Министерство науки и высшего образования Российской Федерации

Федерального государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ**

Специальность: 09.02.07 Информационные системы и программирование

Квалификация: Программист

ПРАКТИЧЕСКИЕ РАБОТЫ

ПО «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

Листов: 10

|  |  |
| --- | --- |
| Выполнила студент | Проверил преподаватель |
| Группы П50-4-22 | \_\_\_\_\_\_\_\_\_\_\_\_Т.Д. Артамонова |
| Юрченко Вячеслав Валерьевич | «\_\_\_» \_\_\_\_\_\_\_\_2024 года |

Москва 2024

ПРАКТИЧЕСКАЯ РАБОТА №3

***Цель работы:*** развернуть частную сеть блокчейна с помощью Geth. Необходимо реализовать:

* Создать 5 аккаунтов
* Создать генезис-блок
* Развернуть частную сеть блокчейна с использованием Geth
* Сделать деплой смарт-контракта (агентство недвижимости) в сети
* Подключить питон к сети и вывести баланс всех аккаунтов

***Ход работы:***

1. Необходимо создать текстовый файл, в котором будут находиться основные команды для работы с сетью блокчейна. Требуется создать текстовый файл с любым названием и вставить туда команды из [notion шпаргалки](https://superficial-spleen-608.notion.site/Geth-6edd48ad8693414e92d14a1a13b5b254).

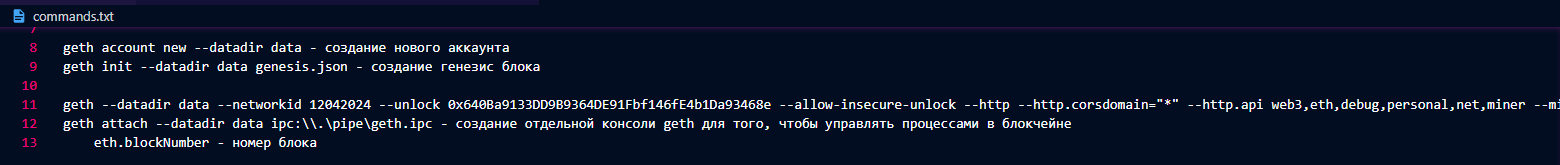


Рисунок 1 – команды для работы с сетью блокчейна

1. Создать файл «**genesis**» с расширением **json**, куда необходимо вставить основу с [официального сайта](https://geth.ethereum.org/docs/fundamentals/private-network#creating-genesis-block). Указать любой **chainId**, который удобен. В адресе **extradata** необходимо найти адрес первого аккаунта из основы, и затем заменить на наш первый адрес. После в **alloc** прописать адреса без **0x** и баланс аккаунта.



Рисунок 2 – genesis.json

1. С помощью команды **geth account new --datadir data** создать пять аккаунтов и записать их в файл с командами, чтобы не забыть их адреса.

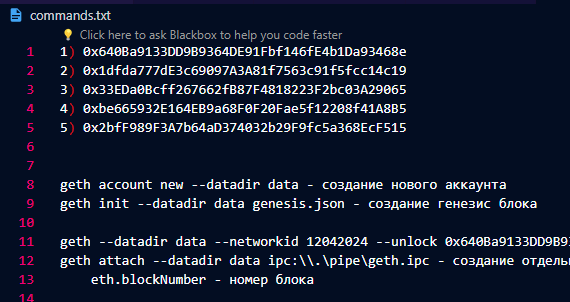


Рисунок 3 – сохранение адресов

1. После необходимо запустить новый терминал и прописать команду для создания генезис-блока - **geth init --datadir data genesis.json.** После этого создастся в текущей директории папка **data** где будут разные файлы, которые нам пока не нужны.
2. Теперь требуется запустить частную сеть блок чейна, используя длинную команду, где до **–networkid** требуется указать наш **chainId** а два адреса, существущих в этой команде, заменить на первый существующий у нас. В итоге команда такая: **geth --datadir data --networkid 12042024 --unlock 0x640Ba9133DD9B9364DE91Fbf146fE4b1Da93468e --allow-insecure-unlock --http --http.corsdomain="\*" --http.api web3,eth,debug,personal,net,miner --miner.etherbase 0x640Ba9133DD9B9364DE91Fbf146fE4b1Da93468e**

После требуется ввести пароль и подтвердить его для разблокировки аккаунта.

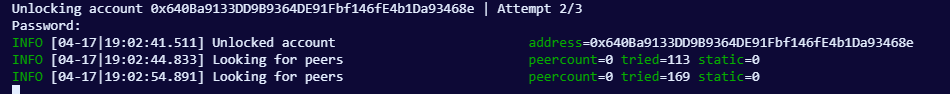


Рисунок 4 – разблокировка аккаунта

1. Теперь нужно запустить майнер с помощью команды: **geth attach --datadir data ipc:\\.\pipe\geth.ipc.**

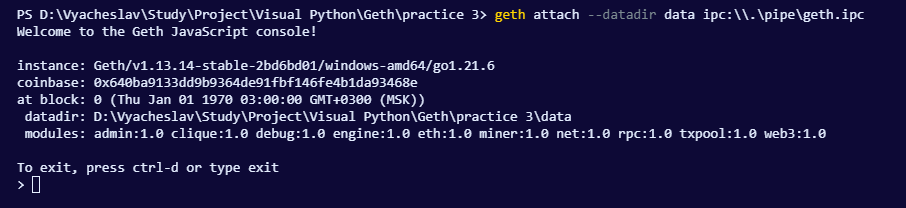
**

Рисунок 5 – запуск данных для майнера

После ввода этой команды нужно прописать **miner.start(),** где в результате должно выдать значение **null.**

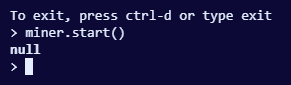
****

Рисунок 6 – запуск майнера

1. Затем нужно перейти на [Remix Ethereum](https://remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=paris&version=soljson-v0.8.25+commit.b61c2a91.js). После необходимо проверить в аккаунты. Для этого нужно перейти в “Solidity compiler”, открыть вкладку “Advanced Configurations” и в поле “EVM VERSION” выставить “paris”.

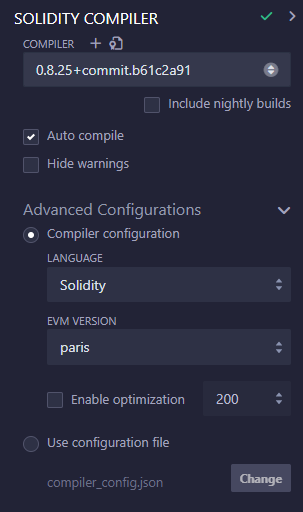


Рисунок 7 – проверка аккаунтов

Далее нужно перейти во вкладку “Deploy & run transactions” и в поле “Environment” выбрать “Custom - External Http Provider”, а после просто подключиться. Потом будут показаны пять аккаунтов которые были созданы ранее.

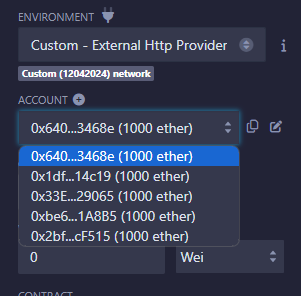


Рисунок 8 – аккаунты

1. Чтобы приступить к следующему пункту необходимо в терминале прописать **pip install web3**. Необходимо создать файл с названием и расширением **contract\_info.py**. После необходимо вставить **abi** смарт-контракта. Для того чтобы скопировать **abi** смарт-контракта необходимо нажать на **Solidity compiler**, пролистать ниже и найти кнопку с надписью **ABI** и скопировать. Далее в ранее созданном файле нужно вставить скопированный **abi** в код после строчки **abi**.

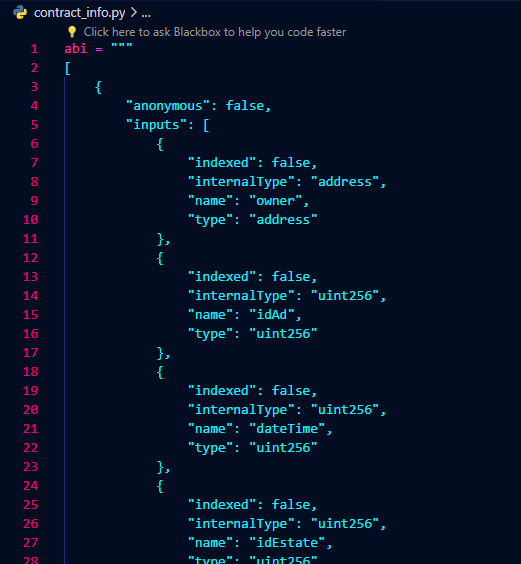


Рисунок 9 – содержание contract\_info.py (1)

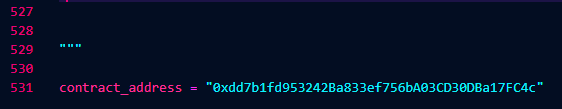


Рисунок 10 - содержание contract\_info.py (2)

1. Необходимо создать файл с название и расширением **main.py**. Далее требуется прописать всё так, как на скриншоте:

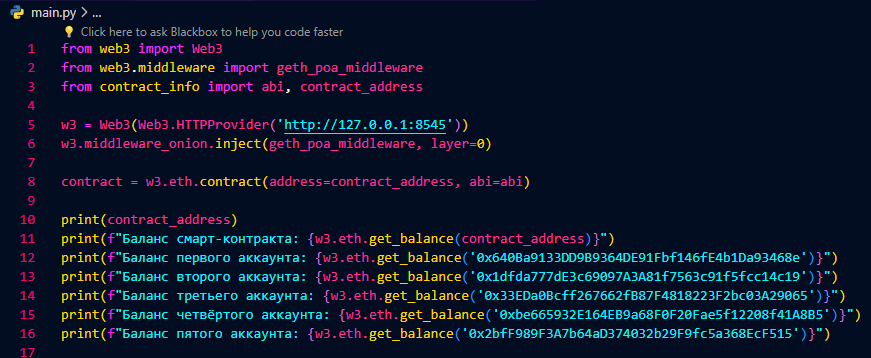


Рисунок 11 – содержание main.py

Для взаимодействия с сетью Ethereum необходимо установить middleware gethpoamiddleware, который предназначен для работы с сетями **Proof of Authority** (PoA), такими как Ganache. Затем нужно импортировать ABI (интерфейс контракта) и адрес смарт-контракта из модуля **contract\_info.py**.

Далее нужно создать объект Web3 и подключиться к локальному узлу Ethereum, который запущен на http://127.0.0.1:8545. Чтобы обеспечить корректную работу в сетях PoA, необходимо внедрить gethpoamiddleware в объект w3.

Следующим шагом является создание экземпляра смарт-контракта на основе его адреса и ABI. ABI описывает структуру и методы контракта, а адрес указывает его местоположение в блокчейне.

Затем нужно отобразить адрес смарт-контракта и запросить его баланс с помощью метода getbalance.

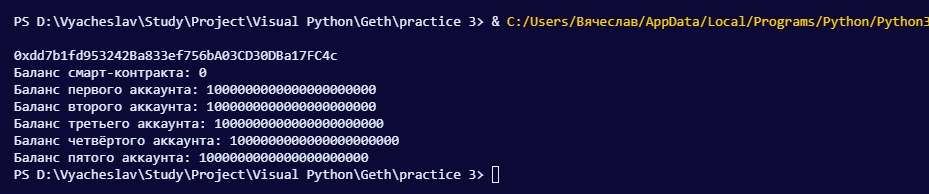


Рисунок 12 – итоговый вывод

***Вывод:*** После выполнения всех необходимых шагов была успешно развернута частная сеть блокчейна с помощью Geth. Были созданы 5 аккаунтов, сгенерирован генезис-блок и развернута сеть. В сети был развернут смарт-контракт, представляющий собой агентство недвижимости. На заключительном этапе была установлена связь между Python и сетью, после чего были выведены балансы всех аккаунтов.