SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

BLOCKCHAIN ZADANIE 1

Roman Páleník
FIIT STU
Cvičenie: Streda 18:00
20.3.2022

Contents

1 Opis riešenia

1.1 Blokový návrh

Blokový návrh sa nachádza na konci dokumentu, kvôli svojej veľkosti.

1.2 Fáza 1

V tejto fáze bolo treba naimplementovať triedu handleTx, ktorá validuje transakcie a bloky potencionálnych transakcíi. Táto logika sa realizuje v metódach txIsValid a handler.

Metóda txIsValid pozostáva z 5 častí: či sú všetky nárokované vstupy aktuálnom UTXO pool, podpisy na každom vstupe sú platné, žiadne UTXO nie je nárokované viackrát, všetky výstupné hodnoty sú nezáporné a súčet vstupných hodnôt je väčší alebo rovný súčtu jej výstupných hodnôt. Metóda sa pozrie na všetky výstupy, vytvorí z nich potencionálne UTXO a to následne porovná s každým v aktuálnom pool-e, ktorý ma trieda handleTx. Následne sa z každého vstupy znova spravia UTXO a porovnávajú sa medzi sebou aby sa odhalil double spending. Kedže sa podpisujú iba inputy tak tie sú postupne prezerané a funckciou verifySignature() sa kontroluje správnosť ich podpisov. Nakoniec sa skontroluje či nie je záporný output a či je $hodnota\ vstupov \leq hodnota\ vystupov$. Ak sa nájde nejaká chyba pri týchto kontrolách je transakcia označená za neplatnú.

Handle metóda pracuje tak, že spracováva transakcie, ktoré jej prídu v poli. Každá transakcia je kontrolovaná metódou txIsValid a ak je v poriadku je pridaná to poľa správnych transakcíi. Medzi týmito transakciami sa ale môže nachádzat double spending, preto pri začiatku celej kontroly bloku sa inicializuje pole všetkých UTXO v bloku a to je potom v txIsValid osobitne kontrolované s každým inputom, či tam nedošlo k double spendingu. Po kontrole je vrátené pole validných transakcíi.

1.3 Fáza 2

Cieľom fáze 2 bolo naprogramovať dôveryhodný uzol z sieti a eliminovať byzanské, ktoré škodia sieti. Preto dôveryhodný uzol kontrolu, čo mu poslali iné uzly a keď mu neprišlo nič. Takto si vytvorí blacklisted, ktorý označuje uzly, ktorým už nebude nič posielať. Na konci sa takto všetky uzli dohodnú na transakciách a prešíria sa všetky transakcie po celej sieti.

1.4 Fáza 3

Vo fáze 3 sme mali implementovať pridávanie nového bloku a vytvorenie nové blockchain-u. Proces začal vytvorením genesis teda prvého bloku a jeho pridanie do nového poľa, ktorý udržiava bloky v reťazi. Za vytvorenie prvého bloku sa dostala odmena, s ktorou mohol ten čo blok ťažil ďalej pracovať.

Block sa pridával v metóde block add. Táto metóda brala argument blok, ktorý sa má pridať. Ako prvé program našiel rodiča, na ktorý sa tento blok napájal, aby sa zobrali aktuálne UTXO pre tento blok. Následne sa skontrolovali transakcie v bloku. Ak boli všetky valídne tak sa skontrolovalo či sedí výška, podľa vzorca $nova\ vyska \leq maximalna\ vyska - CUT\_OFF\_AGE$. Robí sa to preto, aby sa blok nevedel napojiť na príliš staré bloky v reťazi, ktoré už považujeme za spracované. Validné transakcie sa vymažú z TransactionPool a pridá odmenu za vyťaženie bloku a pridá blok do reťaze. Na konci sa vymažú bloky, ktoré sú už dostatočne dlho v blockhain-e a preto ich už považujeme za platné a môžeme ich vymazať.

2 Voľba implemetačného prostredia

Implementáčné prostredie som sa rozhodol pre Javu. Nie len zdrojové kódy boli v Jave ale považujem ju za lepšiu v tomto prípade. Pri študovaní cudzieho kódu je dôležité vedieť aké typy môžem použiť. Podobne aj volané funkcie majú presné parametre a ľahšie sa orientuje v kóde. Do pythonu som to neprepísal, pretože Java má lepšie spracovanie pre triedy, je to čitateľnejšie a pomáha to v orientácii.

3 Záver

V tomto zadaní som sa naučil ako funguje blockchain detailnejšie ako doteraz. Implementáciou som si uvedomil, aké procesy tam prebiehajú a na čo všetko sa musí dávať pozor a kde existuje určitá limitácia tejto technológie. Z tohto hľadiska bolo zadanie dobre spracované.

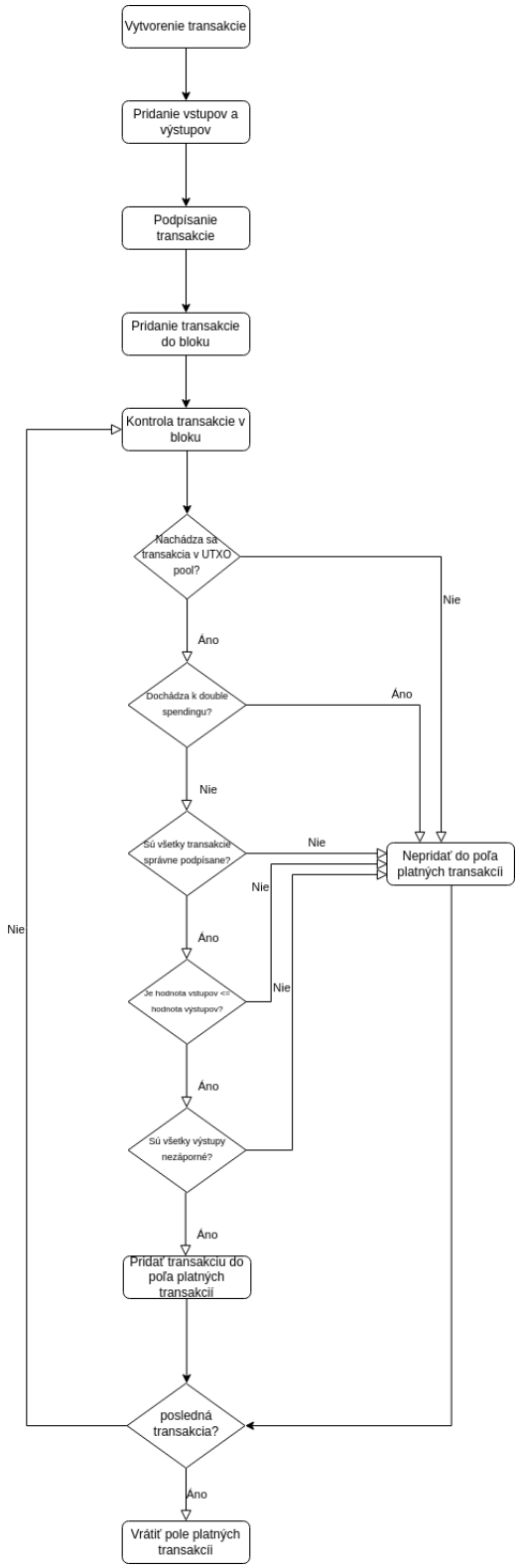

Figure 1: UML diagram fungovania fázy 1

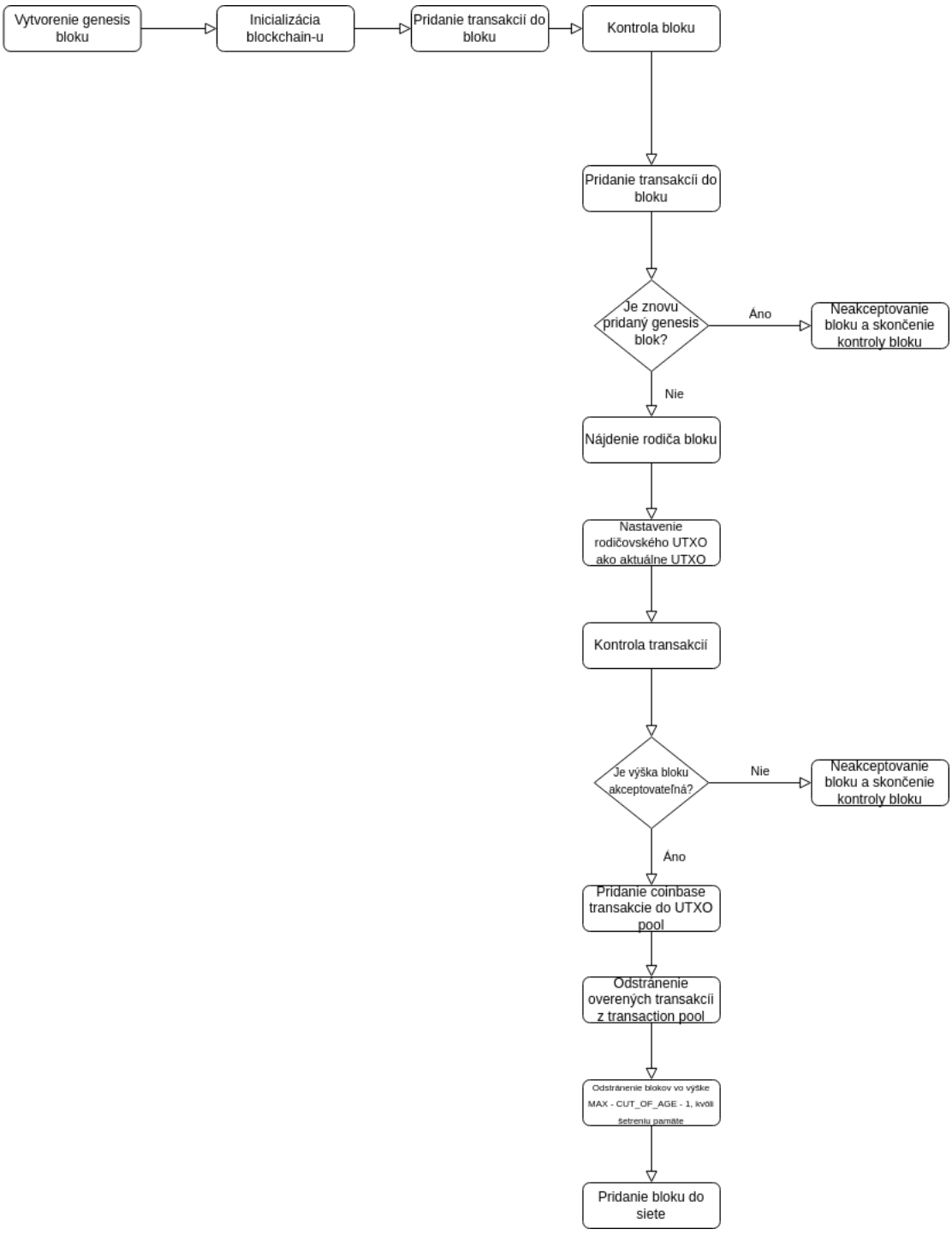

Figure 2: UML diagram fungovania fázy 3