Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

Predmet

– Digitálne meny a Blockchain –

- Dokumentácia -

MyBlockchain

Ak. Rok : 2021/2022, letný semester

Cvičiaci:

Ing. Viktor Valaštín

Študent:

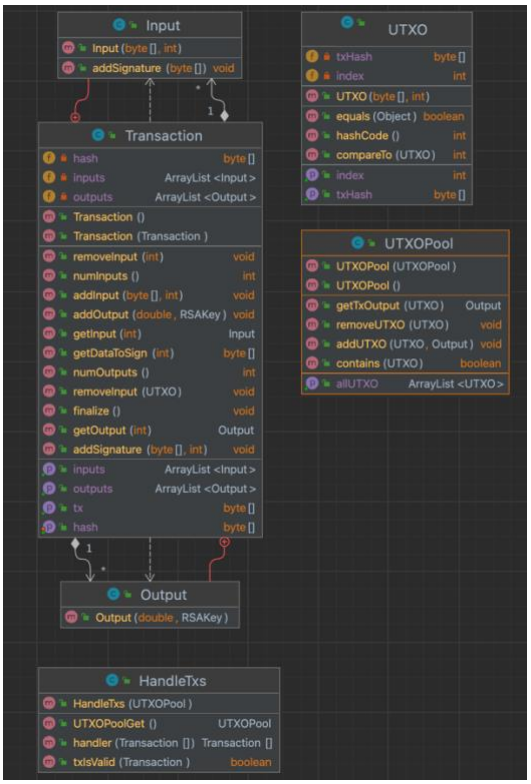
Martin Rudolf 97029

Bratislava, 2022.

Obsah:

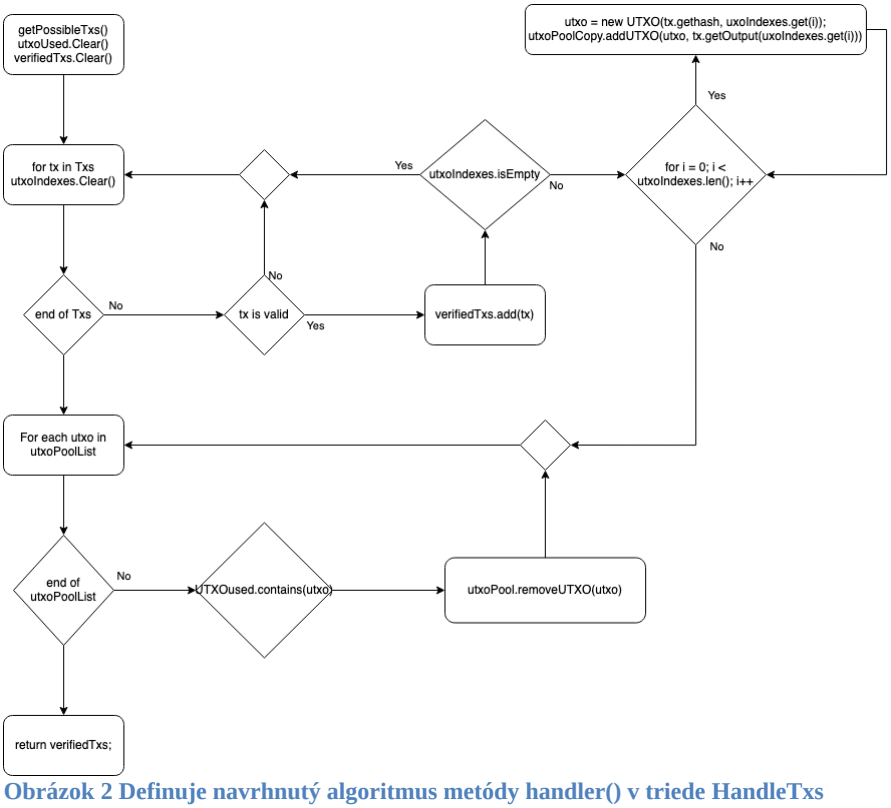
1) Blokový návrh riešenia

a) UML class diagram Faza 1



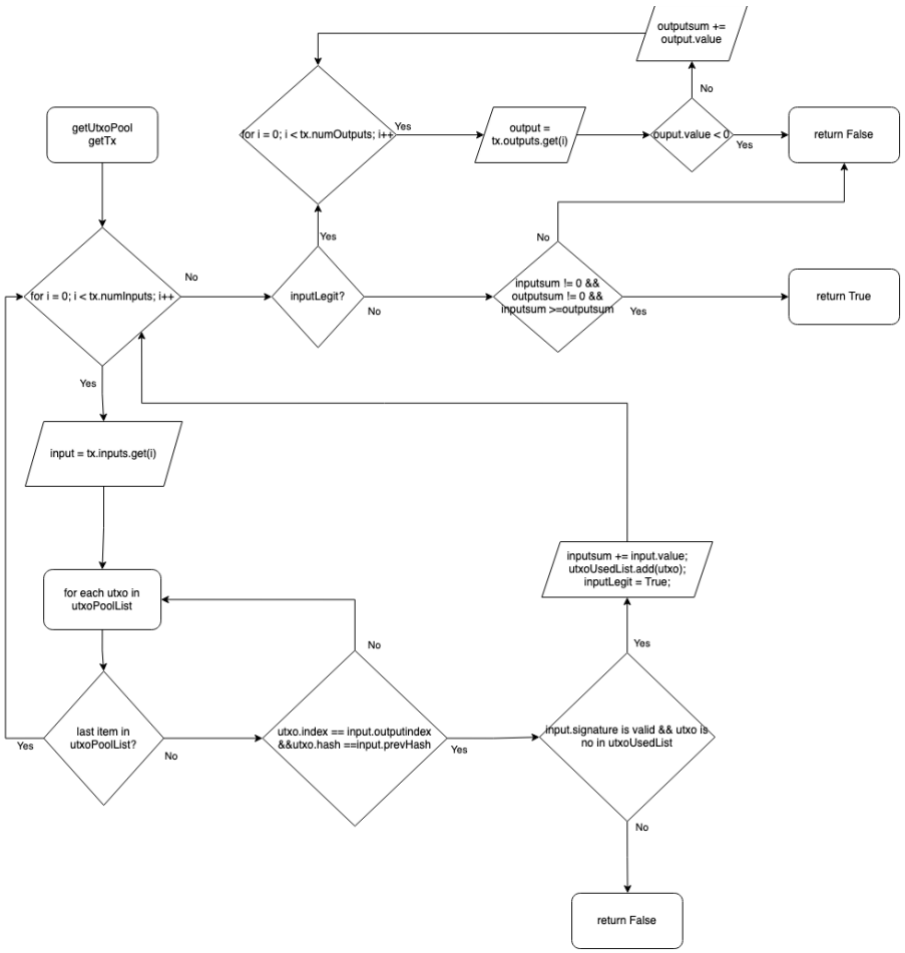
Obrázok 1 UML class diagram definuje triedy vo Fáze 1, ich atribúty a metódy

b) HandleTxs.handler() Flowchart



Obrázok 2 Definuje navrhnutý algoritmus metódy handler() v triede HandleTxs

c) HandleTxs.txIsValid() Flowchart



Obrázok 3 Definuje navrhnutý algoritmus metódy txIsValid() triedy HandleTxs

2) Voľba implementáčneho prostredia

Pre zadanie 1 boli poskytnuté predkódené triedy v jazyku Java. Pre vývoj v tomto jazyku mi príde ako prirodzené vývojové prostredie IntelliJ IDEA verzia 2021.3.3, preto aj kôli zvyku som siahol po tomto prostredí pre prácu na tomto zadaní. Java Development Kit verzie 11.0.13.

3) Definovanie a vysvetlenie jednotlivých fáz

a) Fáza 1

V tejto fáze zadania bolo potrebné implementovať triedu `HandleTxs`.

Podľa môjho návrhu trieda disponuje atribútmi znázornenými v UML diagrame prvej kapitoly.

Konštruktor do utxoPoolcopy atribútu triedy, priradí kópiu inštancie triedy UTXOPool.

Metóda `txIsValid()` dostane na vstupe inštanciu triedy `Transaction` ktorá nesie svoj hash, pole vstupov a pole výstupov. Logika tejto metódy spočíva v tom že algoritmus berie zaradom vstupy transakcie a porovná hash a index vstupu s každým prvkom v poli ktoré obsahuje aktuálne inštancie triedy UTXO (takéto pole generuje metóda inštancie utxoPoolcopy). Ak dôjde k zhode hashu a indexu tak algoritmus následne overí podpis vstupu pomocou metódy `verifySignature()` v knižnici rsa a skontroluje či dané UTXO už nebolo použité v nejakej predchádzajúcej transakcii. Ak sa jedná o neplatný podpis alebo UTXO už bolo použité tak metóda vráti hodnotu False a teda transakcia je neplatná. Inak pripočítam hodnotu vstupu k celkovej hodnote všetkých vstupov, pridá do poľa adries adresu z výstupu z ktorého pochádza matchnuté UTXO, pridá ho do poľa `UTXOused` a nastaví príznak `inputLegit` na `true`. Toto sa opakuje pre každý vstup transakcie. Následne ak je nastavený príznak `inputLegit` na `true`, tak prejde každý výstup transakcie kde skontrolujeme jeho hodnotu. Ak je atribút `value` záporný metóda vráti `false`, inak pripočíta hodnotu k celkovej hodnote všetkých výstupov a pozrie sa či sa adresa výstupu nachádza medzi adresami ktoré generovali použité UTXO, tak nájdeme indexy výstupov ktoré generujú nové UTXO. Po skontrolovaní výstupov skontrolujeme či sučet hodnôt vstupov je väčší alebo rovný ako súčet hodnôt výstupov, ak áno tak metóda vráti `true`, inak `false`. Algoritmus je blokovo opísaný vývojovým diagramom v predchádzajúcej kapitole.

Metóda `handler` dostane ako vstupný argument pole potencionálnych transakcií, v ktorom skontroluje každú transakciu pomocou vyššie spomenutej metódy `txIsValid()`. Ak sa transakcia vyhodnotí ako platná tak sa pridá do poľa `verifiedTxsList`. Ak pole `utxoIndexes`, ktoré nesie indexy výstupov danej transakcie ktoré generujú nové UTXO, je neprázdne tak pre každý prvok v poli sa vytvorí nová inštancia UTXO a je pridaná do aktuálneho `utxoPoolcopy`. Toto sa vykoná pre každú transakciu z poľa vstupného argumentu metódy. Následne metóda prejde všetky UTXO inštancie z aktuálneho poolu a ak sa dané UTXO nachádza v poli `UTXOused` tak odstráni dané UTXO z aktualného poolu. Nakoniec metóda vráti pole platných transakcií. Blokový opis tohto algoritmu si možno všimnúť v prvej kapitole.

4) Používateľská príručka

Na spustenie fázy 1 spustíme súbor Main_faza1.java ktorý definuje main funkciu v ktorej sú vytvorené transakcie na testovanie implementovaných metód. Obmieňaním rôznych parametrov transakcií sa testovali rôzne scenáre, najme scenáre opísané v textovom súbore s opismi jednotkových testov od garanta predmetu.

5) Záver

Cieľom tohto zadania bolo pochopiť ako fungujú Bitcoin transakcie a celkovo UTXO logika, rovnako aj nadobudnutie znalosti ako vyzerá a ako sa správa dôveryhodný a nedôveryhodný, takzvaný Byzantský, uzol v sieti, a teda docielení konsenzu v sieti. V neposlednom rade sme si osvojili teoretickú znalosť v spravovaní a overovaní blokov a začleňovanie ich do blockchainovej siete.

Nadobudnutie teoretických vedomostí pri tomto zadaní bolo plodné a výživné.
Teoretické znalosti som si stihol úspešne vyskúšať len Fáze 1, kde som navrhol a implementoval algoritmus na overovanie a spravovanie transakcií. Určite po refaktoringu by tento algoritmus mohol byť efektívnejší, no myslím že cieľ tohto zadania spĺňa.