

# Bank model

## Introduction

This assignment should take about **2-4 hours**. The test can be done in any statically typed object-oriented language such as Java. You will probably need some kind of library to write unit tests, your choosing. If you feel like there is something that is unclear with the test, you are free to make any assumptions by yourself, but please document them.

## Assignment

We work with a lot of bank information. To give you a sense of how that is, we want you to create models for that kind of data. On the next page, you will find some example data of a bank account. A bank account holds transactions which consist of transfers and payments. Payments have a recipient number, while transfers don't. Card transactions only consist of payments. When you create the models have in mind that they should be **extendable** with other types of accounts and transactions.

## Tasks

1. Implement a **model** to represent the bank data.
2. Create a function that given many bank accounts (with their transactions) as input, **excludes all transactions with a positive amount** and outputs only the bank accounts that still have at least one transaction, also returns the remaining events in the output.
  - a. Create at least one unit test for this.
3. Create a function that **calculates the balance** of a bank account.
  - a. Create at least one unit test for this.
4. Build a function to **detect the time interval** between subsequent transactions for a specific group (by text). You should be able to detect monthly and biweekly (every other week) intervals and represent that with appropriate data types.
  - a. Create a unit test with the example bank account and detect that "Video streaming" occurs monthly.
  - b. Create a unit test with the example bank account and detect that "Gym" occurs biweekly. The transaction at 2023-06-28 should be considered as noise.

# Example data

## bank account

Identifier: d65c1be2-0e07-4ec8-ba44-60dfbe8fef77

User Identifier: 15d88afa-d5b9-4bf4-9b91-5333ebb6f139

### Transactions

Date	Text	Type	Recipient number	Amount
2023-08-01	Gym	Payment	123-456	-200
2023-07-23	Video streaming	Transfer		-99
2023-07-18	Gym	Payment	123-456	-200
2023-07-04	Gym	Payment	123-456	-200
2023-06-28	Gym	Payment	123-456	-50
2023-06-25	Salary	Transfer		10000
2023-06-22	Video streaming	Transfer		-99
2023-06-20	Gym	Payment	123-456	-200
2023-05-23	Video streaming	Transfer		-99

## Credit card

Number: \*\*\*\*-\*\*\*\*-\*\*\*\*-1234

User Identifier: 15d88afa-d5b9-4bf4-9b91-5333ebb6f139

### Transactions

Date	Text	Type	Amount
2023-08-01	Pizza	Payment	70
2023-07-25	Bar	Payment	120
2023-07-20	Grocery store	Payment	99

# Submission

Send us a tar or zip archive with your code and relevant project files.