Slovenská Technická Univerzita v Bratislave Fakulta informatiky a informačných technológií

Digitálne meny a Blockchain

Zadanie 1 – MyBlockchain

Fáza 1

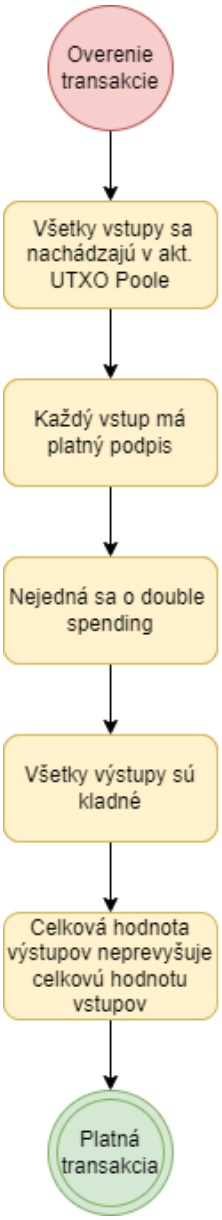
Úlohou bolo vytvoriť implementovať logiku na overovanie a spracovanie blockchainových transakcií a vytvorenie ledgeru. V každom bloku dostane centrálna autorita zoznam transakcií, ktoré musia byť overené a následne zverejní zoznam platných transakcií. Avšak môže nastať situácia, kedy sa jedna transakcia bude odkazovať na inú v tom istom bloku. Taktiež sa môže stať to, že v jednom bloku sa nachádza viac ako jedna transakcia míňajúca rovnaký výstup (UTXO) – double-spending. Vtedy je transakcia neplatná. Takže transakcie nemôžu byť kontrolované izolovanie.

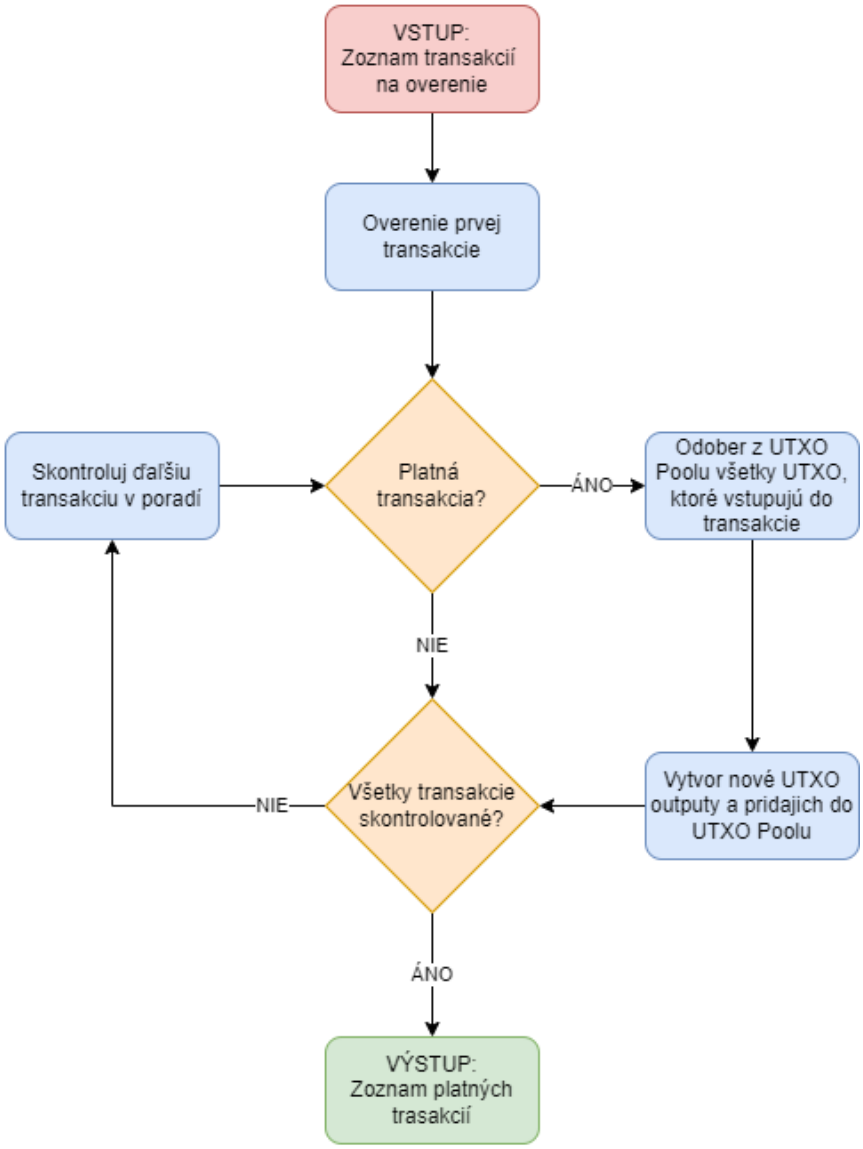
Funkcia ***txIsValid(Transaction tx)*** v triede **HandleTxs** sa stará o to, aby skontrolovala, či je konkrétna transakcia platná. Pri kontrole platnosti transakcie sa kontroluje, či:

- sú všetky výstupy nárokované v aktuálnom UTXO pool
- podpisy na každom vstupe sú platné,
- žiadne UTXO nie je nárokované viackrát,
- všetky výstupné hodnoty s sú nezáporné a
- súčet vstupných hodnôt je väčší alebo rovný súčtu jej výstupných hodnôt

Funkcia ***Transaction[] handler(Transaction[] possibleTxs)*** v triede **HandleTxs** postupne kontroluje a spracováva všetky obdržané transakcie. Kontrola transakcií prebieha vo viacerých kolách, kedy sa zisťuje, či sa v predošlom kole našla aspoň jedna platná transakcia. To rieši problém, kedy by sa nejaká transakcia mohla odkazovať na inú v tom istom bloku. Pokiaľ áno, tak sa kontrola opakuje. Ak nie, kontrola končí a funkcia vráti zoznam platných transakcií.

Blokový návrh riešenia





Fáza 3

Úlohou bolo implementovať uzol, ktorý je súčasťou distribuovaného konsenzuálneho protokolu založeného na blokových reťazcoch. Uzol prijíma prichádzajúce transakcie a bloky a udržuje aktualizovaný blockchain. Ten môže mať obrovskú veľkosť, preto sa udržuje iba niekoľko posledných úrovní blokov.

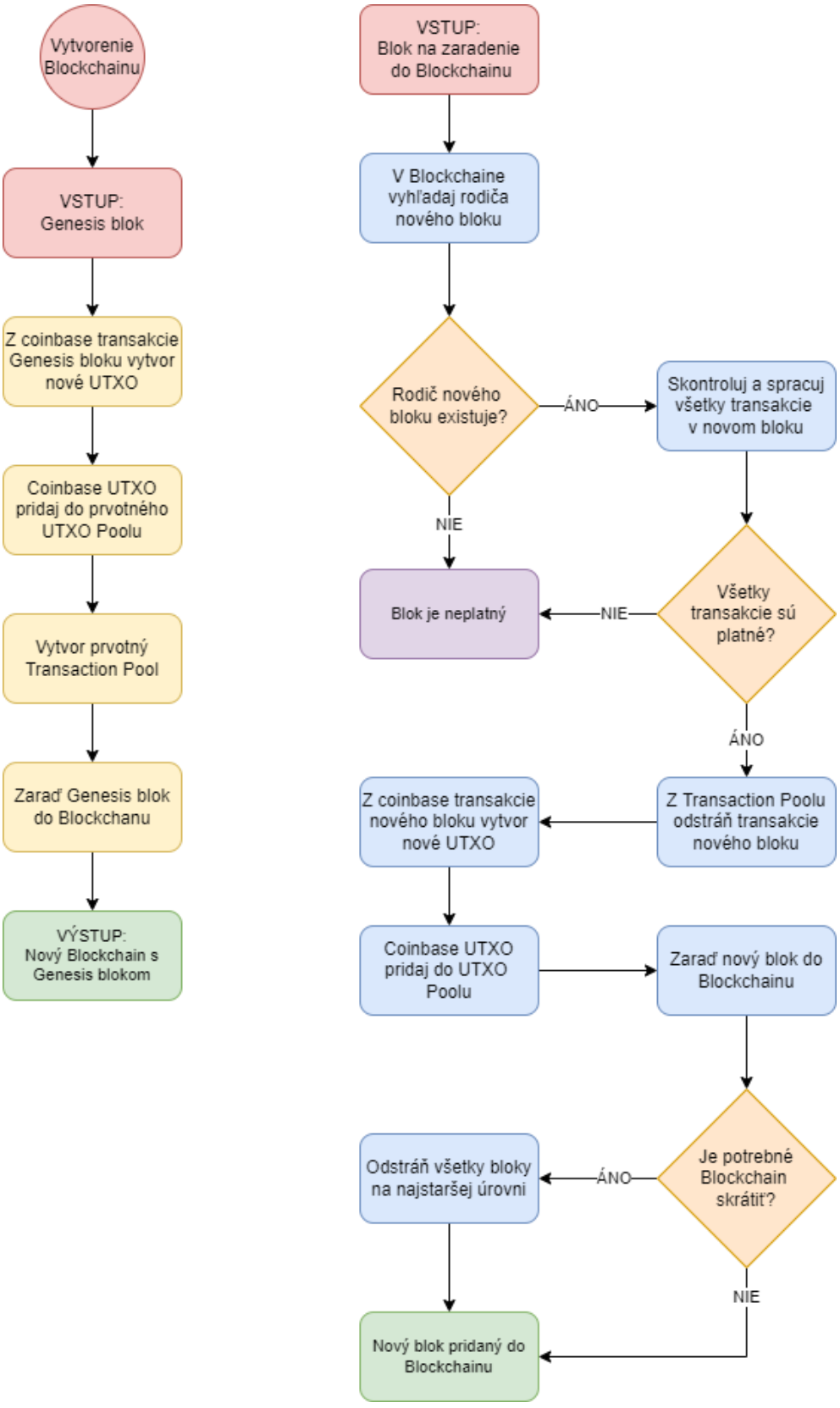
Konštruktor ***Blockchain(Block genesisBlock)*** v triede **Blockchain** vytvorí nový blockchain s prvotným Genesis blokom a Transaction Poolom.

Funkcia ***boolean blockAdd(Block block)*** v tiede **Blockchain** zabezpečuje zaradenie bloku do blockchainu, pokiaľ je platný a spĺňa nasledujúce požiadavky:

- rodič nového bloku sa nachádza v udržiavanom zozname blokov
- všetky transakcie v nového bloku sú platné

Funkcia ďalej zabezpečuje skracovanie zoznamu blokov podľa aktuálnej maximálnej výšky blockchainu a maximálnej udržiavanej úrovne blokov.

Blokový návrh riešenia



Zhodnotenie

Fáza 1

V prvej fáze som sa naučil, ako funguje overovanie transakcií, resp. že na to, aby bola transakcia považovaná za platnú je potrebné, aby spĺňala určité podmienky. Ďalej som sa zistil, čo je to "double spending" a kedy pri spracovaní transakcie môže nastať. A taktiež to, že transakcie sa môžu skladať z viacerých vstupov a výstupov, ktoré môžu na seba v jednom bloku odkazovať.

Fáza 3

V tretej fáze som zistil, ako sa napájajú nové bloky do blockchainu na už existujúce bloky. Taktiež to, že pri každom novo-vyťaženom bloku vzniká odmena za jeho vyťaženie (coinbase), ktorá je pripísaná tomu, kto daný blok vyťažil. Ako fungujú je to forky v blockchaine a ako môže jedna vetva prerásť druhú, resp. čo sa následne stane s transakciami a odmenami.

Všeobecné informácie

- Vývojové prostredie
  - *IntelliJ IDEA 2021.3.2 (Ultimate Edition)*
- Programovací jazyk
  - *Java SE 16*