Adevinta coding challenge task

The task is to write program that computes bank balances and some other statistics based on several transaction logs.

Terminology

Bank: is an organisation that issues transactions between two accounts.
Each bank is identified by a code composed by 4 chars (e.g. B001).
All chars in a bank code are either digits between 0 and 9 or uppercase letters (`[0-9A-F]`).

Bank time zone: timezone to calculate start and end of day for a bank. Specified in format `UTCsHH:MM` where

- `s` is sign: `+` or `-`
- `HH` is hour in 24-hour format
- `MM` is minutes

For example `UTC+02:00` or `UTC-07:00`.

Account: accounts are registered in bank and identified by UUID string `BBBBxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx`
where `BBBB` is bank code and `x` is character from range `[0-9A-F]`.

Transaction: is a money transfer operation between two *accounts*. Each transaction have a date,
a source account id, a destination account id, a currency and an amount.

Outgoing transaction for the bank `B001`: if the *transaction* source account belongs to the bank `B001`
and the destination belongs to a different bank such transaction is called *outgoing*.

Incoming transaction for the bank `B001`: if the *transaction* destination account belongs to the bank `B001`
and the source belongs to a different bank such transaction is called *incoming*.

Internal transaction for the bank `B001`: if both source and destination accounts belong to the same bank `B001`
such transaction is called *internal*.

Program parameters

The program must accept two parameters: `<transactions folder> <result folder>`.
Both parameters are folders. `<transactions folder>` must exist and must not be empty.
`<result folder>` must not exist.

Transactions folder content

The transactions folder must contain following files:

- A transaction log index `transactions.csv` with list of all transaction logs meta-

information.
- Several BBBB_DDDD.csv files containing transaction logs from bank BBBB. Each bank may submit its logs in separate files.

transactions.csv file content

This CSV file contains the list of all transaction logs successfully submitted by banks. File format is:

```
DATE,BANK-CODE,BANK-TIME-ZONE,TRANSACTION-LOG-FILE-PATH,BANK-NAME
```

Where:

- **DATE**: ISO 8601 date of log submission.
- **BANK-CODE**: 4-symbol bank code.
- **BANK-TIME-ZONE** : main timezone of a bank.
- **TRANSACTION-LOG-FILE-PATH**: path to transaction log submission.
  Path is relative to `<transactions folder>`.
- **BANK-NAME**: human readable bank name.

For example:

```
2019-01-23T23:48:12+01:00,B001,UTC+02:00,./B001_001.csv,Confidence
2019-01-23T20:01:45+01:00,B002,UTC-07:00,./B002_001.csv,Alex_brothers
```

Transaction log submission format BBBB_DDDD.csv

A CSV file with transaction details acknowledged by the bank BBBB.
It is guaranteed that all transactions submitted by a bank are uniq.
But it's possible the same transaction can be submitted by two different banks:
by one bank as *outgoing* and by other one as *incoming*.

File format is:

```
DATE,TRANSACTION-ID,TRANSACTION-SOURCE,TRANSACTION-
DESTINATION,AMOUNT,CURRENCY,CATEGORY
```

Where:

- **DATE**: ISO 8601 date of transaction.
- **TRANSACTION-ID**: UUID of transaction.
- **TRANSACTION-SOURCE**, **TRANSACTION-DESTINATION**: account IDs of source and destination of transaction.
- **AMOUNT**: amount of money transferred by transaction.
- **CURRENCY**: ISO 4217 currency code (e.g. EUR or USD).
- **CATEGORY**: some internal for this bank identifier of transaction category.
  String with character from range `[0-9A-Za-z_.-]`, for example `taxi`, `groceries` or `spendings_01`.

For example:

```
2019-05-01T19:52:01+02:00,12ACEE28-65AC-11E9-B427-D3857AC0CCC7,B0011DBB-590B-
4D55-9C2C-B82E54C37E92,B002E425-6454-424B-BB67-
2C8303A9837D,100.0,EUR,Category01
```

Result folder content

This folder must contain the result files of the execution of the program, concretely:

- `banks.csv`, containing the list of all banks involved in transactions
- `BBBB_daily_balance.csv`, containing banks daily transactions statistics
- `BBBB_categories.csv`, containing all bank transaction statistics per category

List of all banks banks.csv

A CSV file with that contains the list of all banks involved in any transaction
using the following format:

```
BANK-CODE,BANK-NAME
```

Where:

- **BANK-CODE**: 4-symbol bank code.
- **BANK-NAME**: human readable bank name. Should be string `UNKNOWN` if the bank name is not known.

The file should be ordered by `BANK-CODE` in lexicographic order.

For example:

```
B001,Confidence
B002,Alex_brothers
B003,UNKNOWN
```

Bank daily balances BBBB_daily_balance.csv

A CSV file containing the daily transaction statistics for the bank `BBBB`.
The bank statistics must be calculated only using transactions submitted by this bank.
The statistics must be calculated for *incoming* and *outgoing transactions* only,
all *internal transactions* must be skipped.
Days should be calculated in *bank timezone*.

The format for this file is:

```
DAY,CURRENCY,TOTAL-OUTGOING-AMOUNT,OUTGOING-TRANSACTION-COUNT,TOTAL-INCOMING-AMOUNT,INCOMING-TRANSACTION-COUNT
```

Where:

- **DAY**: ISO 8601 day in *bank timezone*.
- **CURRENCY**: ISO 4217 currency code.
- **TOTAL-OUTGOING-AMOUNT**: total amount of all *outgoing transactions* during that day.
- **OUTGOING-TRANSACTION-COUNT**: number of *outgoing transactions* during that day.
- **TOTAL-INCOMING-AMOUNT**: total amount of all *incoming transactions* during that day.
- **INCOMING-TRANSACTION-COUNT**: number of *incoming transactions* during that day.

File must be ordered by `DAY` and `CURRENCY` in lexicographic order.

For example `B001_daily_balance.csv`:

```
2019-05-01,EUR,15643.32,1043,943.09,571
2019-05-02,EUR,745.86,1347,2467.45,1803
```

Transaction statistics for categories BBBB_categories.csv

A CSV file containing the list of all known transaction categories for bank BBBB and statistics for all transactions.

The format for this file is:

```
CATEGORY,CURRENCY,TOTAL-AMOUNT,TRANSACTION-COUNT
```

Where:

- **CATEGORY**: transaction category.
- **CURRENCY**: ISO 4217 currency code.
- **TOTAL-AMOUNT**: total amount of *all transactions* in that category.
- **TRANSACTION-COUNT**: number of *all transactions* in that category.

File must be ordered by `CATEGORY` and `CURRENCY` in lexicographic order.

For example:

```
groceries,EUR,392.74,159
spendings_01,USD,15934.03,4578
taxi,EUR,4383.04,332
```

Program requirements

Input data limits

Input data is limited:

- There are no more than 100 banks
- Each bank submits less than 100 transaction logs
- Each transaction log is less than 10000 transactions
- Each bank have less than 1000 different categories

Challenge submissions requirements

- Coding challenge should be submitted as *ZIP* folder
- There are should be full source code for the program. It's better to remove all compiled binary artifacts.
- Solution should be tested and unit-tests should included
- Submission should have `README.MD` file describing:
    - How to compile and run program
    - How to run tests
    - Some description of the algorithms and decisions taken during implementation