HeadQuarters
Companion application to ExchangeSite

Authors:
Johannes Ahlström
Karl Lindberg
Peter Angell
Albin Hjort
Jakob Solberg

Version 1.
Last updated 2021-05-14 by Johannes Ahlström

```
HQTheApp::HQTheApp
-logger: Logger
-fh: FileHandler
+startUp(String[] args, HQ hq, Config config, HQIO hqio): void
+statsSiteChoiceMenu(Map<String, String> filter, Config config): Map<String, String>
+statsPeriodChoiceMenu(Map<String, String> filter, HQ hq, Config config): Map<String, String>
+statsCurrencyChoiceMenu(Map<String, String> filter, HQ hq, Config config): Map<String, String>
+main(String[] args): void
-controlMoneyBox(Config config, HQIO hqio): void
+createConfiguration(Config config): void
+createProjectConfigTxt(List<String> currency, List<Integer> value, String refCurrency): void
Holds main function and provides the functionality of user input.
Contains several menus which allows the user to set filters serving as parameters for
choosing which transactions will survive the filtering process.
```

```
affix.java.project.moneyservice::Statistics
-superCurrencyMap: Map<LocalDate, Map<String, Double»
-exchangeRateMap: Map<LocalDate, List<ExchangeRate»
-sellList: List<Transaction>
-buyList: List<Transaction>
-recalculatedSell: DoubleSummaryStatistics
-recalculatedBuy: DoubleSummaryStatistics
-doubleList = new ArrayList(): List<DoubleSummaryStatistics>
-result: Double
-logger: Logger
+Statistics(Config config): ctor
+presentStatistics(Map<String, String> filter, HQ hq): void
+statClear(): void
+filterFinish(List<Transaction> reducedList, Config config): void
+getSuperCurrencyMap(): Map<LocalDate, Map<String, Double»
+setSuperCurrencyMap(Map<LocalDate, Map<String, Double» superCurrencyMap): void
+getExchangeRateMap(): Map<LocalDate, List<ExchangeRate»
+setExchangeRateMap(Map<LocalDate, List<ExchangeRate» exchangeRateMap): void
+getSellList(): List<Transaction>
+setSellList(List<Transaction> sellList): void
+getBuyList(): List<Transaction>
+setBuyList(List<Transaction> buyList): void
+getRecalculatedSell(): DoubleSummaryStatistics
+setRecalculatedSell(DoubleSummaryStatistics recalculatedSell): void
+getRecalculatedBuy(): DoubleSummaryStatistics
+setRecalculatedBuy(DoubleSummaryStatistics recalculatedBuy): void
+getDoubleList(): List<DoubleSummaryStatistics>
+setDoubleList(List<DoubleSummaryStatistics> doubleList): void
+getResult(): Double
+setResult(Double result): void
+getLogger(): Logger
+setLogger(Logger logger): void
+toString(): String
Provides the functionality for running statistics on the transactions and presenting output.
Uses the filters to reduce, and then stream to filter the rest on TransactionMode.
Extracts attributes for statistics.
```

```
affix.java.project.moneyservice::HQ
-transactionList: List<List<Transaction»
-reductionList: List<Transaction>
-fileNames: Map<String, Map<String, String»
-startDate: LocalDate
-thisYear: int
-filterDate: LocalDate
-logger: Logger
+HQ(): ctor
+runStatistics(Map<String, String> filter, Config config, HQIO hqio): void
+clearHQ(): boolean
+getTransactionList(): List<List<Transaction»
+setTransactionList(List<List<Transaction» transactionList): void
+getReductionList(): List<Transaction>
+setReductionList(List<Transaction> reductionList): void
+getFileNames(): Map<String, Map<String, String»
+setFileNames(Map<String, Map<String, String» fileNames): void
+getStartDate(): LocalDate
+setStartDate(LocalDate startDate): void
+getThisYear(): int
+setThisYear(int thisYear): void
+getFilterDate(): LocalDate
+setFilterDate(LocalDate filterDate): void
Container for filtering process and triggering statistics.
```

```
affix.java.project.moneyservice::Config
-fileList = new ArrayList(): List<String>
-exchangeRateList = new ArrayList(): List<ExchangeRate>
-currencyMap = new HashMap(): Map<String, Double>
-superCurrencyMap = new HashMap(): Map<LocalDate, Map<String, Double»
-exchangeRateMap = new HashMap(): Map<LocalDate, List<ExchangeRate»
-siteNames = { "North", "South", "East", "Center", "West", "ALL" }: String[]
-periods = { "Day", "Week", "Month" }: String[]
-siteCounter = new HashMap(): Map<String, Integer>
-BUYMARGIN = 0.0f: Float
-SELLMARGIN = 0.0f: Float
-siteTransMap: Map<String, Map<String, String»
+logger: Logger
+Config(): ctor
+Configure(HQ hq, Config config, HQIO hqio): boolean
+findTransactions(): Map<String, Map<String, String»
+findParseAndMapConfigFiles(HQIO hqio): boolean
+filterFilesOnSite(Map<String, Map<String, String» fileMap, Map<String, String> filter): Map<String, Map<String, String»
+filterFilesOnPeriod(Map<String, Map<String, String» fileMap, Map<String, String> filter, HQ hq): Map<String, Map<String, String»
+filterOnCurrency(List<Transaction> filterMe, String filter): List<Transaction>
+getFileList(): List<String>
+setFileList(List<String> fileList): void
+getExchangeRateList(): List<ExchangeRate>
+setExchangeRateList(List<ExchangeRate> exchangeRateList): void
+getCurrencyMap(): Map<String, Double>
+setCurrencyMap(Map<String, Double> currencyMap): void
+getSuperCurrencyMap(): Map<LocalDate, Map<String, Double»
+setSuperCurrencyMap(Map<LocalDate, Map<String, Double» superCurrencyMap): void
+getExchangeRateMap(): Map<LocalDate, List<ExchangeRate»
+setExchangeRateMap(Map<LocalDate, List<ExchangeRate» exchangeRateMap): void
+getSiteNames(): String[]
+setSiteNames(String[] siteNames): void
+getPeriods(): String[]
+setPeriods(String[] periods): void
+getSiteCounter(): Map<String, Integer>
+setSiteCounter(Map<String, Integer> siteCounter): void
+getBUYMARGIN(): Float
+setBUYMARGIN(Float bUYMARGIN): void
+getSELLMARGIN(): Float
+setSELLMARGIN(Float sELLMARGIN): void
+getSiteTransMap(): Map<String, Map<String, String»
+setSiteTransMap(Map<String, Map<String, String» siteTransMap): void
Provides the specialized read and filter-functionalities using stream-api. Holds the major data-base components gathered from parsing.
```

```
affix.java.project.moneyservice::HQIO
-parsedSiteName: String
-parsedDate: LocalDate
-refDate: LocalDate
-refCurrency: String
-logName = "HQLog": String
-logFormat = "text": String
-logLevel = "OFF": String
-logger: Logger
+HQIO(): ctor
+parseHQConfig(HQ hq, Config config): boolean
+readTextFiles(String filename): List<String>
+parseProjectConfig(List<String> listToBeParsed): Map<String, Double>
+parseCurrencyConfig(List<String> listToBeParsed): List<ExchangeRate>
+readReport(String filename): List<Transaction>
+getParsedSiteName(): String
+setParsedSiteName(String parsedSiteName): void
+getParsedDate(): LocalDate
+setParsedDate(LocalDate parsedDate): void
+getRefDate(): LocalDate
+setRefDate(LocalDate refDate): void
+getRefCurrency(): String
+setRefCurrency(String refCurrency): void
+getLogName(): String
+setLogName(String logName): void
+getLogFormat(): String
+setLogFormat(String logFormat): void
+getLogLevel(): String
+setLogLevel(String logLevel): void
+toString(): String
Provides the functionality of reading, parsing and interacting with the filesystem.
```

```
affix.java.project.moneyservice::Transaction
-serialVersionUID = 1L: long
-id: int
-timeStamp: LocalDateTime
-currencyCode: String
-amount: int
-mode: TransactionMode
-uniqueId = 0: int
+Transaction(Order order): ctor
+Transaction(Order order, LocalDateTime ref): ctor
+getUniqueId(): int
+setUniqueId(int newuniqueId): void
+getId(): int
+getTimeStamp(): LocalDateTime
+getCurrencyCode(): String
+getAmount(): int
+getMode(): TransactionMode
+toString(): String
The result of a completed Order*.
Allows for reading and storing transactions.
```