# **Transactions Processor**

### **Summary**

Suppose you are part of a scrum team developing a component “Transaction Processor” that process financial transactions.

The Transaction Processor allows users to import financial transactions (each transaction is either of type debit or credit), it shall handle transactions in different formats; namely CSV (comma-separated values) and XML. Once the transactions are imported, the system can validate them and reports back any violation; Violation is defined in case that file contains an invalid transaction record e.g. Invalid transaction type, missing amount...etc. The component can also check if the loaded transactions are balanced or not; the balanced transactions means that sum of Credit transactions must equals the sum of Debit transactions

### **Implementation Details**

A transaction consists of the following:

* **type**: whether the transaction is debit or credit, the value will be D for debit or C for credit.
* **amount**: the amount of the transaction, the amount should be > 0, there is a test case that will return a violation when the amount is 0.
* **narration**: a description/purpose of the transaction

##### **Supported Formats**

The CSV format is <type>,<amount>,<narration>, see the example below:

D,61.22,Electricity bill

D,52.14,Social security payment

D,200.00,Payment sent to x

C,1920.00,Salary

D,150.00,Car rental

hint: each line represnts a single transaction.

Following is a sample for the XML format:

<TransactionList>

<Transaction type="D" amount="61.22" narration="Electricity bill" />

<Transaction type="D" amount="52.14" narration="Social security payment" />

<Transaction type="D" amount="200.00" narration="Payment sent to x" />

<Transaction type="C" amount="1920.00" narration="Salary" />

<Transaction type="D" amount="150.00" narration="Car rental" />

</TransactionList>

##### **Implementation**

You are provided with the interface TransactionProcessor that encapsulates the required operations. You are expected to write two implementations for it; one that handles the CSV format and the other for the XML.

The TransactionProcessor interface has four methods:

* importTransactions: this method accepts a java.util.InputStream, you should read this stream and parse its content into a java.util.List of Transaction (this class is provided for you)
* getImportedTransactions: this method returns the transactions list prepared by the importTransactions method
* validate: this method validates each imported transaction and returns the errors as a List of Violations (this class is provided for you). Each Violation contains the name of the property that has the error (e.g. type, amount...), a description of the error and the order of the transaction that has this error.
* isBalanced: this method returns true if the amount sum of all the debit transactions equals the amount sum of all the credit transactions.

##### **JUnit tests**

JUnit is a framework to help you test your code and make sure that each piece of code works as intended.

You are provided with two JUnit test classes (for the CSV and XML implementations) that will help you understand what is required from you and help you test your work.

##### **Maven**

Maven is a software project management tool that helps you develop and build your project.

The Java project provided to you is a maven project, no prior knowledge of maven is required. You can simply import the project to eclipse using the following steps:

* Open Eclipse
* Go to File --> Import...
* Select Maven --> Existing Maven Projects and click Next
* Browse to the project directory and click OK
* Click Finish

##### **Delivery**

You should create two implementations of the TransactionProcessor interface, one that handles XML and another for CSV.

The TransactionProcessor interface should not be modified.

Update the setup methods of the Test classes XmlTransactionProcessorTest and CsvTransactionProcessorTest to call your implementations, and do not update anything else on these classes.

**You should deliver a full maven project, just as it is provided for you.**

##### **Evaluation Criteria:**

* We will run the provided unit tests to verify that your delivered code is working and **all tests should pass**.
* Code quality and simplicity is required so try to find the simplest solution with clean code style .