Code Challenge — Authorizer

You are tasked with implementing an application that authorizes a transaction for a specific account following a set of predefined rules.

Please read the instructions below, and feel free to ask for clarifications if needed.

Packaging

Your README file should contain a description on relevant code design choices, along with instructions on how to build and run your application.

Building and running the application must be possible under Unix or Mac operating systems. Dockerized builds are welcome.

You may use open source libraries you find suitable, but please refrain as much as possible from adding frameworks and unnecessary boilerplate code.

Your program is going to be provided `json` lines as input in the `stdin`, and should provide a `json` line output for each one — imagine this as a stream of events arriving at the authorizer.

Sample usage

```
$ cat operations
{ "account": { "activeCard": true, "availableLimit": 100 } }
{ "transaction": { "merchant": "Burger King", "amount": 20, "time": "2019-02-13T10:00:00.000Z" } }
{ "transaction": { "merchant": "Habbib's", "amount": 90, "time": "2019-02-13T11:00:00.000Z" } }

$ authorize < operations

{ "account": { "activeCard": true, "availableLimit": 100 }, "violations": [] }
{ "account": { "activeCard": true, "availableLimit": 80 }, "violations": [] }
{ "account": { "activeCard": true, "availableLimit": 80 }, "violations": [ "insufficient-limit" ] }
```

State

The program **should not** rely on any external database. Internal state should be handled by an explicit in-memory structure. State is to be reset at application start.

Operations

The program handles two kinds of operations, deciding on which one according to the line that is being processed:

1. Account creation
2. Transaction authorization

For the sake of simplicity, you can assume all monetary values are integers, using a currency without cents.

---

1. Account creation

Input

Creates the account with `availableLimit` and `activeCard` set. For simplicity sake, we will assume the application will deal with just one account.

Output

The created account's current state + any business logic violations.

Business logic violations

- Once created, the account should not be updated or recreated: `account-already-initialized`.