

**Федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Факультет информационных технологий
Кафедра «Информатика и вычислительная техника»**

Направление подготовки/ специальность: системная и программная инженерия

ИТОГОВЫЙ ОТЧЕТ

по второй части проектной практики

Студент: Ланщекhov Егор Алексеевич Группа: 241-326

Место прохождения практики: Московский Политех, кафедра «Информатика
и вычислительная техника»

Отчет принят с оценкой _____ Дата _____

Руководитель практики: Чернова Вера Михайловна, кафедра "Информатика
и вычислительная техника"

Москва 2025

Оглавление	
ВВЕДЕНИЕ	3
ИНДИВИДУАЛЬНЫЙ ПЛАН	4
Этапы выполнения практики:	4
ИЗУЧЕНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ.....	5
2.1. Общие сведения	5
2.2. Минимальная реализация блокчейна	5
ПРОЕКТ НА ОСНОВЕ ТЕХНОЛОГИИ.....	8
3.1. Цель проекта	8
3.2. Архитектура приложения	8
3.3. Основные элементы GUI	9
3.4. Логика работы	9
МОДИФИКАЦИЯ (ТВОРЧЕСКАЯ ЧАСТЬ)	11
ТАБЛИЦА ФУНКЦИЙ ПРИЛОЖЕНИЯ.....	12
ИНТЕРФЕЙС ПРИЛОЖЕНИЯ	13
ЗАКЛЮЧЕНИЕ	17
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	18

ВВЕДЕНИЕ

Целью выполнения вариативной части проектной практики являлось ознакомление с современными информационными технологиями и реализация учебного проекта на основе выбранной темы. В качестве направления исследования была выбрана технология блокчейн — распределённая децентрализованная система хранения данных, обеспечивающая прозрачность, надёжность и неизменяемость информации.

На первом этапе практики было проведено ознакомление с теоретическими аспектами блокчейна, а также изучен пример программной реализации базовой цепочки блоков на языке Python. На втором этапе был выполнен учебный проект, заключающийся в создании графического интерфейса пользователя для работы с блокчейном на базе библиотеки PyQt5.

Данный отчет отражает хронологическую последовательность этапов выполнения работы:

Изучение теоретической части и воспроизведение примера кода.

Систематизация полученных знаний в техническом руководстве по технологии.

Разработка учебного приложения с графическим интерфейсом.

Внесение творческой модификации и подготовка итоговой технической документации.

ИНДИВИДУАЛЬНЫЙ ПЛАН

Этапы выполнения практики:

- Ознакомление с теорией технологии блокчейн.
- Изучение минимального кода реализации блокчейна на Python.
- Подготовка технического отчёта о технологии.
- Разработка графического интерфейса на PyQt5.
- Реализация творческой модификации (кнопка удаления блока).
- Подготовка финальной технической документации и отчёта.

ИЗУЧЕНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

2.1. Общие сведения

Блокчейн — это распределённый реестр, состоящий из цепочки блоков, где каждый блок содержит набор данных, временную метку и ссылку на предыдущий блок в виде хэша. Связь между блоками строится с использованием криптографических методов, что обеспечивает их неизменяемость.

Ключевые особенности технологии:

- **Децентрализация** — отсутствует единый центр управления.
- **Прозрачность** — все участники сети могут проверять целостность данных.
- **Неизменяемость** — изменения в уже записанных блоках невозможны без нарушения всей цепочки.

2.2. Минимальная реализация блокчейна

В рамках ознакомления с технологией был воспроизведён упрощённый пример кода на Python.

Структура блока:

- `index` — порядковый номер блока,
- `timestamp` — время создания,
- `data` — полезные данные,
- `previous_hash` — хэш предыдущего блока,
- `hash` — собственный хэш.

Основные функции:

- `create_genesis_block()` — создание генезис-блока (первого блока цепочки);
- `next_block(previous_block)` — формирование нового блока.

Пример кода иллюстрировал, как формируется цепочка и как каждый новый блок зависит от предыдущего.

```

import hashlib as hasher

class Block:
    def __init__(self, index, timestamp, data, previous_hash):
        self.index = index
        self.timestamp = timestamp
        self.data = data
        self.previous_hash = previous_hash
        self.hash = self.hash_block()

    def hash_block(self):
        sha = hasher.sha256()
        # Create encoded string as a hash body
        hash_body = (str(self.index) + str(self.timestamp) + str(self.data) + str(self.previous_hash)).encode()
        sha.update(hash_body)
        return sha.hexdigest()

import datetime as date

def create_genesis_block():
    # Manually construct a block with
    # index zero and arbitrary previous hash
    return Block(0, date.datetime.now(), "Genesis Block", "0")

```

```

def next_block(last_block):
    index = last_block.index + 1
    timestamp = date.datetime.now()
    data = "Hey! I'm block " + str(index)
    return Block(index, timestamp, data, last_block.hash)

# Create the blockchain and add the genesis block
blockchain = [create_genesis_block()]
last_block = blockchain[0]

# How many blocks should we add to the chain
# after the genesis block
num_of_blocks_to_add = 5

# Add blocks to the chain
for i in range(0, num_of_blocks_to_add):
    block_to_add = next_block(last_block)
    blockchain.append(block_to_add)
    last_block = block_to_add

print("Block #{0} has been added to the blockchain!".format(block_to_add.index))
print("Hash: {0}\n".format(block_to_add.hash))

```

Вывод:

```
Block #1 has been added to the blockchain!  
Hash: a100cf36a5d24cb7ca33ebdbb9ad56ad7652255ba5ecfbb60a2b3b275e66600d  
  
Block #2 has been added to the blockchain!  
Hash: 9b1b905f212cd0274b2d5e63ae570d44b22c98e28c74dab13d16ee72b4965ac6  
  
Block #3 has been added to the blockchain!  
Hash: a1ef010444097dcbd351b611b4b1f5944443976532cfdd943bc167f1c14a25ee  
  
Block #4 has been added to the blockchain!  
Hash: df3dc747cc73b9ebba3fc4e496a99d369474c003b41f0d56ba7255b0bac3ba01  
  
Block #5 has been added to the blockchain!  
Hash: a299db8281260059bdb8f952eec997e2bf2c8d9832cd9d9e925da740364da5fc
```

ПРОЕКТ НА ОСНОВЕ ТЕХНОЛОГИИ

3.1. Цель проекта

На основе изученной реализации блокчейна был разработан учебный проект, обеспечивающий удобное взаимодействие пользователя с блокчейном. Для этого использовалась библиотека **PyQt5**, позволяющая строить графический интерфейс.

3.2. Архитектура приложения

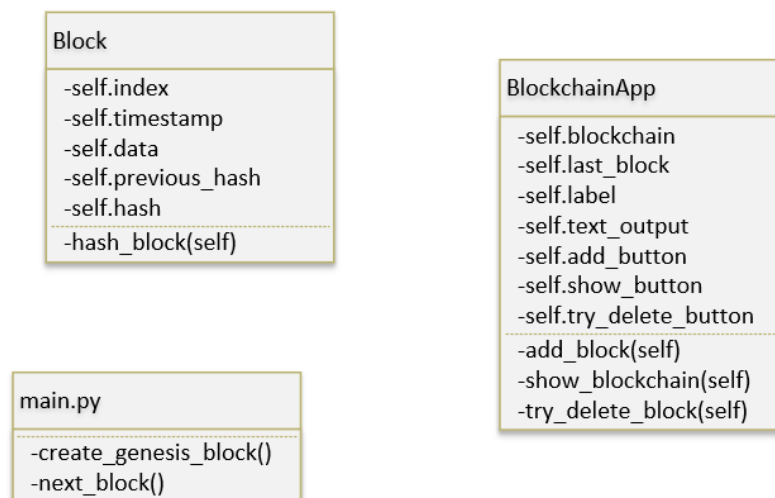
Приложение включает две части:

Backend (логика блокчейна):

- реализация структуры блока;
- функции добавления новых блоков;
- формирование хэшей и проверка связности цепочки.

Frontend (графический интерфейс):

- три кнопки для управления блокчейном;
- текстовое поле для вывода состояния цепочки;
- визуализация сообщений о результатах действий.



3.3. Основные элементы GUI

Интерфейс приложения содержит:

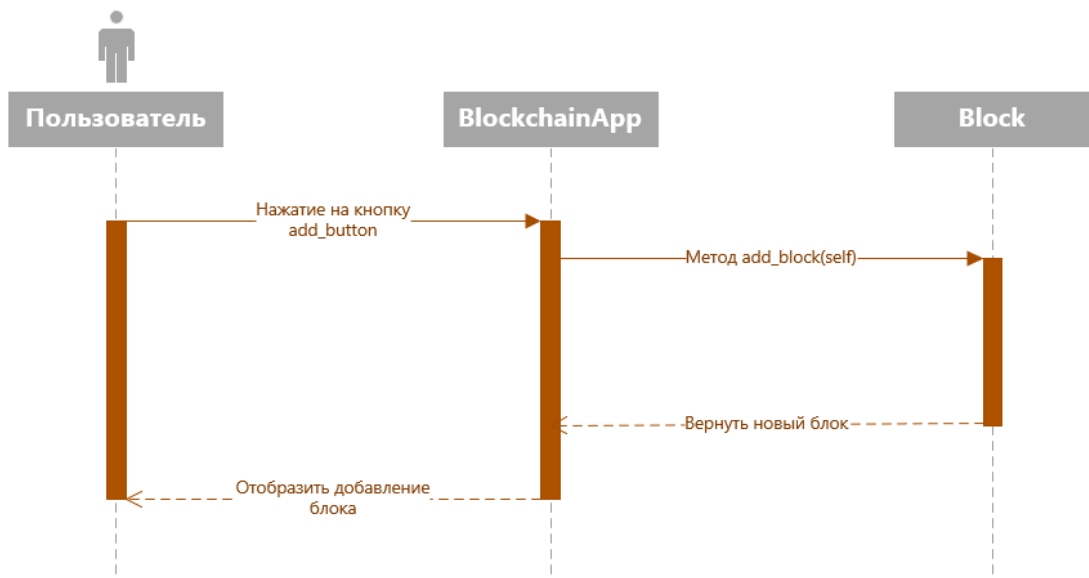
- метку с описанием программы;
- текстовое поле (QTextEdit) для отображения сообщений;
- кнопку «Добавить новый блок»;
- кнопку «Показать весь блокчейн»;
- кнопку «Попробовать удалить блок».

3.4. Логика работы

При запуске создаётся генезис-блок.

Пользователь может:

- добавить новый блок — цепочка увеличивается, на экран выводится уведомление;
- просмотреть всю цепочку — список блоков отображается в текстовом поле;
- попробовать удалить блок — на экран выводится сообщение об ошибке («структура блокчейна неизменяема»).



МОДИФИКАЦИЯ (ТВОРЧЕСКАЯ ЧАСТЬ)

В рамках творческой модификации была добавлена функция «Попробовать удалить блок».

Цель — наглядно показать неизменяемость блокчейна. При нажатии кнопки выводится сообщение об ошибке:

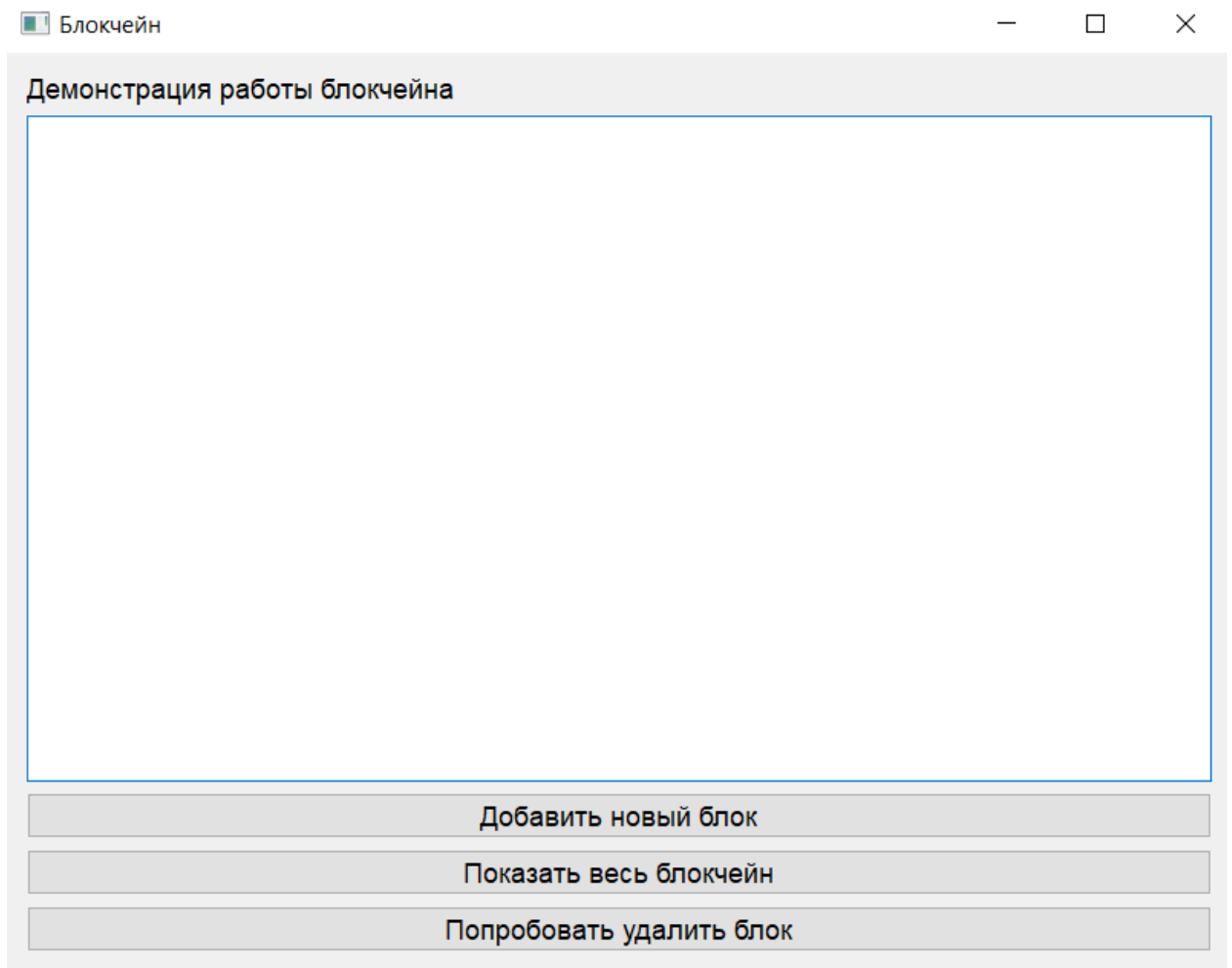
ОШИБКА: структура блокчейна неизменяема.

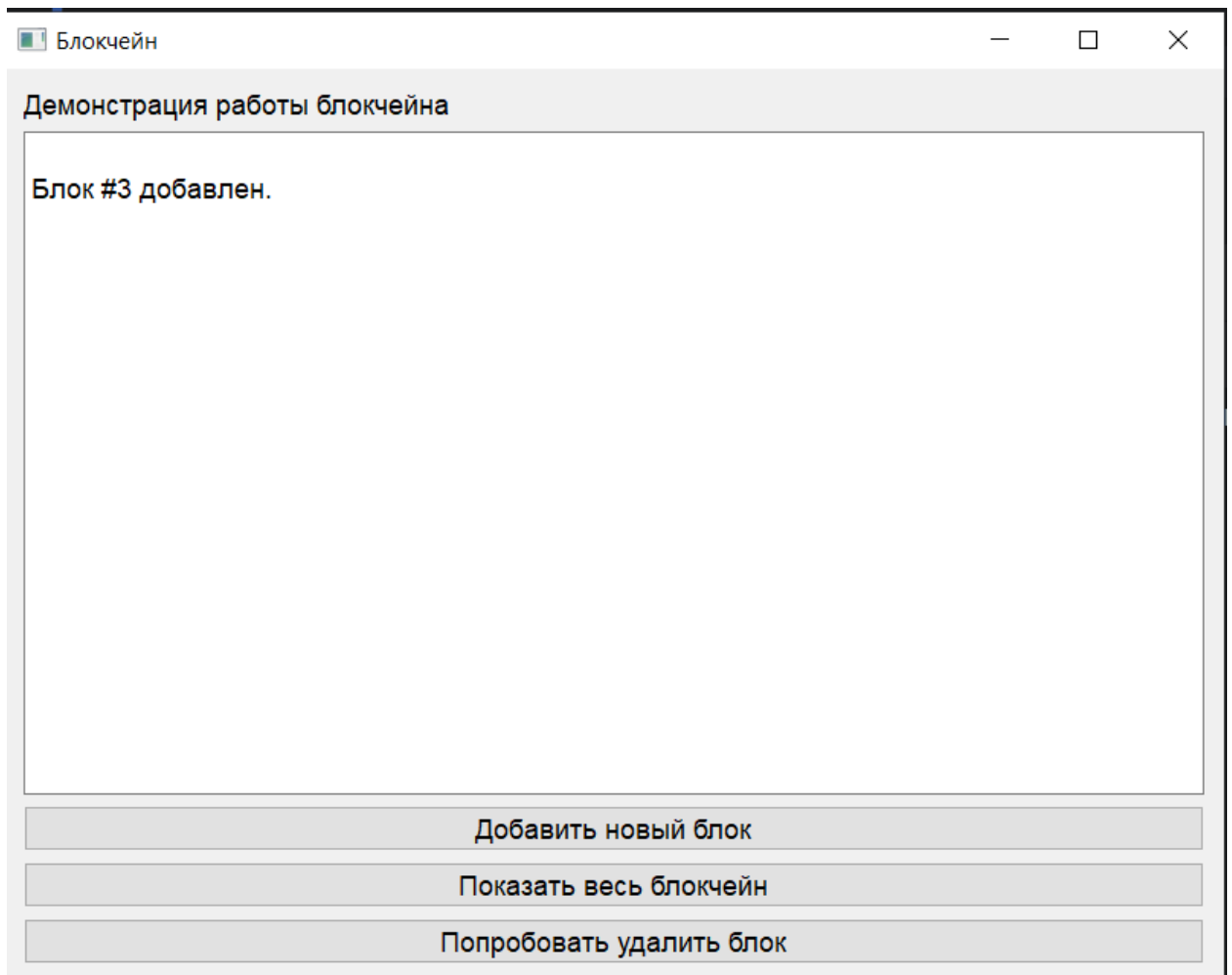
Таким образом, пользователю демонстрируется одно из ключевых свойств технологии блокчейн.

ТАБЛИЦА ФУНКЦИЙ ПРИЛОЖЕНИЯ

Кнопка	Действие	Результат на экране
Добавить новый блок	Создание нового блока	Сообщение о добавлении, увеличение цепочки
Показать весь блокчейн	Просмотр цепочки	Текстовый список всех блоков
Попробовать удалить блок	Демонстрация неизменяемости	Сообщение об ошибке

ИНТЕРФЕЙС ПРИЛОЖЕНИЯ





Демонстрация работы блокчейна

Текущий блокчейн

Общее количество блоков: 4

Блок #0

Хеш: #6a9a2fd5b681f59b83fc63f2929c6c5a14154f476c292de62fbfbefaf7271946

Блок #1

Хеш: #31ca696e99860209142fe89525fe03b0baed9bd967a18395844e12915f601ca5

Блок #2

Хеш: #b2e58c7bf086c0e3eb7c147c4c1bfcc0abd8d19b3bd88302838faa1bdbde89af

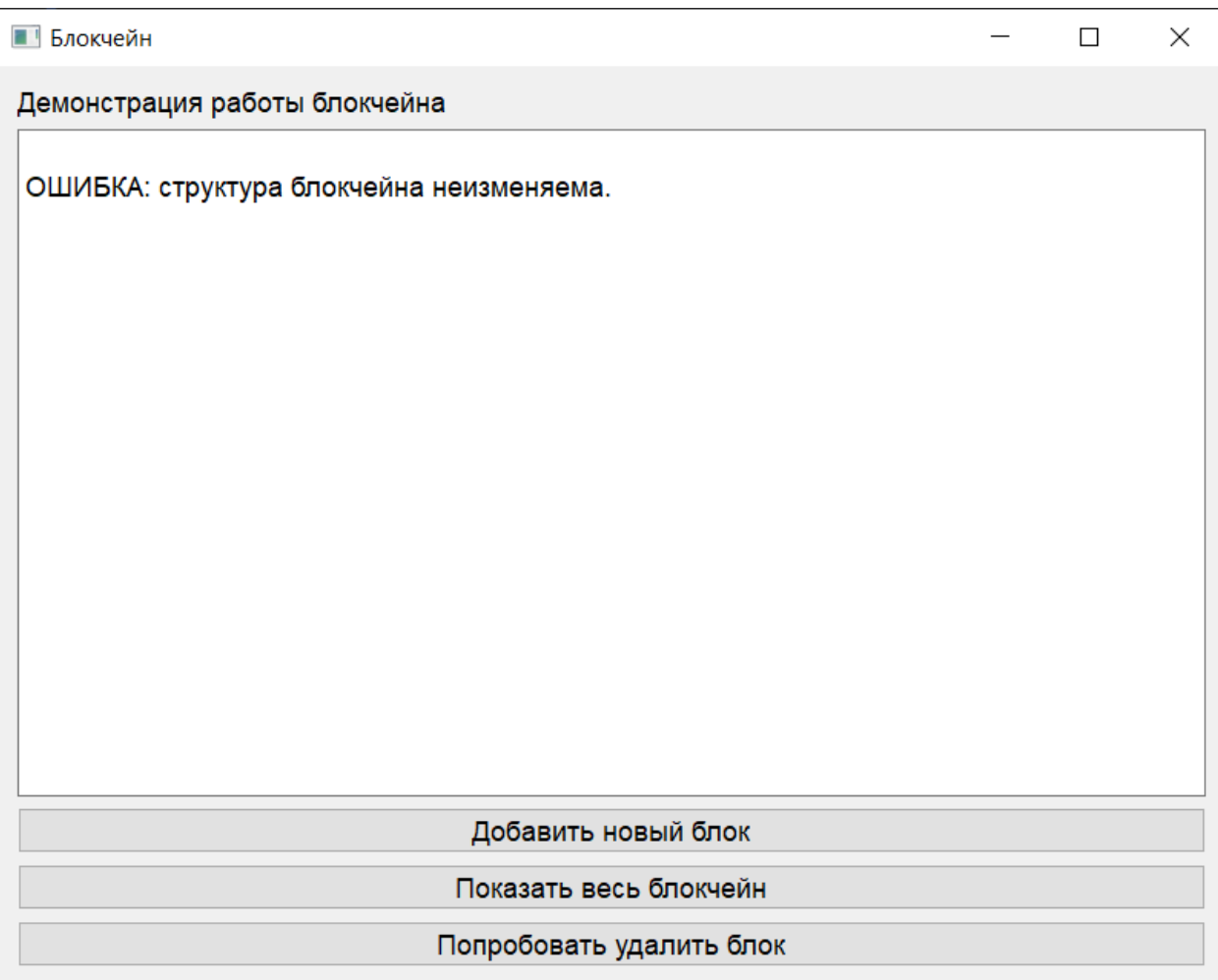
Блок #3

Хеш: #6ca1d1eb0e19b3321c10de25e676b848a0b2121d2e4f53d980f1ba1f50e658ce

Добавить новый блок

Показать весь блокчейн

Попробовать удалить блок



ЗАКЛЮЧЕНИЕ

В результате выполнения проектной практики было:

- освоено понятие и базовые принципы технологии блокчейн;
- изучена минимальная программная реализация блокчейна;
- создано учебное приложение с интерфейсом на PyQt5 для работы с цепочкой блоков;
- реализована творческая модификация, позволяющая продемонстрировать неизменяемость структуры блокчейна;
- подготовлена техническая документация с использованием UML-диаграмм, схем, таблиц и скриншотов.

Разработанный проект имеет учебный характер и может применяться для визуальной демонстрации принципов технологии блокчейн студентам и начинающим разработчикам.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Информация о практике Московского политеха <https://mospolytech.ru/obuchauschimsya/praktika/?ysclid=m9fpo3pwmu710957340> (дата обращения: 16.09.2025).
2. Информация о проектной деятельности <https://mospolytech.ru/obuchauschimsya/proektnaya-deyatelnost/?ysclid=m9fpsda3ad786727228> (дата обращения: 16.09.2025).
3. Официальный сайт организации-партнера <https://mospolytech.ru/> (дата обращения: 16.09.2025).
4. Организационная структура организации-партнера <https://mospolytech.ru/sveden/struct/> (дата обращения: 16.09.2025).
5. Репозиторий GitHub, созданный в рамках проектной практики https://github.com/lanshchekov/practice_2025_1 (дата обращения: 16.09.2025).
6. Сайт проекта https://lanshchekov.github.io/practice_2025_1/site/index.html (дата обращения: 16.09.2025).
7. Документация PyQt <https://pythonist.ru/rukovodstvo-po-pyqt5/> (дата обращения 16.09.25)