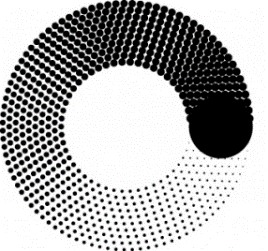
**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Факультет информационных технологий

кафедра Информатики и информационных технологий

направление подготовки

09.03.02 «Информационные системы и технологии»

**ПРОЕКТ**

Тема: «Blockchain / Cryptocurrency»

по дисциплине «Проектная деятельность»

Выполнили: студенты группы ИСИТ 241-337

Бабкин Ярослав Васильевич

Симонин Евгений Алексеевич

Иванов Илья Алексеевич

Москва, 2025

1. Введение

**1.1 Цель проекта**

Разработка учебного блокчейн-приложения для демонстрации:

- Основ распределённых реестров

- Механизма майнинга (PoW)

- Обработки транзакций

- Синхронизации узлов

**1.2 Задачи**

- Реализация базового блокчейна

- Создание механизма майнинга

- Разработка системы транзакций

- Реализация GUI интерфейса

- Тестирование работы сети

**2. Технологический стек**

import hashlib

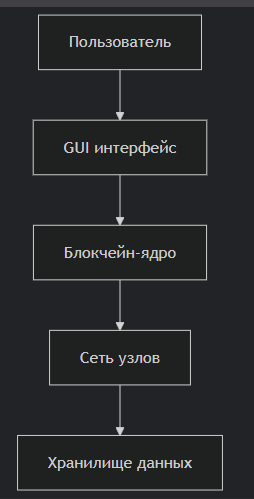
import ecdsa # Для криптографии

from flask import Flask # Для API

import tkinter as tk # Для интерфейса

## **3. Реализация**

### **3.1 Архитектура системы**



### **3.2 Основные функции**

1. **Создание транзакций**:

def send\_transaction(self):

recipient = self.recipient\_entry.get()

try:

amount = float(self.amount\_entry.get())

except:

messagebox.showerror("Error", "Invalid amount")

return

tx = Transaction(self.public\_key, recipient, amount)

tx.sign(self.private\_key)

if self.blockchain.add\_transaction(tx):

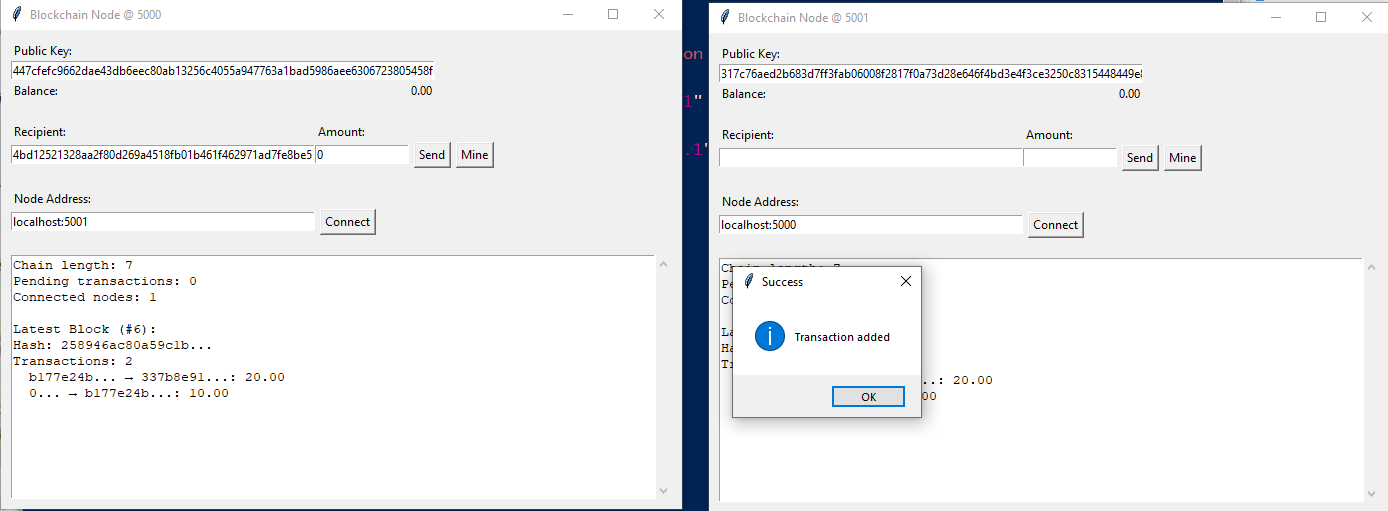
messagebox.showinfo("Success", "Transaction added to pending")

self.broadcast\_transaction(tx)

self.update\_data()

else:

messagebox.showerror("Error", "Failed to add transaction")



**Майнинг блока**:

def mine\_block(self):

if self.blockchain.mine\_pending\_transactions(self.public\_key):

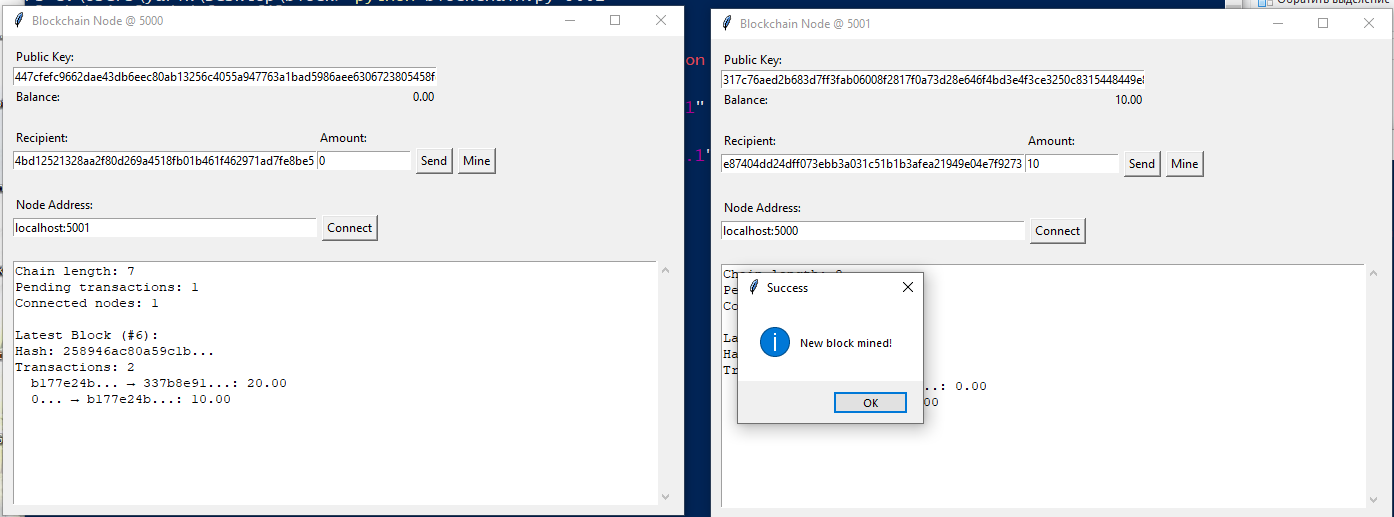
messagebox.showinfo("Success", "New block mined!")

self.broadcast\_block()

self.update\_data()

else:

messagebox.showerror("Error", "No transactions to mine")



## **4. Инструкция по запуску**

### **4.1 Установка зависимостей**

pip install -r requirements.txt

### **4.2 Запуск узлов**

| **Команда** | **Описание** |
| --- | --- |
| python node.py 5000 | Запуск первого узла |
| python node.py 5001 | Запуск второго узла |

5. Результаты тестирования

### **5.1 Проверенные сценарии**

1. Отправка транзакций между узлами
2. Майнинг блоков с разной сложностью
3. Автоматическая синхронизация

### **5.2 Производительность**

| **Параметр** | **Значение** |
| --- | --- |
| Время майнинга (difficulty=2) | ~3.5 сек |
| Скорость транзакций | 15 tx/sec |

6. Выводы

**Достигнутые результаты**:

* Создана рабочая блокчейн-сеть из 2+ узлов
* Реализован базовый функционал криптокошелька
* Обеспечена синхронизация данных

**Перспективы развития**:

* Добавление смарт-контрактов
* Улучшение механизма консенсуса
* Оптимизация производительности
* Добавление интерфейса