Holding Objects Lab 2:

Cryptocurrencies are a type of currency that is defined entirely in software. Cryptocurrencies are a hot topic right now, with many different currencies in circulation, all with their own values and unique traits. One of the most popular and well-known cryptocurrencies is called Bitcoin.

Bitcoin uses a technology called Blockchain to keep a public ledger of all bitcoin-denominated transaction. The blockchain gets its name from the fact that transactions are grouped together into batches called blocks, and a new block is added to the ledger approximately every ten minutes. Each block has a reference, or link, to the previous block, hence the 'chain' part of the name.

In this lab you will implement a model of a blockchain, using Java Collection objects.

Instructions

Create the following classes:

- `Address` - contains a 64-bit number (called a "public key") with an optional `name` field. If `name` is not specified it should be the empty string "". Addresses can be renamed, but the number must remain constant and be unique.
- `Transaction` - Contains a source and destination address, and a transaction amount (bitcoins are expensive, so we want to be able to trade fractions of a bitcoin *hinthint*)
- `Block` - Contains a group of transactions as well as a reference to the previous block. Transaction order is important, and repeated identical transactions are possible -- be sure to use an appropriate data type.
- `BlockChain` - Contains a reference to the most recently added block, which in turn has a reference to the previous block and so on.

All of these classes should have overridden toString methods that allow them to be converted to strings and printed nicely. BlockChain should provide methods for getting the latest block, or getting the entire list of blocks.

Demonstrate that your code works by generating a series of transactions, creating a block from them, and adding it to the blockchain. Do this for several blocks.

Part II:

Create a `Wallet` class. A Wallet contains an Address, a balance, and a private key -- a secret number used to access the wallet. Normally private keys are generated with a cryptographic function, but for this lab you can

use a random number.

A Wallet can contain only one address, but to preserve their anonymity users often have many wallets. Create a `WalletManager` class that stores Wallets in a Map, using the Wallet's public key as the key in the Map.

Extra Challenge

Add a `validate` method to `Blockchain`. It should take a `Block` object as its argument and check all transactions **in order**, verifying that the source addresses had the requisite funds to make the transaction. Bogus transactions are a common problem and preventing their acceptance is a primary purpose of public ledger technology like blockchain.

Once the `validate` method works, refactor your `addBlock` method to only add a block if its verification is successful. Complete this challenge and you may be eligible to join the elite group of cryptocurrency engineers known only as "Zyptocoders".