

Харківський національний університет імені В. Н. Каразіна
ННІ Комп'ютерних наук та штучного інтелекту
Кафедра інтелектуальних програмних систем і технологій

ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №2
на тему «Програмна реалізація примітивів блок і транзакція»
з дисципліни «Технології блокчейн та криптовалютні операційні
платформи»

Виконав: студент групи КС32

Травченко С.М.

Перевірила:

PhD, доц. Родінко М.Ю.

Харків – 2024

Завдання №1

Ознайомитись зі структурою транзакцій Bitcoin.

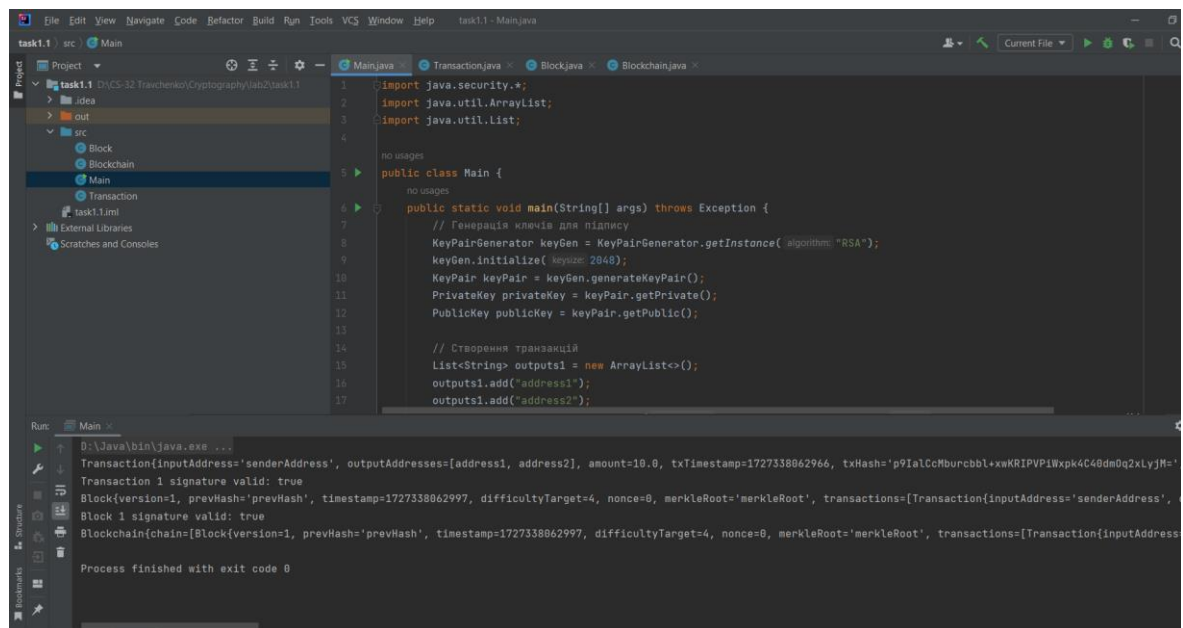
Розробити програмну реалізацію примітивів блок і транзакція мовою програмування Java (або іншою на вибір за бажанням).

- Розробити програмний примітив, що представляє транзакцію,
- Розробити програмний примітив, що представляє блок, тобто, клас Block,
- Розробити клас Blockchain, основним полем якого є список блоків (наприклад, List<Block>). Додати конструктор, методи для додавання нового блока, отримання останнього блока в ланцюгу (або заданого). За потреби можна додавати й інші методи.
- Провести тестування створених класів. Створити декілька блоків, кожен з яких містить по декілька транзакцій. Протестувати методи верифікації транзакцій та блоків, відповідних геш-значень та цифрових підписів.

Лістинги на гітхабі.

Результати виконання завдання №1 наведено:

1. На малюнку 1.1, 1.2 – результат виконання програми.

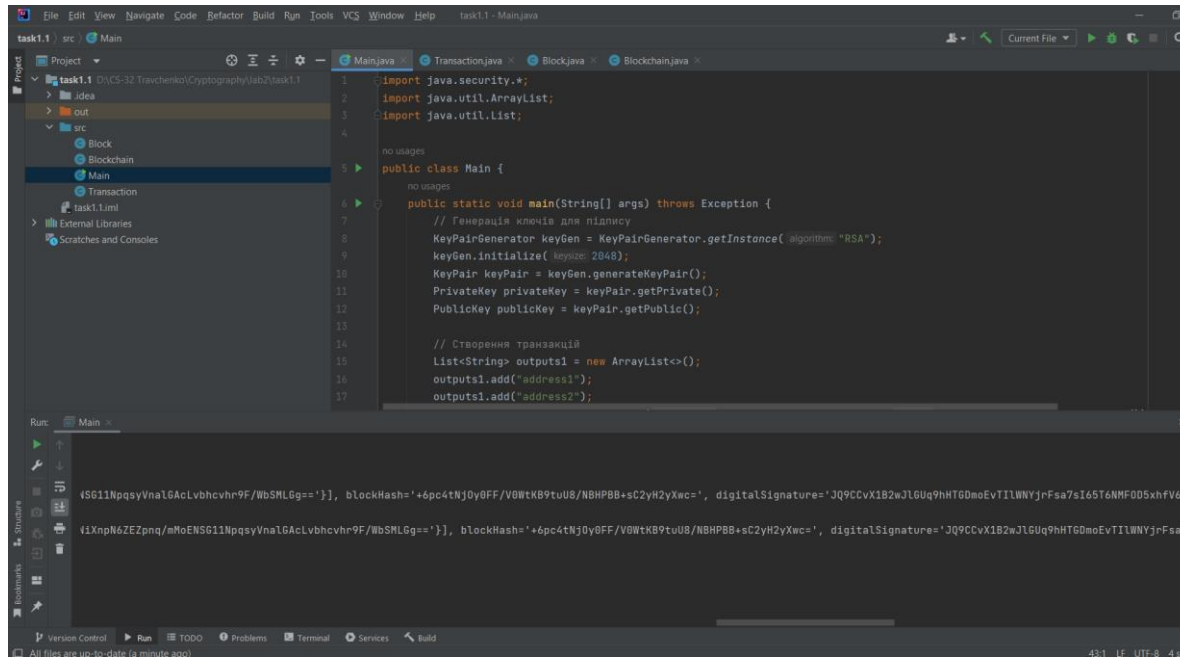


The screenshot displays an IDE with a project named 'task1.1'. The source files include 'Main.java', 'Transaction.java', 'Block.java', and 'Blockchain.java'. The 'Main.java' file is open, showing the following code:

```
1 import java.security.*;
2 import java.util.ArrayList;
3 import java.util.List;
4
5 public class Main {
6     // Генерує ключі для підпису
7     KeyPairGenerator keyGen = KeyPairGenerator.getInstance("RSA");
8     keyGen.initialize( keysize: 2048);
9     KeyPair keyPair = keyGen.generateKeyPair();
10    PrivateKey privateKey = keyPair.getPrivate();
11    PublicKey publicKey = keyPair.getPublic();
12
13    // Створення транзакцій
14    List<String> outputs1 = new ArrayList<>();
15    outputs1.add("address1");
16    outputs1.add("address2");
17}
```

The Run console shows the output of the program:

```
0:\Java\bin\java.exe ...
Transaction(inputAddress='senderAddress', outputAddresses=[address1, address2], amount=10.0, txTimestamp=1727338062966, txHash='p9Ia1CcMburcbb1+xxKRIPVP1WxpK4C40dm0q2xLyJM=')
Transaction 1 signature valid: true
Block(version=1, prevHash='prevHash', timestamp=1727338062997, difficultyTarget=4, nonce=0, merkleRoot='merkleRoot', transactions=[Transaction(inputAddress='senderAddress',
Block 1 signature valid: true
Blockchain(chain=[Block(version=1, prevHash='prevHash', timestamp=1727338062997, difficultyTarget=4, nonce=0, merkleRoot='merkleRoot', transactions=[Transaction(inputAddress=
Process finished with exit code 0
```



Малюнок 1.1 – результат виконання програми