# Introduction

We want to understand how you think as a programmer, and the level of craft you bring to bear when building software.

Please note that not following the below instruction will result in an automated rejection.

Rules

1. You have 2 hours to implement a solution, do try your best.
2. **Your solution does not have to be completed but must be a working piece of software.**
3. We are really interested in your object oriented or functional design skills, so please craft the most beautiful code you can.
4. We are also interested in understanding how you make assumptions when building software.
5. You have to solve below problem in **Java** **without using any external libraries** to the core language except for a testing library for TDD.
6. **Use GIT for version control and we expect you send us a standard zip or tarball of your code including GIT metadata.**
7. **Write comprehensive unit tests. Submitting your solution without tests will result in a rejection.**
8. **Your solution should build and run on Linux/MacOS.**
9. We interact with your solution via a simple set of commands which produce a specific output. Hence, your solution should provide us with an interactive command prompt-based shell where commands can be typed in. A command can be accompanied with one or more input parameters. Please take a look at the example below.

# Problem statement

As a payment service company, we want to build a software solution that provides customers with bill payment service.

1. Each customer is able to add fund into his account.
2. He is able to create, delete, update, view and search for a bill of particular service.
3. He is able to pay a valid bill using his available fund.
4. He is able to pay multiple bills of different service providers any time using his available fund. Payment would be prioritized for bill with early due dates.
5. He is also able to keep track of his bill due dates so that he is able to pay his bills in time.
6. He has an ability to pay multiple and/or different bills at the same time.
7. He desires a possibility of scheduled bill payment so that the software solution will automatically do bill payment with a schedule that he has configured.
8. He often checks payment transaction history to ensure that there is nothing wrong with his fund as well.

# Examplex

The following is sample of expected output when running your solution:

$ path/to/your\_solution\_programm CASH\_IN 1000000

Your available balance: 1000000

$ path/to/your\_solution\_programm LIST\_BILL

Bill No. Type Amount Due Date State PROVIDER

1. ELECTRIC 200000 25/10/2020 NOT\_PAID EVN HCMC

2. WATER 175000 30/10/2020 NOT\_PAID SAVACO HCMC

3. INTERNET 800000 30/11/2020 NOT\_PAID VNPT

$ path/to/your\_solution\_programm PAY 1

Payment has been completed for Bill with id 1.

Your current balance is: 800000

$ path/to/your\_solution\_programm PAY 10

Sorry! Not found a bill with such id

$ path/to/your\_solution\_programm PAY 2 3

Sorry! Not enough fund to proceed with payment.

$ path/to/your\_solution\_programm DUE\_DATE

Bill No. Type Amount Due Date State PROVIDER

2. WATER 175000 30/10/2020 NOT\_PAID SAVACO HCMC

3. INTERNET 800000 30/11/2020 NOT\_PAID VNPT

$ path/to/your\_solution\_programm SCHEDULE 2 28/10/2020

Payment for bill id 2 is scheduled on 28/10/2020

$ path/to/your\_solution\_programm LIST\_PAYMENT

No. Amount Payment Date State Bill Id

1. 200000 25/10/2020 PROCESSED 1

2. 175000 30/10/2020 PENDING 2

3. 800000 30/11/2020 PENDING 3

$ path/to/your\_solution\_programm SEARCH\_BILL\_BY\_PROVIDER VNPT

Bill No. Type Amount Due Date State PROVIDER

3. INTERNET 800000 30/11/2020 NOT\_PAID VNPT

$ path/to/your\_solution\_programm

Good bye!