Afterpay Touch Coding Exercise

Introduction

Welcome to the Afterpay Touch Coding Exercise

We'd like to get a feel for how you approach problems, how you think, and how you design your code. Please complete the exercise below in Java and send us your working and tested solution.

Please do not share the exercise on social media (GitHub, Twitter etc).
Thank you and have fun!

Exercise

Consider the following credit card fraud detection algorithm

A credit card transaction is comprised of the following elements.

- hashed credit card number
- timestamp - of format 'year-month-dayThour:minute:second'
- amount - of format 'dollars.cents'

Transactions are to be received as a comma separated string of elements.
eg. `10d7ce2f43e35fa57d1bbf8b1e2, 2014-04-29T13:15:54, 10.00`

A credit card will be identified as fraudulent if the sum of amounts for a unique hashed credit card number over a 24 hour sliding window period exceeds the price threshold.

Write a method, which when given a sequence of transactions in chronological order, and a price threshold, returns the hashed credit card numbers that have been identified as fraudulent.

Guidance

Feel free to create any additional classes you need to support the design of your solution.

We expect to see tests that prove your code works.

We are looking for pragmatic, testable, and maintainable code. If in doubt, refer to the KISS principle.

> Any fool can write code that a computer can understand. Good programmers write code that humans can understand.

- Martin Fowler