**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №6

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-35Б |  |  |
| Талаев А.П. |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2024 г

**Задание:**

Программа, которая подсчитывает выгоду арбитража между биржами MEXC и Binance, запрашивая торговую пару и размер покупки-продажи.

**Выполнение:**  
import requests

def get\_price\_mexc(symbol):

url = f'https://www.mexc.com/open/api/v2/market/ticker?symbol={symbol}'

response = requests.get(url)

if response.status\_code != 200:

raise ValueError(f"Ошибка при запросе к MEXC: {response.status\_code}")

data = response.json()

print("Ответ от MEXC:", data)

if 'data' in data and len(data['data']) > 0:

return float(data['data'][0]['last'])

else:

raise ValueError("Не удалось получить данные о ценах с MEXC.")

def get\_price\_binance(symbol):

url = f'https://api.binance.com/api/v3/ticker/price?symbol={symbol}'

response = requests.get(url)

if response.status\_code != 200:

raise ValueError(f"Ошибка при запросе к Binance: {response.status\_code}")

data = response.json()

print("Ответ от Binance:", data)

if 'price' in data:

return float(data['price'])

else:

raise ValueError("Не удалось получить данные о ценах с Binance.")

def calculate\_arbitrage\_profit(buy\_price, sell\_price, amount):

total\_buy\_cost = buy\_price \* amount

total\_sell\_revenue = sell\_price \* amount

profit = total\_sell\_revenue - total\_buy\_cost

return profit

def main():

trading\_pairs = {

'BTC/USDT': {

'mexc': 'BTC\_USDT',

'binance': 'BTCUSDT'

},

'ETH/USDT': {

'mexc': 'ETH\_USDT',

'binance': 'ETHUSDT'

},

}

print("Доступные торговые пары:")

for pair in trading\_pairs.keys():

print(pair)

chosen\_pair = input("Выберите торговую пару (например, BTC/USDT): ")

if chosen\_pair not in trading\_pairs:

print("Неверная торговая пара.")

return

symbol\_mexc = trading\_pairs[chosen\_pair]['mexc']

symbol\_binance = trading\_pairs[chosen\_pair]['binance']

try:

buy\_price\_mexc = get\_price\_mexc(symbol\_mexc)

sell\_price\_binance = get\_price\_binance(symbol\_binance)

print(f"Цена покупки на MEXC: {buy\_price\_mexc:.2f} USDT")

print(f"Цена продажи на Binance: {sell\_price\_binance:.2f} USDT")

amount = float(input("Введите количество актива для торговли: "))

profit = calculate\_arbitrage\_profit(buy\_price\_mexc, sell\_price\_binance, amount)

if profit > 0:

print(f"Потенциальная прибыль от арбитража: {profit:.2f} USDT")

else:

print("Арбитраж невыгоден.")

except ValueError as e:

print(e)

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Результаты:**

