РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра информационных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № **5**

Дисциплина: Методы машинного обучения

Студент: Шалыгин Георгий

Группа: НФИ-02

Москва **2023**

Вариант № **15**

1. При помощи модуля **pandas_datareader** считайте котировки указанной в индивидуальным задании акции за указанный период времени**.**

Считайте котировки акции с указанным ниже тикером за **2018-2019** год**:**

KO Coca-Cola Company

In [ ]:

```
#!pip install yfinance
```

In [1]:

```
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import pandas as pd
import tensorflow as tf
tf.__version__
```

Out[1]:

```
'2.12.0'
```

In [2]:

```
from pandas_datareader import data as pdr
import yfinance as yfin
import datetime as dt

yfin.pdr_override()
```

In [3]:

```
aapl = pdr.get_data_yahoo('KO',
                          start=dt.datetime(2018, 1, 1),
                          end=dt.datetime(2019, 1, 1))
aapl.head()
```

```
[**********100%**********]  1 of 1 completed
```

Out[3]:

Date	Open	High	Low	Close	Adj Close	Volume
2018-01-02	45.910000	45.939999	45.509998	45.540001	38.548206	10872200
2018-01-03	45.490002	45.689999	45.340000	45.439999	38.463558	12635600
2018-01-04	45.560001	46.220001	45.450001	46.080002	39.005295	12709400
2018-01-05	46.020000	46.200001	45.790001	46.070000	38.996834	13113100
2018-01-08	45.950001	46.099998	45.880001	46.000000	38.937584	7068600

In [4]:

```
aapl.tail()
```

Out[4]:

Date	Open	High	Low	Close	Adj Close	Volume
2018-12-24	47.619999	47.869999	45.830002	45.959999	40.249268	10733700
2018-12-26	45.880001	46.959999	45.660000	46.939999	41.107498	14342600
2018-12-27	46.740002	47.549999	46.029999	47.529999	41.624199	16966500
2018-12-28	47.889999	48.009998	47.029999	47.200001	41.335190	13218200
2018-12-31	47.490002	47.540001	46.959999	47.349998	41.466545	10576300

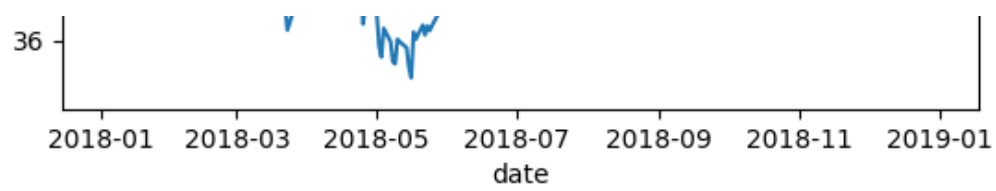
1. Визуализируйте котировки акции (столбец `Adj Close`) за весь период на графике. Подпишите оси и рисунок.

In [6]:

```
data = aapl['Adj Close']
plt.plot(data)
plt.xlabel('date')
plt.ylabel('Adj Close')
plt.title('Котировки акций cocacola')
```

Out[6]:

```
Text(0.5, 1.0, 'Котировки акций cocacola')
```



1. Вычислите и визуализируйте заданный показатель акции в соответствии с индивидуальным заданием.

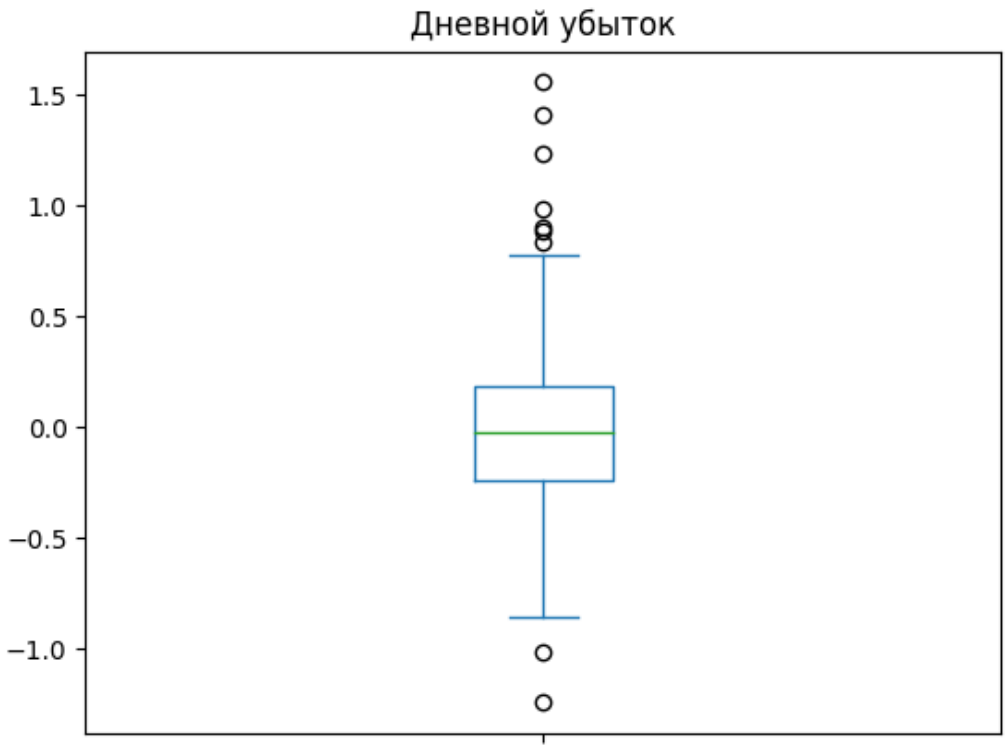
Показатель акции: дневной убыток, способ визуализации – диаграмма размаха

In [7]:

```
diff = np.append(np.array(data), 0) - np.insert(np.array(data), 0, [0])
diff *= -1
diff = diff[1:-1]
pd.Series(diff).plot.box(title='Дневной убыток')
```

Out[7]:

```
<Axes: title={'center': 'Дневной убыток'}>
```



1. Сформируйте обучающую, тестовую и валидационные выборки для обучения нейронной сети в соответствии с индивидуальным заданием.

   Прогнозирование стоимости акции через **12** дней по данным за предыдущие **25** дней.

In [8]:

```
adj = aapl['Adj Close']
raw_data = aapl.drop(['Adj Close'], axis=1)
```

In [9]:

```
num_train_samples = int(0.5 * len(raw_data))
num_val_samples = int(0.25 * len(raw_data))
num_test_samples = len(raw_data) - num_train_samples - num_val_samples
print("num_train_samples:", num_train_samples)
print("num_val_samples:", num_val_samples)
print("num_test_samples:", num_test_samples)
```

```
num_train_samples: 125
num_val_samples: 62
num_test_samples: 64
```

In [10]:

```
# mean = raw_data[:num_train_samples].mean(axis=0)
# raw_data -= mean
# std = raw_data[:num_train_samples].std(axis=0)
# raw_data /= std # whole raw_data normalized w.r.t. first num_train_samples rows

# mean.shape, std.shape
```

In [11]:

```
sampling_rate = 1
sequence_length = 25
delay = 25 + 12 - 1
batch_size = 256

train_dataset = tf.keras.utils.timeseries_dataset_from_array(
    raw_data[:-delay],
    targets=adj[delay:],
    sampling_rate=sampling_rate,
    sequence_length=sequence_length,
    shuffle=True,
    batch_size=batch_size,
    start_index=0,
    end_index=num_train_samples)

val_dataset = tf.keras.utils.timeseries_dataset_from_array(
    raw_data[:-delay],
    targets=adj[delay:],
    sampling_rate=sampling_rate,
    sequence_length=sequence_length,
    shuffle=True,
    batch_size=batch_size,
    start_index=num_train_samples,
    end_index=num_train_samples + num_val_samples)

test_dataset = tf.keras.utils.timeseries_dataset_from_array(
    raw_data[:-delay],
    targets=adj[delay:],
    sampling_rate=sampling_rate,
    sequence_length=sequence_length,
    shuffle=True,
    batch_size=batch_size,
    start_index=num_train_samples + num_val_samples)
```

1. Постройте нейронную сеть **MLP** с нормализующим слоем и одним плотным скрытым слоем из **16** нейронов для прогнозирования стоимости акции и обучите ее на обучающей выборке**.** Оцените качество прогнозирования при помощи заданного показателя качества на тестовой выборке**.**

   Показатель качества **MAE**

In [12]:

```
df_normalizer = tf.keras.layers.Normalization()
df_normalizer.adapt(raw_data)
print(df_normalizer.mean.numpy())
print(df_normalizer.variance.numpy())
```

```
[[4.5413548e+01 4.5724541e+01 4.5099438e+01 4.5409920e+01 1.2525482e+07]]
[[4.4421110e+00 4.5318952e+00 4.3046637e+00 4.4931951e+00 1.7014095e+13]]
```

In [90]:

```
inputs = tf.keras.Input(shape=(sequence_length, raw_data.shape[-1]))
x = df_normalizer(inputs)
x = tf.keras.layers.Flatten()(x)
x = tf.keras.layers.Dense(16, activation="relu")(x)
outputs = tf.keras.layers.Dense(1)(x)
model = tf.keras.Model(inputs, outputs)
```

```
model.summary()
model.compile(optimizer=tf.keras.optimizers.Adam(0.1), loss="mse", metrics=["mae"])
history = model.fit(train_dataset,
                    epochs=50,
                    validation_data=val_dataset,
                   )

print(f"Test MAE: {model.evaluate(test_dataset)[1]:.2f}")
```

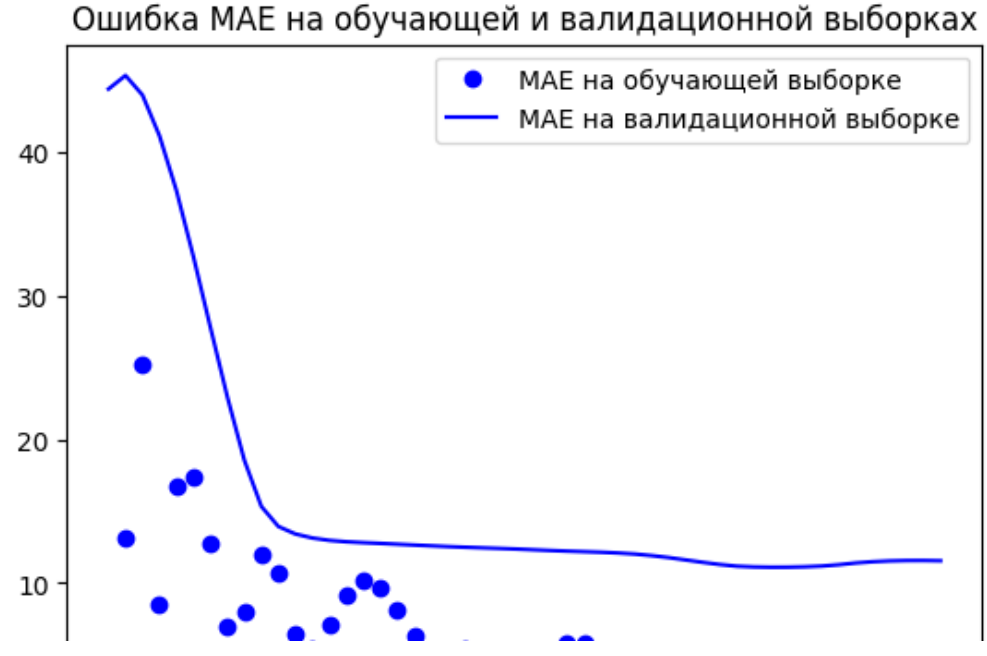
```
Model: "model_18"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_20 (InputLayer)       [(None, 25, 5)]           0

 normalization (Normalizatio  (None, 25, 5)            11
 n)

 flatten_10 (Flatten)        (None, 125)               0

 dense_41 (Dense)            (None, 16)                2016

 dense_42 (Dense)            (None, 1)                 17

=================================================================
Total params: 2,044
Trainable params: 2,033
Non-trainable params: 11
_________________________________________________________________
Epoch 1/50
1/1 [==============================] - 2s 2s/step - loss: 1334.8717 - mae: 36.5151 - val_
loss: 1972.3865 - val_mae: 44.3707
Epoch 2/50
1/1 [==============================] - 0s 120ms/step - loss: 318.1675 - mae: 13.1325 - va
l_loss: 2055.7756 - val_mae: 45.3129
Epoch 3/50
1/1 [==============================] - 0s 105ms/step - loss: 844.2757 - mae: 25.2650 - va
l_loss: 1933.4203 - val_mae: 43.9533
Epoch 4/50
1/1 [==============================] - 0s 95ms/step - loss: 135.1945 - mae: 8.5364 - val_
loss: 1694.0490 - val_mae: 41.1207
Epoch 5/50
1/1 [==============================] - 0s 90ms/step - loss: 329.9633 - mae: 16.7738 - val
_loss: 1399.2942 - val_mae: 37.3062
Epoch 6/50
1/1 [==============================] - 0s 84ms/step - loss: 366.5819 - mae: 17.4335 - val
_loss: 1086.6844 - val_mae: 32.7129
Epoch 7/50
1/1 [==============================] - 0s 122ms/step - loss: 189.1449 - mae: 12.7430 - va
l_loss: 803.0599 - val_mae: 27.7929
Epoch 8/50
1/1 [==============================] - 0s 95ms/step - loss: 65.8306 - mae: 7.0155 - val_l
oss: 575.3167 - val_mae: 22.9462
Epoch 9/50
1/1 [==============================] - 0s 121ms/step - loss: 113.3880 - mae: 8.0222 - val
_loss: 413.0074 - val_mae: 18.5405
Epoch 10/50
1/1 [==============================] - 0s 95ms/step - loss: 205.4850 - mae: 11.9650 - val
_loss: 310.3462 - val_mae: 15.3304
Epoch 11/50
1/1 [==============================] - 0s 105ms/step - loss: 176.5858 - mae: 10.6961 - va
l_loss: 252.3762 - val_mae: 13.9358
Epoch 12/50
1/1 [==============================] - 0s 94ms/step - loss: 85.3985 - mae: 6.4980 - val_l
oss: 223.1028 - val_mae: 13.4129
Epoch 13/50
1/1 [==============================] - 0s 93ms/step - loss: 43.5760 - mae: 5.4193 - val_l
oss: 209.7725 - val_mae: 13.1315
Epoch 14/50
1/1 [==============================] - 0s 124ms/step - loss: 64.1764 - mae: 7.1268 - val_
loss: 203.8006 - val_mae: 12.9653
```

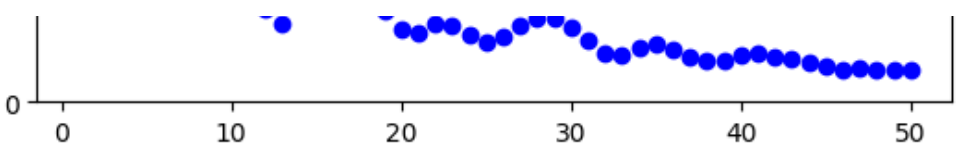
```
Epoch 15/50
1/1 [==============================] - 0s 93ms/step - loss: 102.6104 - mae: 9.1618 - val_
loss: 201.2509 - val_mae: 12.8690
Epoch 16/50
1/1 [==============================] - 0s 123ms/step - loss: 121.0824 - mae: 10.1281 - va
l_loss: 200.1558 - val_mae: 12.8077
Epoch 17/50
1/1 [==============================] - 0s 124ms/step - loss: 109.4977 - mae: 9.6054 - val
_loss: 199.5119 - val_mae: 12.7557
Epoch 18/50
1/1 [==============================] - 0s 87ms/step - loss: 78.3373 - mae: 8.1312 - val_l
oss: 198.8014 - val_mae: 12.6995
Epoch 19/50
1/1 [==============================] - 0s 93ms/step - loss: 48.1694 - mae: 6.3166 - val_l
oss: 198.0558 - val_mae: 12.6415
Epoch 20/50
1/1 [==============================] - 0s 92ms/step - loss: 37.3624 - mae: 4.9898 - val_l
oss: 197.3218 - val_mae: 12.5873
Epoch 21/50
1/1 [==============================] - 0s 119ms/step - loss: 47.9954 - mae: 4.7737 - val_
loss: 196.6581 - val_mae: 12.5339
Epoch 22/50
1/1 [==============================] - 0s 94ms/step - loss: 61.9531 - mae: 5.4495 - val_l
oss: 196.2396 - val_mae: 12.4887
Epoch 23/50
1/1 [==============================] - 0s 93ms/step - loss: 60.0364 - mae: 5.3529 - val_l
oss: 195.8798 - val_mae: 12.4471
Epoch 24/50
1/1 [==============================] - 0s 94ms/step - loss: 43.7506 - mae: 4.6633 - val_l
oss: 195.5004 - val_mae: 12.4070
Epoch 25/50
1/1 [==============================] - 0s 101ms/step - loss: 29.4107 - mae: 4.1691 - val_
loss: 195.0046 - val_mae: 12.3643
Epoch 26/50
1/1 [==============================] - 0s 119ms/step - loss: 27.7436 - mae: 4.5448 - val_
loss: 194.3437 - val_mae: 12.3130
Epoch 27/50
1/1 [==============================] - 0s 121ms/step - loss: 35.8326 - mae: 5.2880 - val_
loss: 193.8077 - val_mae: 12.2627
Epoch 28/50
1/1 [==============================] - 0s 91ms/step - loss: 43.9220 - mae: 5.7715 - val_l
oss: 193.5300 - val_mae: 12.2180
Epoch 29/50
1/1 [==============================] - 0s 121ms/step - loss: 44.5981 - mae: 5.7500 - val_
loss: 193.5132 - val_mae: 12.1799
Epoch 30/50
1/1 [==============================] - 0s 92ms/step - loss: 36.7468 - mae: 5.1480 - val_l
oss: 193.4114 - val_mae: 12.1410
Epoch 31/50
1/1 [==============================] - 0s 98ms/step - loss: 25.6113 - mae: 4.2200 - val_l
oss: 193.1207 - val_mae: 12.0885
Epoch 32/50
1/1 [==============================] - 0s 107ms/step - loss: 18.3118 - mae: 3.3330 - val_
loss: 192.5210 - val_mae: 12.0159
Epoch 33/50
1/1 [==============================] - 0s 117ms/step - loss: 18.6390 - mae: 3.2276 - val_
loss: 191.2669 - val_mae: 11.9047
Epoch 34/50
1/1 [==============================] - 0s 95ms/step - loss: 24.2878 - mae: 3.7162 - val_l
oss: 189.3012 - val_mae: 11.7674
Epoch 35/50
1/1 [==============================] - 0s 130ms/step - loss: 27.3551 - mae: 3.9692 - val_
loss: 187.0116 - val_mae: 11.6020
Epoch 36/50
1/1 [==============================] - 0s 120ms/step - loss: 23.6998 - mae: 3.6882 - val_
loss: 185.5043 - val_mae: 11.4361
Epoch 37/50
1/1 [==============================] - 0s 120ms/step - loss: 16.9093 - mae: 3.1068 - val_
loss: 185.2414 - val_mae: 11.2861
Epoch 38/50
1/1 [==============================] - 0s 92ms/step - loss: 12.6653 - mae: 2.8030 - val_l
oss: 185.9741 - val_mae: 11.1683
```

```
Epoch 39/50
1/1 [==============================] - 0s 102ms/step - loss: 13.2012 - mae: 2.8835 - val_
loss: 187.5298 - val_mae: 11.1186
Epoch 40/50
1/1 [==============================] - 0s 88ms/step - loss: 15.5102 - mae: 3.1885 - val_l
oss: 189.0164 - val_mae: 11.0992
Epoch 41/50
1/1 [==============================] - 0s 125ms/step - loss: 15.9993 - mae: 3.3233 - val_
loss: 189.9563 - val_mae: 11.1026
Epoch 42/50
1/1 [==============================] - 0s 126ms/step - loss: 14.6822 - mae: 3.1743 - val_
loss: 190.8377 - val_mae: 11.1273
Epoch 43/50
1/1 [==============================] - 0s 157ms/step - loss: 13.6943 - mae: 2.9645 - val_
loss: 192.5115 - val_mae: 11.1775
Epoch 44/50
1/1 [==============================] - 0s 247ms/step - loss: 12.6497 - mae: 2.7866 - val_
loss: 195.7293 - val_mae: 11.2703
Epoch 45/50
1/1 [==============================] - 0s 143ms/step - loss: 10.0800 - mae: 2.4572 - val_
loss: 200.0051 - val_mae: 11.3925
Epoch 46/50
1/1 [==============================] - 0s 145ms/step - loss: 8.4988 - mae: 2.2727 - val_l
oss: 202.6202 - val_mae: 11.4802
Epoch 47/50
1/1 [==============================] - 0s 146ms/step - loss: 8.5948 - mae: 2.2928 - val_l
oss: 203.6814 - val_mae: 11.5357
Epoch 48/50
1/1 [==============================] - 0s 149ms/step - loss: 8.4931 - mae: 2.2733 - val_l
oss: 203.5324 - val_mae: 11.5622
Epoch 49/50
1/1 [==============================] - 0s 144ms/step - loss: 8.1204 - mae: 2.2115 - val_l
oss: 202.6252 - val_mae: 11.5646
Epoch 50/50
1/1 [==============================] - 0s 111ms/step - loss: 7.5052 - mae: 2.1564 - val_l
oss: 201.3511 - val_mae: 11.5470
1/1 [==============================] - 0s 35ms/step - loss: 281.2939 - mae: 16.2223
Test MAE: 16.22
```

In [91]:

```
loss = history.history["mae"]
val_loss = history.history["val_mae"]
epochs = range(1, len(loss) + 1)
plt.figure()
plt.plot(epochs[1:], loss[1:], "bo", label="MAE на обучающей выборке")
plt.plot(epochs, val_loss, "b", label="MAE на валидационной выборке")
plt.title("Ошибка MAE на обучающей и валидационной выборках")
plt.legend()
plt.show()
```





1. Примените указанную в индивидуальном задании технику решения проблемы исчезающих градиентов и постройте нейронную сеть **MLP** с нормализующим слоем и тремя плотными скрытыми слоями из **16** нейронов для прогнозирования стоимости акции и обучите ее на обучающей выборке. Оцените качество прогнозирования при помощи заданного показателя качества для тестовой выборки.

   Техника борьбы с исчезающими градиентами: Функции активации без насыщения

In [92]:

```
inputs2 = tf.keras.Input(shape=(sequence_length, raw_data.shape[-1]))
x2 = tf.keras.layers.Flatten()(inputs2)
x2 = tf.keras.layers.Dense(16, activation="selu")(x2)
x2 = tf.keras.layers.Dense(16, activation="selu")(x2)
x2 = tf.keras.layers.Dense(16, activation="selu")(x2)
outputs2 = tf.keras.layers.Dense(1)(x2)
model2 = tf.keras.Model(inputs2, outputs2)
model2.summary()

model2.compile(optimizer=tf.keras.optimizers.Adam(0.08), loss="mse", metrics=["mae"])
history2 = model2.fit(train_dataset,
                      epochs=50,
                      validation_data=val_dataset,
                     )

print(f"Test MAE: {model2.evaluate(test_dataset)[1]:.2f}")
```

```
Model: "model_19"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_21 (InputLayer)       [(None, 25, 5)]           0

 flatten_11 (Flatten)        (None, 125)               0

 dense_43 (Dense)            (None, 16)                2016

 dense_44 (Dense)            (None, 16)                272

 dense_45 (Dense)            (None, 16)                272

 dense_46 (Dense)            (None, 1)                 17

=================================================================
Total params: 2,577
Trainable params: 2,577
Non-trainable params: 0
_________________________________________________________________
Epoch 1/50
1/1 [==============================] - 1s 1s/step - loss: 9409820360704.0000 - mae: 29656
57.0000 - val_loss: 64973191512064.0000 - val_mae: 8050021.0000
Epoch 2/50
1/1 [==============================] - 0s 118ms/step - loss: 103185146445824.0000 - mae:
10098205.0000 - val_loss: 291651518464.0000 - val_mae: 525309.1875
Epoch 3/50
1/1 [==============================] - 0s 92ms/step - loss: 468475969536.0000 - mae: 6697
30.9375 - val_loss: 326845530112.0000 - val_mae: 561682.2500
Epoch 4/50
1/1 [==============================] - 0s 123ms/step - loss: 515617030144.0000 - mae: 704
072.6250 - val_loss: 145038160.0000 - val_mae: 3300.7517
Epoch 5/50
1/1 [==============================] - 0s 97ms/step - loss: 164604080.0000 - mae: 3091.02
42 - val_loss: 1391.9651 - val_mae: 37.3050
Epoch 6/50
1/1 [==============================] - 0s 89ms/step - loss: 1213.0656 - mae: 34.8196 - va
l_loss: 1328.1663 - val_mae: 36.4399
```

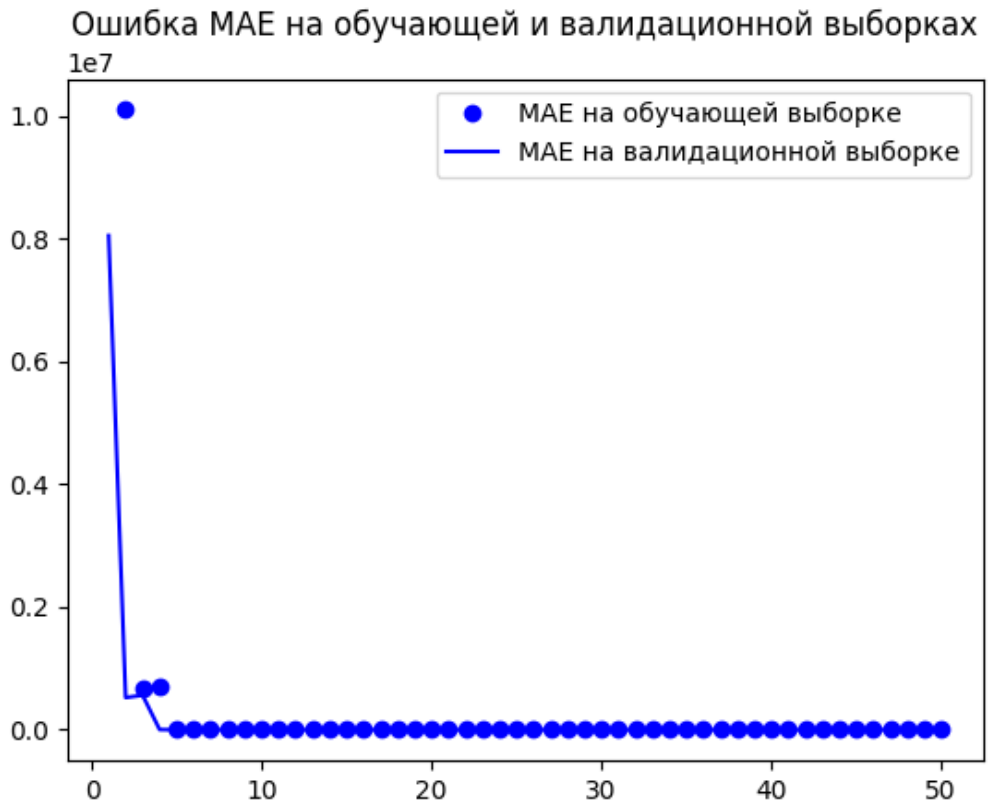
```
l_loss: 1328.1665 - val_mae: 36.4399
Epoch 7/50
1/1 [==============================] - 0s 95ms/step - loss: 1153.5673 - mae: 33.9545 - va
l_loss: 1310.5618 - val_mae: 36.1976
Epoch 8/50
1/1 [==============================] - 0s 129ms/step - loss: 1137.1676 - mae: 33.7122 - v
al_loss: 1304.2726 - val_mae: 36.1106
Epoch 9/50
1/1 [==============================] - 0s 100ms/step - loss: 1131.3107 - mae: 33.6252 - v
al_loss: 1297.2444 - val_mae: 36.0131
Epoch 10/50
1/1 [==============================] - 0s 116ms/step - loss: 1124.7668 - mae: 33.5277 - v
al_loss: 1285.5486 - val_mae: 35.8504
Epoch 11/50
1/1 [==============================] - 0s 97ms/step - loss: 1113.8800 - mae: 33.3650 - va
l_loss: 1261.8826 - val_mae: 35.5188
Epoch 12/50
1/1 [==============================] - 0s 120ms/step - loss: 1091.8624 - mae: 33.0334 - v
al_loss: 1229.4293 - val_mae: 35.0590
Epoch 13/50
1/1 [==============================] - 0s 119ms/step - loss: 1061.6947 - mae: 32.5736 - v
al_loss: 1188.0408 - val_mae: 34.4636
Epoch 14/50
1/1 [==============================] - 0s 88ms/step - loss: 1023.2653 - mae: 31.9782 - va
l_loss: 1135.8031 - val_mae: 33.6972
Epoch 15/50
1/1 [==============================] - 0s 119ms/step - loss: 974.8373 - mae: 31.2118 - va
l_loss: 1073.1824 - val_mae: 32.7549
Epoch 16/50
1/1 [==============================] - 0s 93ms/step - loss: 916.9008 - mae: 30.2695 - val
_loss: 998.3251 - val_mae: 31.5915
Epoch 17/50
1/1 [==============================] - 0s 97ms/step - loss: 847.8263 - mae: 29.1062 - val
_loss: 907.0637 - val_mae: 30.1125
Epoch 18/50
1/1 [==============================] - 0s 94ms/step - loss: 763.9167 - mae: 27.6271 - val
_loss: 810.2496 - val_mae: 28.4596
Epoch 19/50
1/1 [==============================] - 0s 117ms/step - loss: 675.3188 - mae: 25.9742 - va
l_loss: 708.1492 - val_mae: 26.6055
Epoch 20/50
1/1 [==============================] - 0s 121ms/step - loss: 582.4352 - mae: 24.1201 - va
l_loss: 597.4759 - val_mae: 24.4372
Epoch 21/50
1/1 [==============================] - 0s 121ms/step - loss: 482.5398 - mae: 21.9518 - va
l_loss: 488.4710 - val_mae: 22.0946
Epoch 22/50
1/1 [==============================] - 0s 116ms/step - loss: 385.1794 - mae: 19.6092 - va
l_loss: 384.9294 - val_mae: 19.6120
Epoch 23/50
1/1 [==============================] - 0s 130ms/step - loss: 293.9783 - mae: 17.1266 - va
l_loss: 287.7568 - val_mae: 16.9546
Epoch 24/50
1/1 [==============================] - 0s 90ms/step - loss: 210.0151 - mae: 14.4692 - val
_loss: 200.9949 - val_mae: 14.1667
Epoch 25/50
1/1 [==============================] - 0s 129ms/step - loss: 137.1111 - mae: 11.6813 - va
l_loss: 127.6983 - val_mae: 11.2871
Epoch 26/50
1/1 [==============================] - 0s 92ms/step - loss: 78.1283 - mae: 8.8017 - val_l
oss: 70.2199 - val_mae: 8.3619
Epoch 27/50
1/1 [==============================] - 0s 88ms/step - loss: 35.1907 - mae: 5.8765 - val_l
oss: 29.9517 - val_mae: 5.4454
Epoch 28/50
1/1 [==============================] - 0s 94ms/step - loss: 9.4195 - mae: 2.9600 - val_lo
ss: 7.0660 - val_mae: 2.6013
Epoch 29/50
1/1 [==============================] - 0s 89ms/step - loss: 0.6711 - mae: 0.6687 - val_lo
ss: 0.3087 - val_mae: 0.4420
Epoch 30/50
1/1 [==============================] - 0s 117ms/step - loss: 7.3347 - mae: 2.5840 - val_l
oss: 6.9429 - val_mae: 2.5776
```

```
oss: 6.9429 - val_mae: 2.5776
Epoch 31/50
1/1 [==============================] - 0s 95ms/step - loss: 26.2913 - mae: 5.0630 - val_l
oss: 22.9541 - val_mae: 4.7597
Epoch 32/50
1/1 [==============================] - 0s 121ms/step - loss: 53.1496 - mae: 7.2451 - val_
loss: 43.5774 - val_mae: 6.5786
Epoch 33/50
1/1 [==============================] - 0s 96ms/step - loss: 82.8142 - mae: 9.0640 - val_l
oss: 64.0658 - val_mae: 7.9854
Epoch 34/50
1/1 [==============================] - 0s 118ms/step - loss: 110.2953 - mae: 10.4708 - va
l_loss: 80.4741 - val_mae: 8.9541
Epoch 35/50
1/1 [==============================] - 0s 91ms/step - loss: 131.5186 - mae: 11.4394 - val
_loss: 90.2068 - val_mae: 9.4820
Epoch 36/50
1/1 [==============================] - 0s 122ms/step - loss: 143.8755 - mae: 11.9674 - va
l_loss: 92.2030 - val_mae: 9.5867
Epoch 37/50
1/1 [==============================] - 0s 117ms/step - loss: 146.3920 - mae: 12.0720 - va
l_loss: 86.7999 - val_mae: 9.3006
Epoch 38/50
1/1 [==============================] - 0s 93ms/step - loss: 139.5670 - mae: 11.7860 - val
_loss: 75.4022 - val_mae: 8.6662
Epoch 39/50
1/1 [==============================] - 0s 99ms/step - loss: 125.0158 - mae: 11.1516 - val
_loss: 60.0852 - val_mae: 7.7322
Epoch 40/50
1/1 [==============================] - 0s 124ms/step - loss: 105.0558 - mae: 10.2175 - va
l_loss: 43.2083 - val_mae: 6.5505
Epoch 41/50
1/1 [==============================] - 0s 97ms/step - loss: 82.3053 - mae: 9.0359 - val_l
oss: 27.0848 - val_mae: 5.1755
Epoch 42/50
1/1 [==============================] - 0s 121ms/step - loss: 59.3470 - mae: 7.6609 - val_
loss: 13.7919 - val_mae: 3.6733
Epoch 43/50
1/1 [==============================] - 0s 93ms/step - loss: 38.5867 - mae: 6.1587 - val_l
oss: 4.7475 - val_mae: 2.1091
Epoch 44/50
1/1 [==============================] - 0s 116ms/step - loss: 21.7674 - mae: 4.5945 - val_
loss: 0.5717 - val_mae: 0.5958
Epoch 45/50
1/1 [==============================] - 0s 93ms/step - loss: 9.7031 - mae: 3.0076 - val_lo
ss: 1.3709 - val_mae: 1.0481
Epoch 46/50
1/1 [==============================] - 0s 92ms/step - loss: 2.7604 - mae: 1.4658 - val_lo
ss: 6.6165 - val_mae: 2.5135
Epoch 47/50
1/1 [==============================] - 0s 119ms/step - loss: 0.6584 - mae: 0.6583 - val_l
oss: 15.2496 - val_mae: 3.8666
Epoch 48/50
1/1 [==============================] - 0s 90ms/step - loss: 2.5654 - mae: 1.3930 - val_lo
ss: 25.8399 - val_mae: 5.0538
Epoch 49/50
1/1 [==============================] - 0s 103ms/step - loss: 7.2544 - mae: 2.5684 - val_l
oss: 36.4113 - val_mae: 6.0094
Epoch 50/50
1/1 [==============================] - 0s 94ms/step - loss: 13.0760 - mae: 3.5240 - val_l
oss: 46.3430 - val_mae: 6.7856
1/1 [==============================] - 0s 37ms/step - loss: 101.9375 - mae: 10.0793
Test MAE: 10.08
```

In [93]:

```
loss = history2.history["mae"]
val_loss = history2.history["val_mae"]
epochs = range(1, len(loss) + 1)
plt.figure()
plt.plot(epochs[1:], loss[1:], "bo", label="MAE на обучающей выборке")
plt.plot(epochs, val_loss, "b", label="MAE на валидационной выборке")
```

```
plt.title("Ошибка MAE на обучающей и валидационной выборках")
plt.legend()
plt.show()
```



1. Постройте рекуррентную нейронную сеть с нормализующим слоем и одним скрытым слоем **LSTM** из **16** нейронов для прогнозирования стоимости акции и обучите ее на обучающей выборке**.** Оцените качество прогнозирования при помощи заданного показателя качества на тестовой выборке**.**

In [94]:

```
inputs3 = tf.keras.Input(shape=(sequence_length, raw_data.shape[-1]))
x3 = df_normalizer(inputs3)
x3 = tf.keras.layers.LSTM(16)(x3)
outputs3 = tf.keras.layers.Dense(1)(x3)
model3 = tf.keras.Model(inputs3, outputs3)

model3.summary()
model3.compile(optimizer=tf.keras.optimizers.Adam(0.06), loss="mse", metrics=["mae"])
history3 = model3.fit(train_dataset,
                      epochs=50,
                      validation_data=val_dataset,
                     )

print(f"Test MAE: {model3.evaluate(test_dataset)[1]:.2f}")
```

```
Model: "model_20"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_22 (InputLayer)       [(None, 25, 5)]           0

 normalization (Normalizatio  (None, 25, 5)            11
 n)

 lstm_10 (LSTM)              (None, 16)                1408

 dense_47 (Dense)            (None, 1)                 17

=================================================================
Total params: 1,436
Trainable params: 1,425
Non-trainable params: 11
```

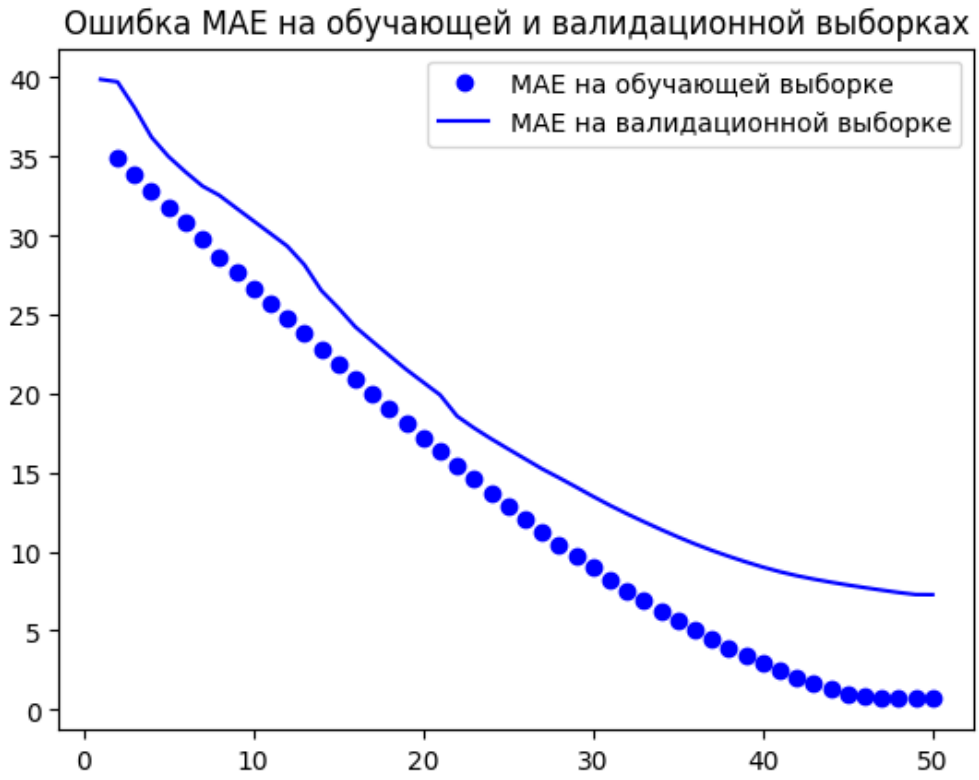
```
_________________________________________________________________
Epoch 1/50
1/1 [==============================] - 4s 4s/step - loss: 1367.2759 - mae: 36.9678 - val_
loss: 1590.6273 - val_mae: 39.8801
Epoch 2/50
1/1 [==============================] - 0s 133ms/step - loss: 1221.8074 - mae: 34.9429 - v
al_loss: 1578.7134 - val_mae: 39.7256
Epoch 3/50
1/1 [==============================] - 0s 108ms/step - loss: 1145.3317 - mae: 33.8305 - v
al_loss: 1454.3834 - val_mae: 38.0916
Epoch 4/50
1/1 [==============================] - 0s 152ms/step - loss: 1078.2831 - mae: 32.8248 - v
al_loss: 1318.6726 - val_mae: 36.2270
Epoch 5/50
1/1 [==============================] - 0s 126ms/step - loss: 1013.3594 - mae: 31.8207 - v
al_loss: 1233.3102 - val_mae: 34.9992
Epoch 6/50
1/1 [==============================] - 0s 133ms/step - loss: 949.2454 - mae: 30.7969 - va
l_loss: 1168.3997 - val_mae: 34.0276
Epoch 7/50
1/1 [==============================] - 0s 98ms/step - loss: 884.5147 - mae: 29.7275 - val
_loss: 1109.8286 - val_mae: 33.1306
Epoch 8/50
1/1 [==============================] - 0s 130ms/step - loss: 821.5508 - mae: 28.6490 - va
l_loss: 1074.9391 - val_mae: 32.5403
Epoch 9/50
1/1 [==============================] - 0s 104ms/step - loss: 764.4670 - mae: 27.6349 - va
l_loss: 1025.9694 - val_mae: 31.7310
Epoch 10/50
1/1 [==============================] - 0s 110ms/step - loss: 711.4966 - mae: 26.6595 - va
l_loss: 978.9792 - val_mae: 30.9299
Epoch 11/50
1/1 [==============================] - 0s 128ms/step - loss: 661.2656 - mae: 25.7003 - va
l_loss: 933.4806 - val_mae: 30.1296
Epoch 12/50
1/1 [==============================] - 0s 135ms/step - loss: 613.1251 - mae: 24.7463 - va
l_loss: 889.0944 - val_mae: 29.3247
Epoch 13/50
1/1 [==============================] - 0s 134ms/step - loss: 566.8693 - mae: 23.7935 - va
l_loss: 822.1046 - val_mae: 28.1565
Epoch 14/50
1/1 [==============================] - 0s 99ms/step - loss: 522.4836 - mae: 22.8419 - val
_loss: 728.8238 - val_mae: 26.5222
Epoch 15/50
1/1 [==============================] - 0s 127ms/step - loss: 480.0139 - mae: 21.8927 - va
l_loss: 671.1282 - val_mae: 25.4225
Epoch 16/50
1/1 [==============================] - 0s 127ms/step - loss: 439.5151 - mae: 20.9475 - va
l_loss: 610.9203 - val_mae: 24.2240
Epoch 17/50
1/1 [==============================] - 0s 126ms/step - loss: 401.0334 - mae: 20.0081 - va
l_loss: 568.8328 - val_mae: 23.3102
Epoch 18/50
1/1 [==============================] - 0s 127ms/step - loss: 364.6005 - mae: 19.0760 - va
l_loss: 528.9741 - val_mae: 22.4129
Epoch 19/50
1/1 [==============================] - 0s 108ms/step - loss: 330.2315 - mae: 18.1529 - va
l_loss: 491.0657 - val_mae: 21.5265
Epoch 20/50
1/1 [==============================] - 0s 144ms/step - loss: 297.9276 - mae: 17.2403 - va
l_loss: 457.6025 - val_mae: 20.7155
Epoch 21/50
1/1 [==============================] - 0s 134ms/step - loss: 267.6768 - mae: 16.3395 - va
l_loss: 425.1559 - val_mae: 19.9098
Epoch 22/50
1/1 [==============================] - 0s 102ms/step - loss: 239.4564 - mae: 15.4518 - va
l_loss: 363.3733 - val_mae: 18.5662
Epoch 23/50
1/1 [==============================] - 0s 105ms/step - loss: 213.2332 - mae: 14.5787 - va
l_loss: 336.7281 - val_mae: 17.8174
Epoch 24/50
1/1 [==============================] - 0s 105ms/step - loss: 188.9651 - mae: 13.7212 - va
l_loss: 314.1085 - val_mae: 17.1359
```

```
Epoch 25/50
1/1 [==============================] - 0s 143ms/step - loss: 166.6019 - mae: 12.8805 - va
l_loss: 293.7347 - val_mae: 16.4961
Epoch 26/50
1/1 [==============================] - 0s 99ms/step - loss: 146.0859 - mae: 12.0579 - val
_loss: 274.8857 - val_mae: 15.8647
Epoch 27/50
1/1 [==============================] - 0s 131ms/step - loss: 127.3526 - mae: 11.2543 - va
l_loss: 256.9485 - val_mae: 15.2339
Epoch 28/50
1/1 [==============================] - 0s 130ms/step - loss: 110.3317 - mae: 10.4709 - va
l_loss: 242.0242 - val_mae: 14.6694
Epoch 29/50
1/1 [==============================] - 0s 110ms/step - loss: 94.9475 - mae: 9.7086 - val_
loss: 227.1814 - val_mae: 14.0856
Epoch 30/50
1/1 [==============================] - 0s 134ms/step - loss: 81.1203 - mae: 8.9682 - val_
loss: 212.4329 - val_mae: 13.4913
Epoch 31/50
1/1 [==============================] - 0s 119ms/step - loss: 68.7663 - mae: 8.2508 - val_
loss: 199.1187 - val_mae: 12.9224
Epoch 32/50
1/1 [==============================] - 0s 104ms/step - loss: 57.7991 - mae: 7.5571 - val_
loss: 187.5295 - val_mae: 12.3883
Epoch 33/50
1/1 [==============================] - 0s 125ms/step - loss: 48.1301 - mae: 6.8877 - val_
loss: 177.3641 - val_mae: 11.8833
Epoch 34/50
1/1 [==============================] - 0s 112ms/step - loss: 39.6695 - mae: 6.2434 - val_
loss: 168.0582 - val_mae: 11.3956
Epoch 35/50
1/1 [==============================] - 0s 106ms/step - loss: 32.3268 - mae: 5.6247 - val_
loss: 159.4341 - val_mae: 10.9249
Epoch 36/50
1/1 [==============================] - 0s 104ms/step - loss: 26.0119 - mae: 5.0322 - val_
loss: 151.7893 - val_mae: 10.4815
Epoch 37/50
1/1 [==============================] - 0s 102ms/step - loss: 20.6357 - mae: 4.4661 - val_
loss: 145.3031 - val_mae: 10.0721
Epoch 38/50
1/1 [==============================] - 0s 147ms/step - loss: 16.1104 - mae: 3.9270 - val_
loss: 139.9293 - val_mae: 9.6956
Epoch 39/50
1/1 [==============================] - 0s 102ms/step - loss: 12.3508 - mae: 3.4149 - val_
loss: 135.5469 - val_mae: 9.3431
Epoch 40/50
1/1 [==============================] - 0s 112ms/step - loss: 9.2746 - mae: 2.9300 - val_l
oss: 132.2924 - val_mae: 9.0150
Epoch 41/50
1/1 [==============================] - 0s 133ms/step - loss: 6.8028 - mae: 2.4725 - val_l
oss: 130.5797 - val_mae: 8.7228
Epoch 42/50
1/1 [==============================] - 0s 128ms/step - loss: 4.8603 - mae: 2.0422 - val_l
oss: 130.4935 - val_mae: 8.4712
Epoch 43/50
1/1 [==============================] - 0s 113ms/step - loss: 3.3763 - mae: 1.6401 - val_l
oss: 131.4650 - val_mae: 8.2538
Epoch 44/50
1/1 [==============================] - 0s 111ms/step - loss: 2.2847 - mae: 1.2884 - val_l
oss: 132.7609 - val_mae: 8.0598
Epoch 45/50
1/1 [==============================] - 0s 112ms/step - loss: 1.5240 - mae: 1.0232 - val_l
oss: 133.9584 - val_mae: 7.8822
Epoch 46/50
1/1 [==============================] - 0s 107ms/step - loss: 1.0381 - mae: 0.8430 - val_l
oss: 134.8031 - val_mae: 7.7156
Epoch 47/50
1/1 [==============================] - 0s 100ms/step - loss: 0.7756 - mae: 0.7238 - val_l
oss: 135.2086 - val_mae: 7.5569
Epoch 48/50
1/1 [==============================] - 0s 129ms/step - loss: 0.6906 - mae: 0.6768 - val_l
oss: 135.2697 - val_mae: 7.4066
```

```
Epoch 49/50
1/1 [==============================] - 0s 101ms/step - loss: 0.7421 - mae: 0.6984 - val_l
oss: 135.7575 - val_mae: 7.2791
Epoch 50/50
1/1 [==============================] - 0s 102ms/step - loss: 0.8940 - mae: 0.7600 - val_l
oss: 142.0625 - val_mae: 7.2705
1/1 [==============================] - 0s 39ms/step - loss: 26.7526 - mae: 5.1405
Test MAE: 5.14
```

In [95]:

```
loss = history3.history["mae"]
val_loss = history3.history["val_mae"]
epochs = range(1, len(loss) + 1)
plt.figure()
plt.plot(epochs[1:], loss[1:], "bo", label="MAE на обучающей выборке")
plt.plot(epochs, val_loss, "b", label="MAE на валидационной выборке")
plt.title("Ошибка MAE на обучающей и валидационной выборках")
plt.legend()
plt.show()
```

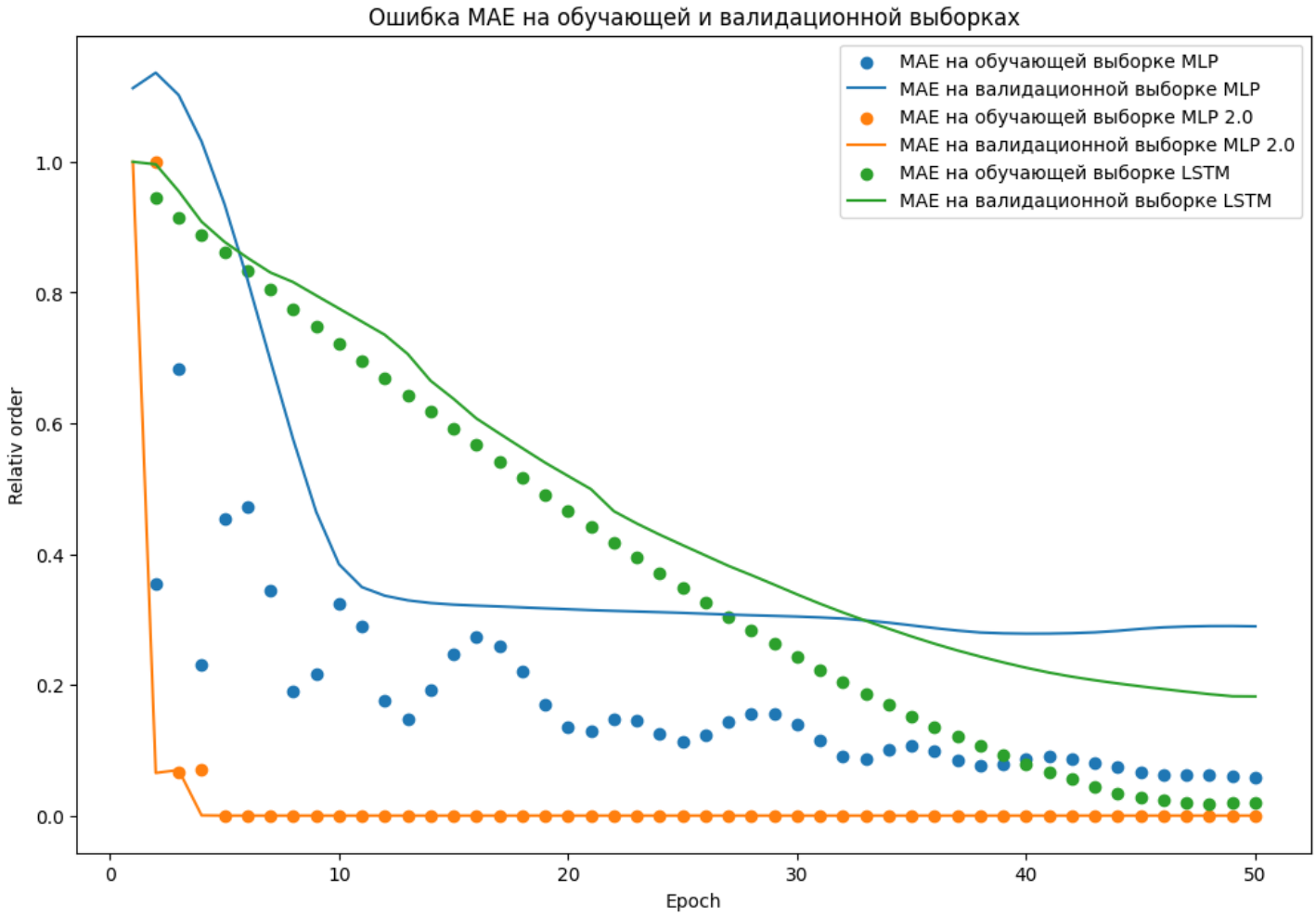


1. Визуализируйте кривые обучения для трех построенных моделей на одном рисунке в зависимости от эпохи обучения, подписывая оси и рисунок и создавая легенду. Используйте для визуализации относительную ошибку (ошибку обучения, деленную на начальную ошибку на первой эпохе).

In [98]:

```
loss1 = np.array(history.history["mae"]) / history3.history["mae"][0]
val_loss1 = np.array(history.history["val_mae"]) / history3.history["val_mae"][0]
loss2 = np.array(history2.history["mae"]) / history2.history["mae"][1]
val_loss2 = np.array(history2.history["val_mae"]) / history2.history["val_mae"][0]
loss3 = np.array(history3.history["mae"]) / history3.history["mae"][0]
val_loss3 = np.array(history3.history["val_mae"]) / history3.history["val_mae"][0]
epochs = range(1, len(loss1) + 1)
plt.figure(figsize=(12,8))
plt.scatter(epochs[1:], loss1[1:], label="MAE на обучающей выборке MLP")
plt.plot(epochs, val_loss1,  label="MAE на валидационной выборке MLP")

plt.scatter(epochs[1:], loss2[1:],  label="MAE на обучающей выборке MLP 2.0")
plt.plot(epochs, val_loss2,  label="MAE на валидационной выборке MLP 2.0")

plt.scatter(epochs[1:], loss3[1:], label="MAE на обучающей выборке LSTM")
plt.plot(epochs, val_loss3, label="MAE на валидационной выборке LSTM")
```

```
plt.xlabel('Epoch')
plt.ylabel('Relativ order')
plt.title("Ошибка MAE на обучающей и валидационной выборках")
plt.legend()
plt.show()
```



1. Визуализируйте весь набор данных и прогнозы трех построенных моделей для обучающей и тестовой выборок на одном рисунке **(**ось **X –** даты**,** ось **Y –** стоимость акции**),** подписывая оси и рисунок и создавая легенду**.**

In [99]:

```
real = np.array(aapl['Adj Close'])[-191:]
```

In [100]:

```
real.shape
```

Out[100]:

```
(191,)
```

In [101]:

```
all_data = tf.keras.utils.timeseries_dataset_from_array(
    raw_data[:-delay],
    targets=adj[delay:],
    sampling_rate=sampling_rate,
    sequence_length=sequence_length,
    shuffle=True,
    batch_size=1000,
    start_index=0)
```

In [102]:

```
for a, b in all_data:
```

```
  samples = a
  targ = b
```

In [103]:

```
y1 = model.predict(samples)
y2 = model2.predict(samples)
y3 = model3.predict(samples)
```

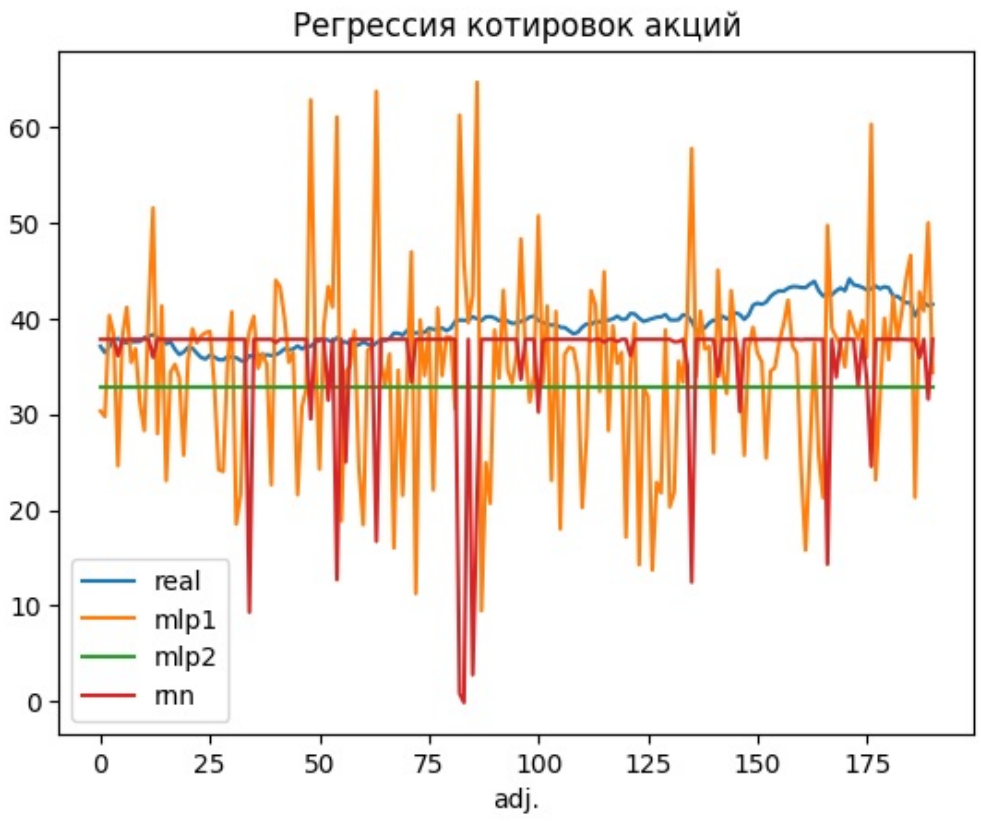
```
6/6 [==============================] - 0s 3ms/step
6/6 [==============================] - 0s 3ms/step
6/6 [==============================] - 0s 4ms/step
```

In [104]:

```
plt.plot(real, label='real')
plt.plot(y1, label='mlp1')
plt.plot(y2, label='mlp2')
plt.plot(y3, label='rnn')
plt.xlabel('day')
plt.xlabel('adj.')
plt.title('Регрессия котировок акций')
plt.legend()
```

Out[104]:

```
<matplotlib.legend.Legend at 0x7f84af7c1420>
```



In [ ]: