Project: MoneyService/Site
Team Center

CLIHelper

~menuInput(): int
~orderRequest(): Order

Responsibilities

OrderGenerator

- MAX_RAND: int
- MIN_RAND: int

+ generateOrders(int): List<Order>
+ generateOrder(): Order
- fillCurrencies():

Responsibilities:

Randomly generates orders.
Use only for testing.

«Type»
TransactionMode

BUY, SELL

Responsibilities

Defines if the Order/Transaction is a
Sell or Buy type.

MoneyServiceApp

~ currencyMap: Map<String, Currency>
~ inventoryMap: Map<String, Double>
~ referenceCurrencyCode: String
~ orderAmountLimit: int
- exchangeOffice: ExchangeOffice

+ main(String[] args)
- configure()
- createExchangeOffice()
- CLIApplication():

Responsibilities
Application trigger and configuration

ServiceConfig

~CONFIG_FILE: String
~ BUY_RATE: double
~ SELL_RATE: double

+readCurrencyConfigFile()
+readProjectConfigFile()
+readMoneyServiceConfigFile(): int

Responsibilities
Applicationi configuration is set up
from a passed file.
Currently using
CurrencyConfig_{YEAR}-{MONTH}-{DAY}.txt &
ProjectConfig.txt

readCurrencyFile should fill
MoneyServiceApp.currencyMap.

«Interface»
MoneyService

+ buyMoney(Order orderData): boolean
+ sellMoney(Order orderData): boolean
+ printSiteReport(String destination):
+ shutDownService(String destination):
+ getCurrencyMap(): Map<String, Currency>
+ getAvailableAmount(String currencyCode): Optional<Double>

Responsibilities

Rules for Money Services.

ExchangeOffice

- name: String
- completedTransactions: List<Transaction>
- inventory: Map<String currencyCode, double amount>

+ ExchangeOffice(String, Map<String, double>)

- validateOrder(Order orderData): boolean

Responsibilities

Defines a exchange office
within the system.

«Value Type»
Order

~ orderNrCounter: int

- orderNr: int
- transactionMode: TransactionMode
- currencyCode: String
- amount: int

+ Order(OrderType, String, double)
+ getOrderNr():int
+ getTransactionMode(): TransactionMode
+ getCurrencyCode(): String
+ getAmount(): int

Responsibilities
Holding information about
the requested exchange
and generate a unique
identification of the order.

«Value Type»
Transaction

- serialVersionUID: Long
- id: int
- timeStamp: LocaleDateTime
- currencyCode: String
- amount: int
- transactionMode: TransactionMode
- uniqueId: int

+ Transaction(LocaleDateTime, String, int, TransactionMode)
+ getTimeStamp: LocalDateTime
+ getCurrencyCode: String
+ getAmount(): int
+ getTransactionMode: TransactionMode
+ getId(): int

Responsibilities
Represents a completed order. When order is
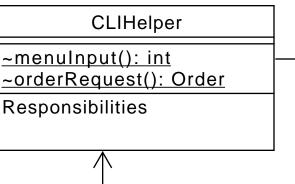validated, a transaction is created.

Currency

- currencyCode: String
- exhangeRate: double

+ Currency(String, double)
+ getExhangeRate(): double
+ getCurrencyCode(): String
+ setExhangeRate(double):

Responsibilities

«Creates»
«Uses»
«Creates»
1
1..n