**Инструкция по запуску блокчейн-приложения**

**1. Установка необходимых компонентов**

Перед запуском установите зависимости:

bash

pip install ecdsa flask requests

**2. Запуск приложения**

1. **Скопируйте полный код** в файл blockchain\_app.py
2. **Откройте два терминала** (для создания сети из 2 узлов)
3. **В первом терминале** запустите узел на порту 5000:

bash

python blockchain\_app.py 5000

1. **Во втором терминале** запустите узел на порту 5001:

bash

python blockchain\_app.py 5001

**3. Подключение узлов между собой**

1. В интерфейсе второго узла (5001):
   * В поле **"Connect to Node"** введите: localhost:5000
   * Нажмите кнопку **"Connect"**
2. Должно появиться сообщение об успешном подключении

**4. Работа с приложением**

**Интерфейс каждого узла содержит:**

* Публичный и приватный ключи вашего кошелька
* Текущий баланс
* Форму для отправки транзакций
* Кнопку для майнинга блоков
* Панель подключения к другим узлам
* Лог операций

**Пример рабочего процесса:**

1. На узле 5000:
   * Скопируйте публичный ключ из поля "Public Key"
   * В поле "Recipient" введите публичный ключ узла 5001
   * Укажите сумму
   * Нажмите "Send Transaction"
2. На любом узле нажмите "Mine Block" для подтверждения транзакций
3. Узлы автоматически синхронизируют свои блокчейны

**5. Дополнительные возможности**

* Для тестирования можно запускать более 2 узлов (используя разные порты)
* Все данные сохраняются в файл blockchain\_data.json
* При перезапуске программа сохраняет состояние блокчейна

**6. Важные примечания**

1. Для реального использования следует:
   * Добавить HTTPS-шифрование
   * Реализовать более надежную систему обнаружения узлов
   * Настроить параметры безопасности
2. Приватные ключи хранятся только в памяти приложения - для постоянного использования нужно реализовать их сохранение в защищенном хранилище.
3. Сложность майнинга намеренно установлена низкой (2 нуля в хеше) для тестирования - в реальной системе следует увеличить этот параметр.
4. Для запуска этого блокчейн-приложения вы можете использовать следующие терминалы в зависимости от вашей операционной системы:

**1. Windows:**

* **CMD (Командная строка)**
  + Откройте через поиск "cmd"
  + Простое и стандартное решение
* **Windows Terminal** (рекомендуется)
  + Скачайте из Microsoft Store
  + Поддержка вкладок, цветов и UTF-8
  + Можно запускать несколько узлов в разных вкладках
* **Git Bash**
  + Поставляется с Git for Windows
  + Поддерживает Linux-подобные команды

**2. macOS:**

* **Terminal** (стандартное приложение)
  + Находится в Finder → Программы → Утилиты
* **iTerm2** (рекомендуется)
  + Улучшенная альтернатива стандартному терминалу
  + Поддержка вкладок и разделения экрана

**3. Linux:**

* **GNOME Terminal** (Ubuntu/Debian)
* **Konsole** (KDE)
* **Terminator** (рекомендуется)
  + Поддержка разделения экрана

**Как запускать в терминале:**

1. Откройте **два отдельных терминала** (или две вкладки)
2. В каждом выполните:

bash

*# В первом терминале (порт 5000)*

python blockchain\_app.py 5000

*# Во втором терминале (порт 5001)*

python blockchain\_app.py 5001

**Советы:**

* Для удобства можете:
  + Разместить окна терминалов рядом (в Windows используйте Win+←/→)
  + В iTerm2/Terminator используйте разделение экрана
  + В Windows Terminal используйте несколько вкладок
* Если команда python не работает, попробуйте python3
* Для копирования/вставки в терминале:
  + Windows: Ctrl+C/Ctrl+V (в CMD) или Ctrl+Shift+C/Ctrl+Shift+V (в Windows Terminal)
  + macOS/Linux: Ctrl+Shift+C/Ctrl+Shift+V

Выбор терминала не влияет на работу приложения - используйте тот, который вам удобнее. Для новичков в Windows подойдет стандартный CMD или Windows Terminal.

**1. Установка Python (если ещё не установлен)**

1. Скачайте Python с [официального сайта](https://www.python.org/downloads/windows/)
2. При установке ОБЯЗАТЕЛЬНО отметьте галочку "Add Python to PATH"
3. Завершите установку

**2. Открытие командной строки**

Нажмите Win+R, введите cmd и нажмите Enter

**3. Установка необходимых библиотек**

В командной строке выполните:

cmd

pip install ecdsa flask requests

Если не работает, попробуйте:

cmd

python -m pip install ecdsa flask requests

**4. Создание файла с программой**

1. Откройте Блокнот
2. Скопируйте полный код из ответа выше
3. Сохраните файл как blockchain\_app.py (в поле "Тип файла" выберите "Все файлы")

**5. Запуск приложения**

Откройте ДВА окна командной строки (в каждом выполните отдельную команду):

**Первое окно (узел 1):**

cmd

python blockchain\_app.py 5000

**Второе окно (узел 2):**

cmd

python blockchain\_app.py 5001

**6. Подключение узлов**

В графическом интерфейсе второго узла (порт 5001):

1. В поле "Connect to Node" введите: localhost:5000
2. Нажмите кнопку "Connect"

**Возможные проблемы и решения:**

1. Если пишет "python не найден":
   * Переустановите Python с галочкой "Add to PATH"
   * Или используйте py вместо python в командах
2. Если не устанавливаются библиотеки:

Cmd

python -m pip install --upgrade pip

pip install ecdsa flask requests

1. Если появляются ошибки при запуске:
   * Убедитесь, что сохранили файл с расширением .py
   * Проверьте, что не удалили случайно часть кода при копировании
2. Для удобства можете использовать:
   * **VS Code** (с расширением Python)
   * **PyCharm Community Edition** (бесплатная версия)

Теперь у вас должна работать P2P-сеть блокчейна между двумя узлами на Windows!

**Как майнить блоки в этом блокчейн-приложении**

Вот пошаговая инструкция по майнингу:

**1. Запустите 2 узла (как описано ранее)**

bash

*# Терминал 1:*

python blockchain\_app.py 5000

*# Терминал 2:*

python blockchain\_app.py 5001

**2. Подключите узлы**

В интерфейсе узла 5001:

1. В поле **"Connect to Node"** введите: localhost:5000
2. Нажмите **"Connect"**

**3. Создайте транзакцию**

На любом узле:

1. Скопируйте **публичный ключ** получателя (из поля "Public Key" второго узла)
2. Вставьте в поле **"Recipient"**
3. Введите сумму
4. Нажмите **"Send Transaction"**

**4. Майнинг блока**

1. Нажмите кнопку **"Mine Block"** на любом узле
2. Программа:
   * Соберет все неподтвержденные транзакции
   * Добавит награду майнеру (10 монет)
   * Найдет хеш блока (с 2 нулями в начале)
   * Добавит блок в цепь

**5. Проверьте результат**

После майнинга:

* Баланс майнера увеличится на 10 монет
* Транзакции перейдут в статус подтвержденных
* Новый блок появится в логе

**Как это работает технически**

1. **Алгоритм майнинга** (упрощенный PoW):

python

while new\_block.hash[:2] != "00": *# Ищем хеш с 2 нулями*

new\_block.nonce += 1

new\_block.hash = new\_block.calculate\_hash()

1. **Награда майнеру**:

python

reward\_tx = Transaction("0", miner\_address, self.mining\_reward)

1. **Сложность** можно изменить, поменяв число нулей:

python

self.difficulty = 3 *# Теперь нужно 3 нуля в хеше*

**Советы**

* Для теста можно майнить на обоих узлах по очереди
* Все изменения автоматически синхронизируются между узлами
* Данные сохраняются в blockchain\_data.json

Если майнинг идет слишком медленно, уменьшите сложность (число нулей в хеше) в коде.