

Work on project. Stage 4/6: You’ve got a message

34 users solved this problem. Latest completion was 1 day ago.

Project: [Blockchain](#)

Hard

21 minutes

§1. Description

For now, we are mining blocks to create a blockchain, but just the blockchain itself is not particularly useful. The most useful information in the blockchain is the data that every block stores. The information can be anything. Let's create a simple chat based on the blockchain. If this blockchain works on the internet, it would be a world-wide chat. Everyone can add a line to this blockchain, but no one can edit it afterward. Every message would be visible to anyone.

In this stage, you need to upgrade the blockchain. A block should contain messages that the blockchain received during the creation of the previous block. When the block was created, all new messages should become a part of the new block, and all the miners should start to search for a magic number for this block. New messages, which were sent after this moment, shouldn't be included in this new block. Don't forget about thread synchronization as there is a lot of shared data.

You don't need any network connections as this is only a simulation of the blockchain. Use single blockchain and different clients that can send the message to the blockchain just invoking one method of the blockchain.

So, the algorithm of adding messages is the following:

1. The first block doesn't contain any messages. Miners should find the magic number of this block.
2. During the search of the current block, the users can send the messages to the blockchain. The blockchain should keep them in a list until miners find a magic number and a new block would be created.
3. After the creation of the new block, all new messages that were sent during the creation should be included in a new block and deleted from the list.
4. After that, no more changes should be made to this block apart of the magic number. All new messages should be included in a list for the next block. The algorithm repeats from step 2.

§2. Output example

To be tested successfully, program should output information about first five blocks of the blockchain. Blocks should be separated by an empty line.

22 / 22 Prerequisites

- ✓

[Decorator](#)

Stage 4
- ✓

[Facade](#)

Stage 4
- ✓

[The Collections Framework overview](#)

Stage 43
- ✓

[List](#)

Stage 43
- ✓

[Set](#)

Stage 4

Show all

```
Block:
Created by miner # 9
Id: 1
Timestamp: 1539866031047
Magic number: 92347626
Hash of the previous block:
0
Hash of the block:
1d12cbbb5bfa278734285d261051f5484807120032cf6adcca5b9a3dbf0e7bb3
Block data: no messages
Block was generating for 0 seconds
N was increased to 1
```

```
Block:
Created by miner # 7
Id: 2
Timestamp: 1539866031062
Magic number: 34678462
Hash of the previous block:
1d12cbbb5bfa278734285d261051f5484807120032cf6adcca5b9a3dbf0e7bb3
Hash of the block:
04a6735424357bf9af5a1467f8335e9427af714c0fb138595226d53beca5a05e
Block data:
Tom: Hey, I'm first!
Block was generating for 0 seconds
N was increased to 2
```

```
Block:
Created by miner # 1
Id: 3
Timestamp: 1539866031063
Magic number: 56736428
Hash of the previous block:
04a6735424357bf9af5a1467f8335e9427af714c0fb138595226d53beca5a05e
Hash of the block:
0061924d48d5ce30e97cfc4297f3a40bc94dfac6af42d7bf366d236007c0b9d3
Block data:
Sarah: It's not fair!
Sarah: You always will be first because it is your blockchain!
Sarah: Anyway, thank you for this amazing chat.
Block was generating for 0 seconds
N was increased to 3
```

```
Block:
Created by miner # 2
Id: 4
Timestamp: 1539866256729
Magic number: 37567682
Hash of the previous block:
0061924d48d5ce30e97cfc4297f3a40bc94dfac6af42d7bf366d236007c0b9d3
Hash of the block:
000856a20d767fbbbc38e0569354400c1750381100984a09a5d8b1cdf09b0bab6
Block data:
Tom: You're welcome :)
Nick: Hey Tom, nice chat
Block was generating for 5 seconds
N was increased to 4
```

Code Editor

IDE



✓ IDE is opened

If you don't see your IDE opened, switch to it manually

Comments (1)

Hints (2)

Useful links (0)

Solutions (0)

Share something, Sergey Kubatko

Post

Please do not post solutions here

D

Dmitriy

5 days ago

Report

You can use the VisualVM tool to visualize your threads

<https://visualvm.github.io>

♡ 0

Reply