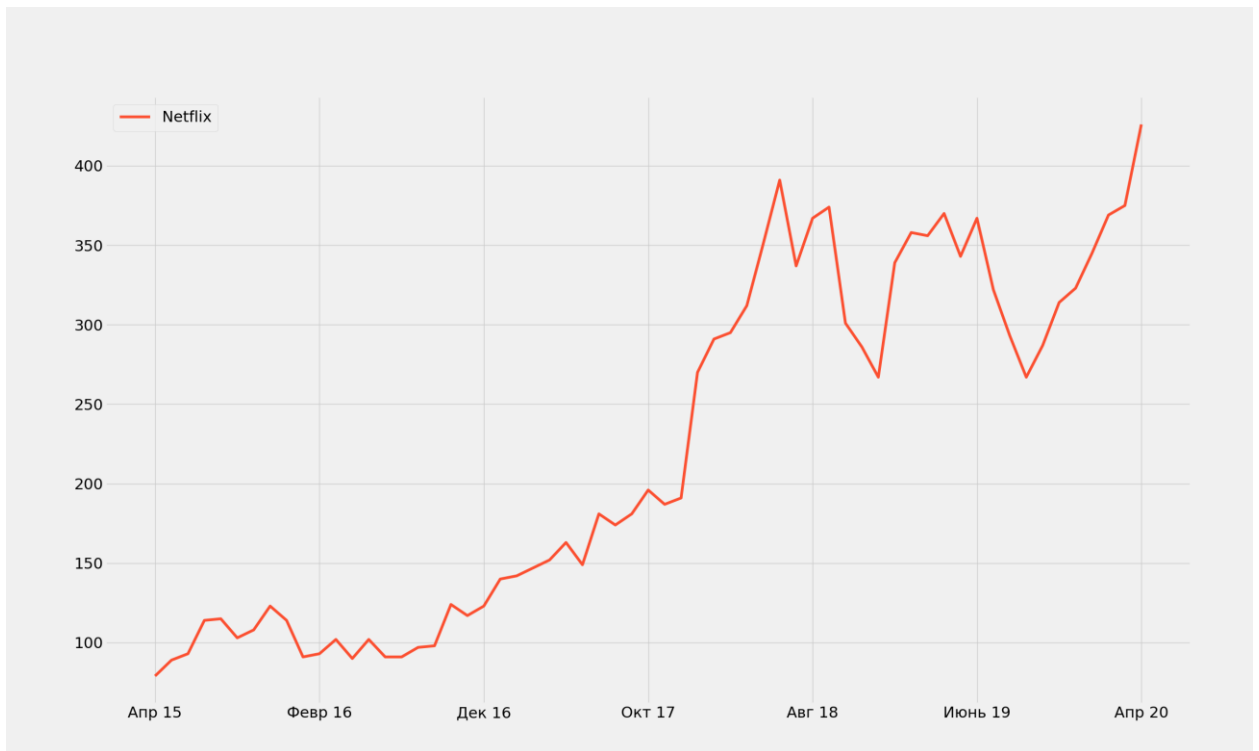
graph('Apple')
graph('Amazon')
graph('Facebook')
graph('Google')
graph('Netflix')
```

При неиспользовании функции plt.close('all') можем получить график со всеми данными.

Результат выполнения программы:







Сравнение всех компаний:

