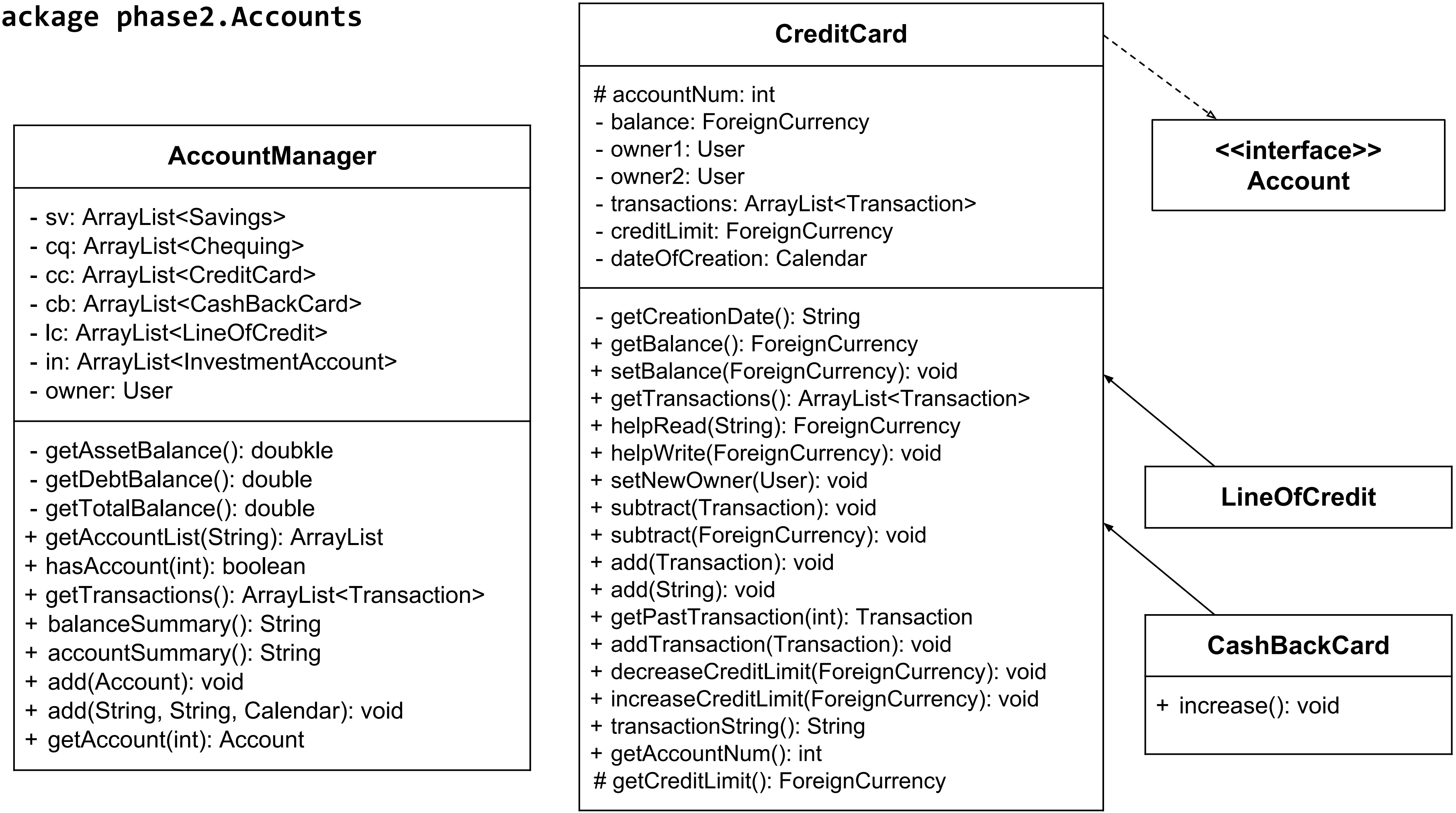
UML DIAGRAMS

group_0308

package phase2.Accounts



AccountManager

```
- sv: ArrayList<Savings>
- cq: ArrayList<Chequing>
- cc: ArrayList<CreditCard>
- cb: ArrayList<CashBackCard>
- lc: ArrayList<LineOfCredit>
- in: ArrayList<InvestmentAccount>
- owner: User
```

```
- getAssetBalance(): doubkle
- getDebtBalance(): double
- getTotalBalance(): double
+ getAccountList(String): ArrayList
+ hasAccount(int): boolean
+ getTransactions(): ArrayList<Transaction>
+ balanceSummary(): String
+ accountSummary(): String
+ add(Account): void
+ add(String, String, Calendar): void
+ getAccount(int): Account
```

CreditCard

```
# accountNum: int
- balance: ForeignCurrency
- owner1: User
- owner2: User
- transactions: ArrayList<Transaction>
- creditLimit: ForeignCurrency
- dateOfCreation: Calendar
```

```
- getCreationDate(): String
+ getBalance(): ForeignCurrency
+ setBalance(ForeignCurrency): void
+ getTransactions(): ArrayList<Transaction>
+ helpRead(String): ForeignCurrency
+ helpWrite(ForeignCurrency): void
+ setNewOwner(User): void
+ subtract(Transaction): void
+ subtract(ForeignCurrency): void
+ add(Transaction): void
+ add(String): void
+ getPastTransaction(int): Transaction
+ addTransaction(Transaction): void
+ decreaseCreditLimit(ForeignCurrency): void
+ increaseCreditLimit(ForeignCurrency): void
+ transactionString(): String
+ getAccountNum(): int
# getCreditLimit(): ForeignCurrency
```

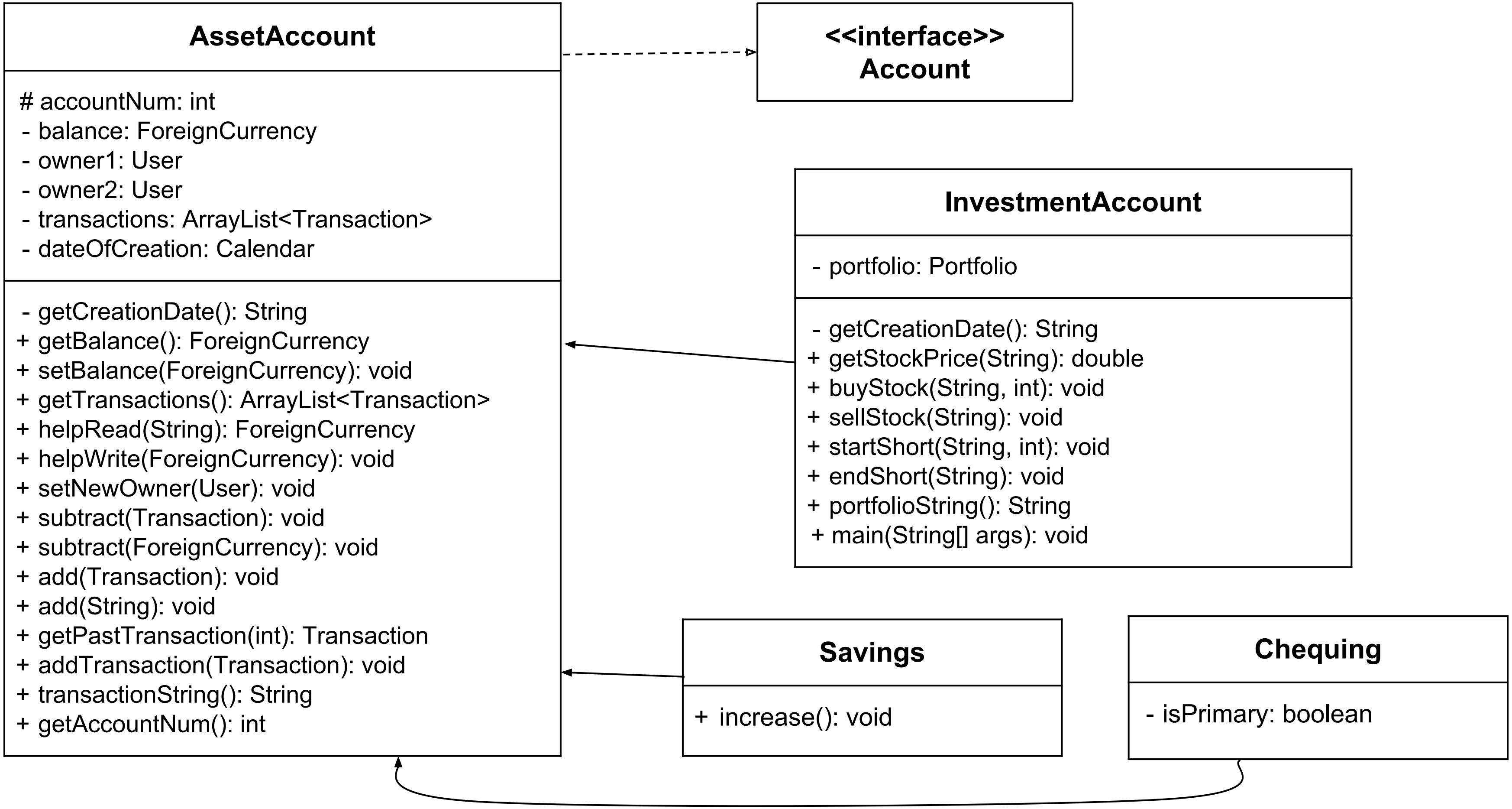
<<interface>>
Account

LineOfCredit

CashBackCard

```
+ increase(): void
```

package phase2.Accounts



AssetAccount

# accountNum: int
- balance: ForeignCurrency
- owner1: User
- owner2: User
- transactions: ArrayList<Transaction>
- dateOfCreation: Calendar

- getCreationDate(): String
+ getBalance(): ForeignCurrency
+ setBalance(ForeignCurrency): void
+ getTransactions(): ArrayList<Transaction>
+ helpRead(String): ForeignCurrency
+ helpWrite(ForeignCurrency): void
+ setNewOwner(User): void
+ subtract(Transaction): void
+ subtract(ForeignCurrency): void
+ add(Transaction): void
+ add(String): void
+ getPastTransaction(int): Transaction
+ addTransaction(Transaction): void
+ transactionString(): String
+ getAccountNum(): int

<<interface>>
Account

InvestmentAccount

- portfolio: Portfolio

- getCreationDate(): String
+ getStockPrice(String): double
+ buyStock(String, int): void
+ sellStock(String): void
+ startShort(String, int): void
+ endShort(String): void
+ portfolioString(): String
+ main(String[] args): void

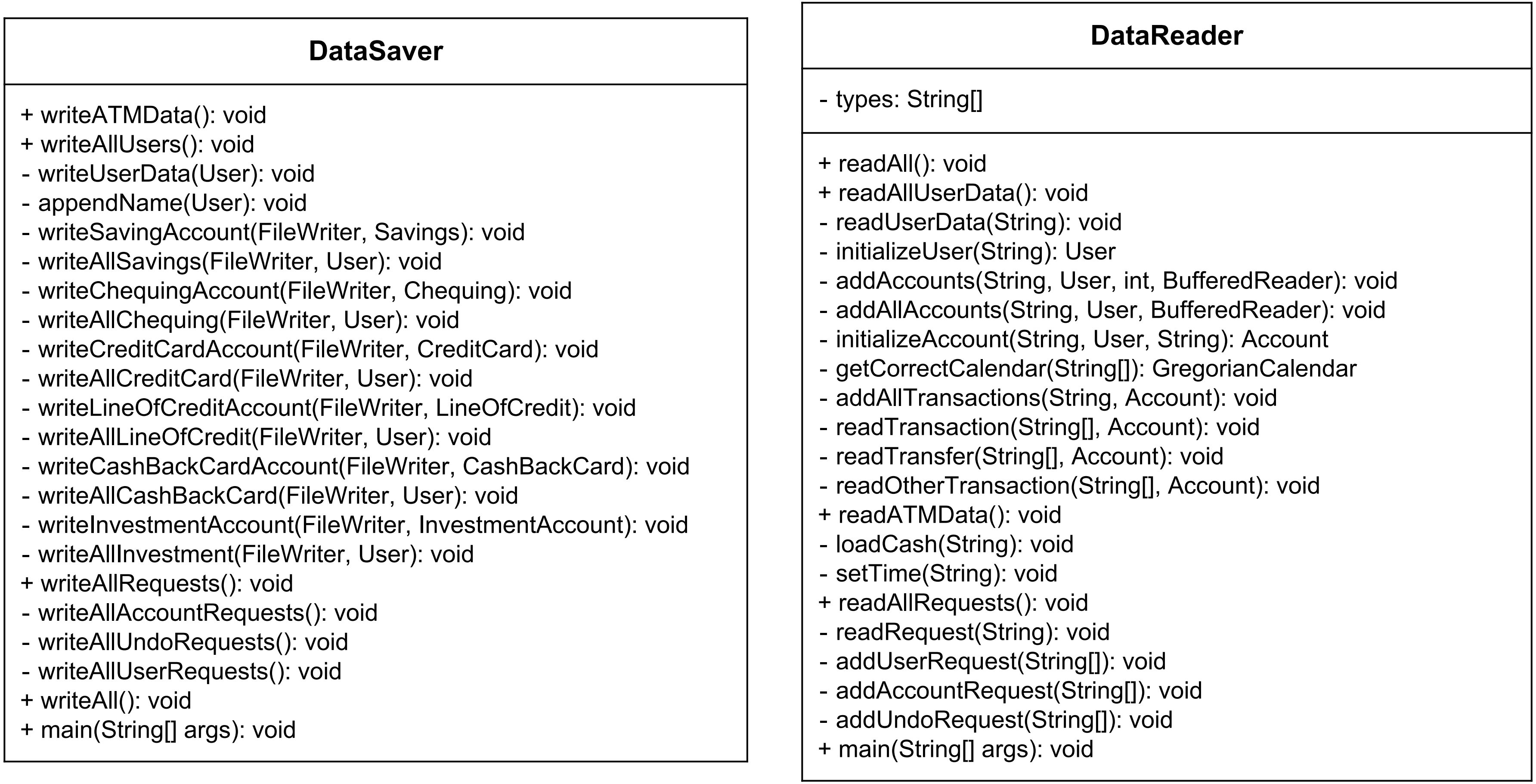
Savings

+ increase(): void

Chequing

- isPrimary: boolean

package phase2.Data



DataSaver

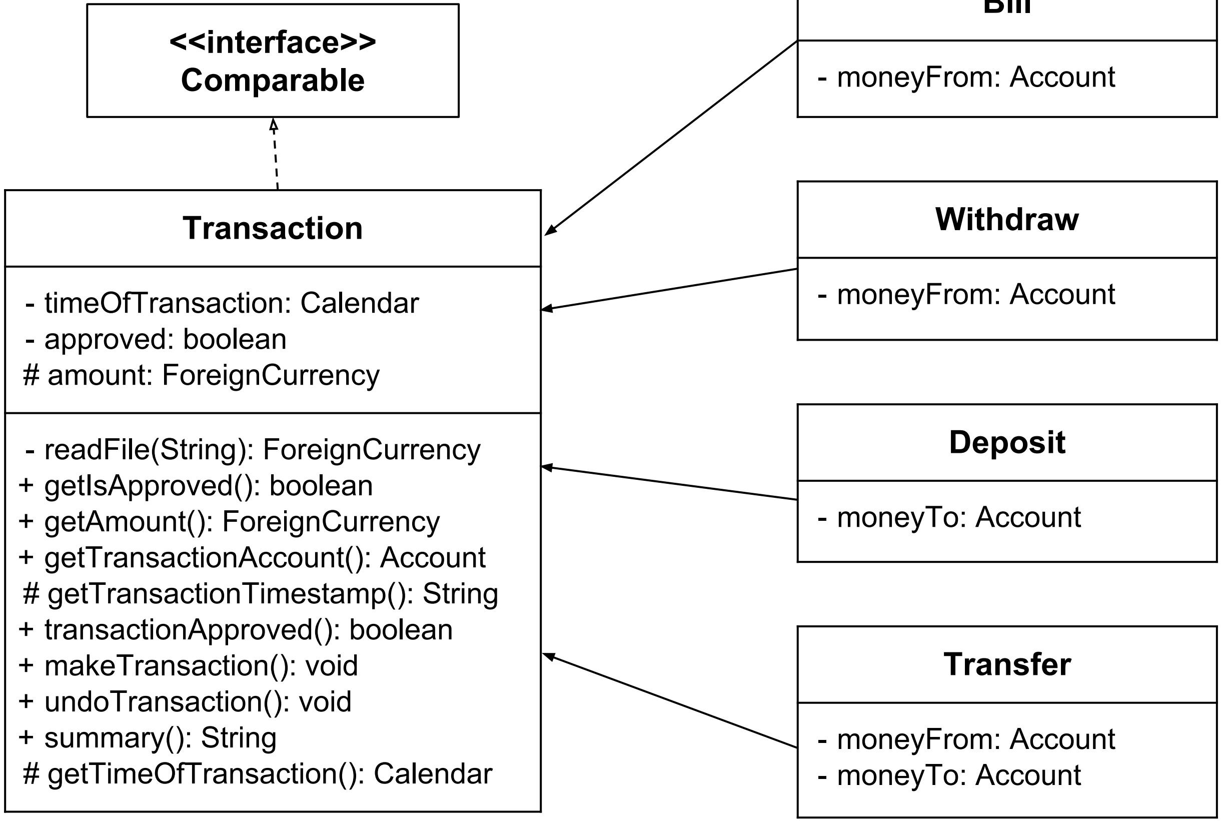
+ writeATMData(): void
+ writeAllUsers(): void
- writeUserData(User): void
- appendName(User): void
- writeSavingAccount(FileWriter, Savings): void
- writeAllSavings(FileWriter, User): void
- writeChequingAccount(FileWriter, Chequing): void
- writeAllChequing(FileWriter, User): void
- writeCreditCardAccount(FileWriter, CreditCard): void
- writeAllCreditCard(FileWriter, User): void
- writeLineOfCreditAccount(FileWriter, LineOfCredit): void
- writeAllLineOfCredit(FileWriter, User): void
- writeCashBackCardAccount(FileWriter, CashBackCard): void
- writeAllCashBackCard(FileWriter, User): void
- writeInvestmentAccount(FileWriter, InvestmentAccount): void
- writeAllInvestment(FileWriter, User): void
+ writeAllRequests(): void
- writeAllAccountRequests(): void
- writeAllUndoRequests(): void
- writeAllUserRequests(): void
+ writeAll(): void
+ main(String[] args): void

DataReader

- types: String[]

+ readAll(): void
+ readAllUserData(): void
- readUserData(String): void
- initializeUser(String): User
- addAccounts(String, User, int, BufferedReader): void
- addAllAccounts(String, User, BufferedReader): void
- initializeAccount(String, User, String): Account
- getCorrectCalendar(String[]): GregorianCalendar
- addAllTransactions(String, Account): void
- readTransaction(String[], Account): void
- readTransfer(String[], Account): void
- readOtherTransaction(String[], Account): void
+ readATMData(): void
- loadCash(String): void
- setTime(String): void
+ readAllRequests(): void
- readRequest(String): void
- addUserRequest(String[]): void
- addAccountRequest(String[]): void
- addUndoRequest(String[]): void
+ main(String[] args): void

package phase2.Transactions



<<interface>>
Comparable

Transaction

- timeOfTransaction: Calendar
- approved: boolean
# amount: ForeignCurrency

- readFile(String): ForeignCurrency
+ getIsApproved(): boolean
+ getAmount(): ForeignCurrency
+ getTransactionAccount(): Account
# getTransactionTimestamp(): String
+ transactionApproved(): boolean
+ makeTransaction(): void
+ undoTransaction(): void
+ summary(): String
# getTimeOfTransaction(): Calendar

Bill

- moneyFrom: Account

Withdraw

- moneyFrom: Account

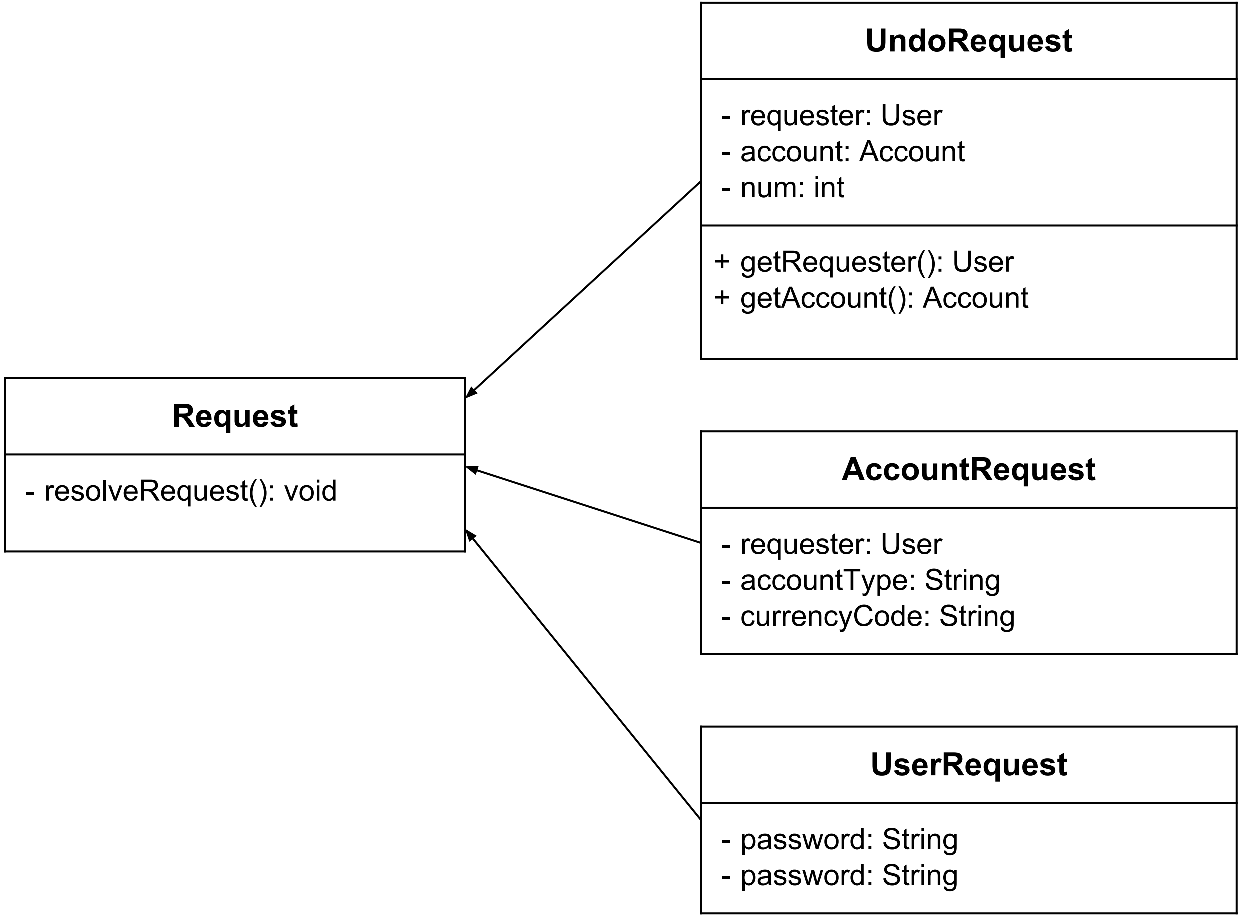
Deposit

- moneyTo: Account

Transfer

- moneyFrom: Account
- moneyTo: Account

package phase2.Request



UndoRequest

- requester: User
- account: Account
- num: int

+ getRequester(): User
+ getAccount(): Account

Request

- resolveRequest(): void

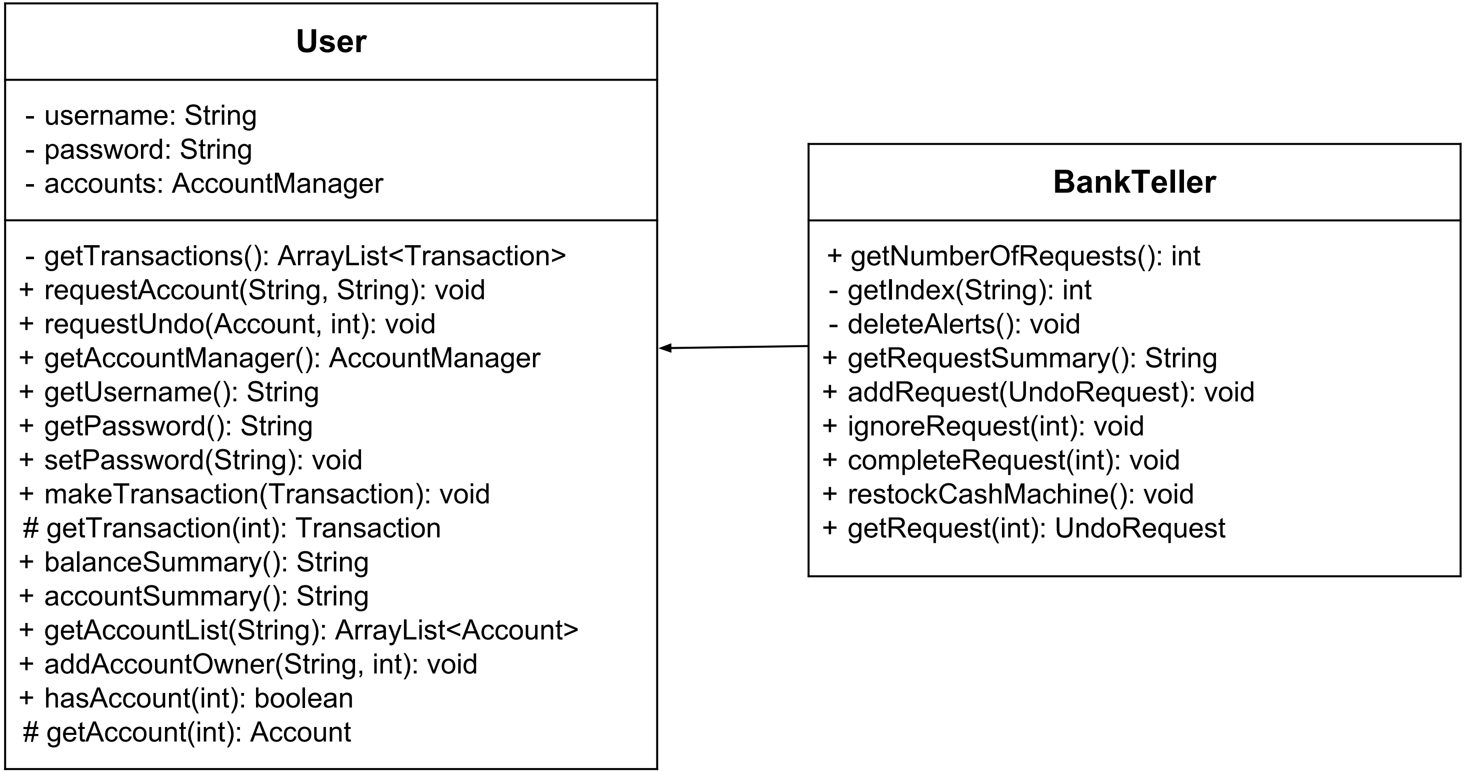
AccountRequest

- requester: User
- accountType: String
- currencyCode: String

UserRequest

- password: String
- password: String

package phase2.People



User

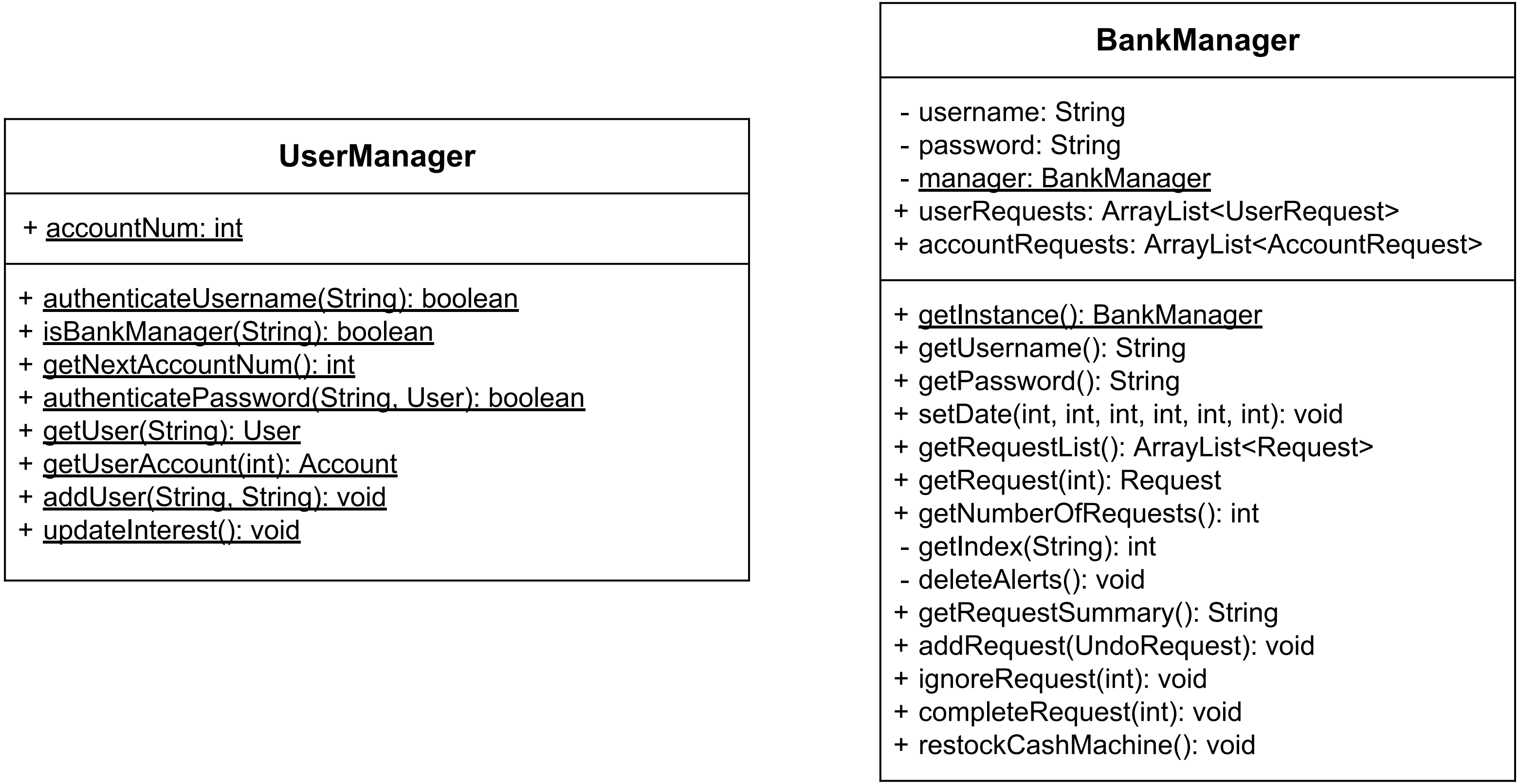
```
- username: String
- password: String
- accounts: AccountManager
```

```
- getTransactions(): ArrayList<Transaction>
+ requestAccount(String, String): void
+ requestUndo(Account, int): void
+ getAccountManager(): AccountManager
+ getUsername(): String
+ getPassword(): String
+ setPassword(String): void
+ makeTransaction(Transaction): void
# getTransaction(int): Transaction
+ balanceSummary(): String
+ accountSummary(): String
+ getAccountList(String): ArrayList<Account>
+ addAccountOwner(String, int): void
+ hasAccount(int): boolean
# getAccount(int): Account
```

BankTeller

```
+ getNumberOfRequests(): int
- getIndex(String): int
- deleteAlerts(): void
+ getRequestSummary(): String
+ addRequest(UndoRequest): void
+ ignoreRequest(int): void
+ completeRequest(int): void
+ restockCashMachine(): void
+ getRequest(int): UndoRequest
```

package phase2.People



UserManager

+ accountNum: int

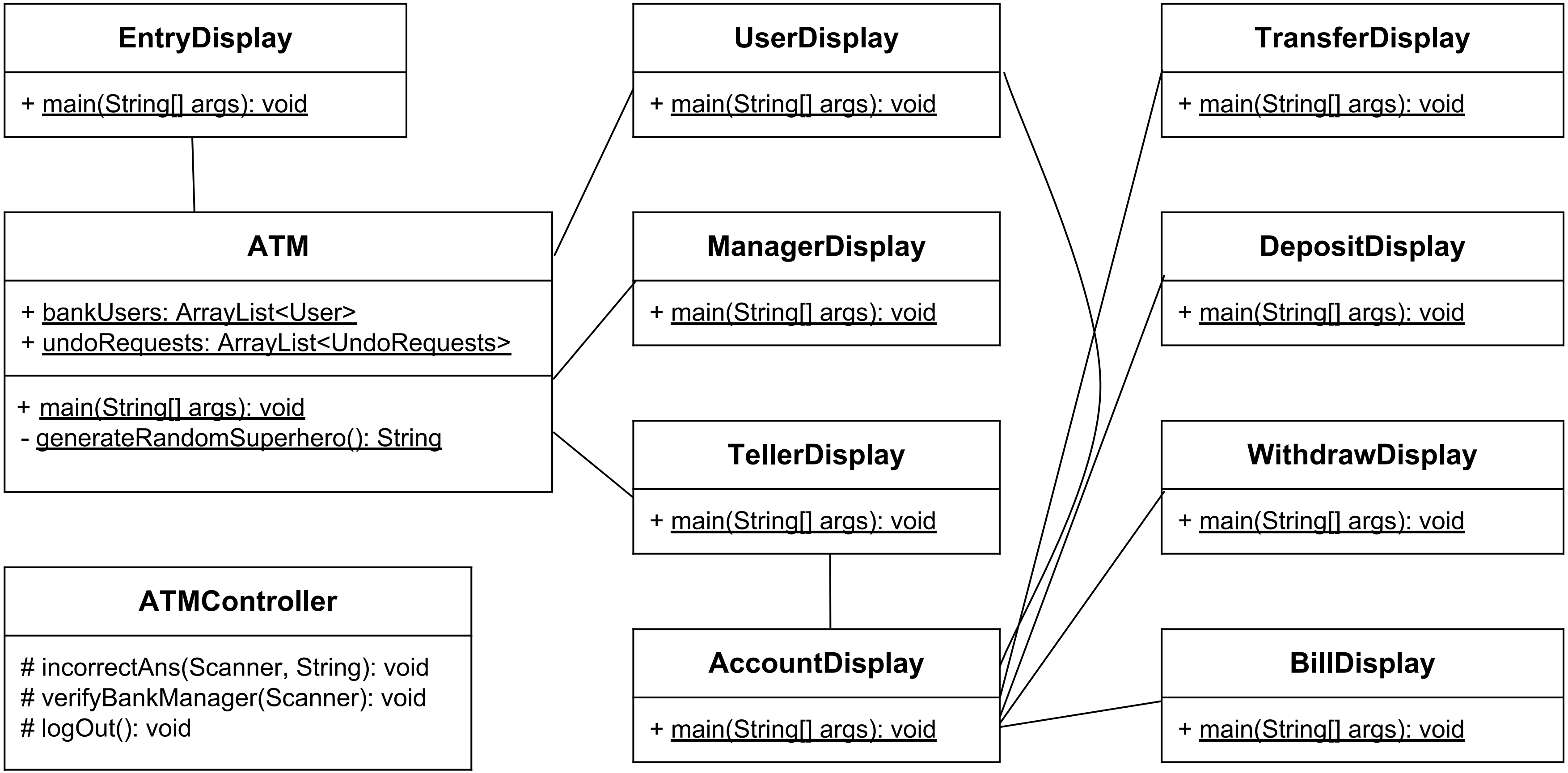
+ authenticateUsername(String): boolean
+ isBankManager(String): boolean
+ getNextAccountNum(): int
+ authenticatePassword(String, User): boolean
+ getUser(String): User
+ getUserAccount(int): Account
+ addUser(String, String): void
+ updateInterest(): void

BankManager

- username: String
- password: String
- manager: BankManager
+ userRequests: ArrayList<UserRequest>
+ accountRequests: ArrayList<AccountRequest>

+ getInstance(): BankManager
+ getUsername(): String
+ getPassword(): String
+ setDate(int, int, int, int, int, int): void
+ getRequestList(): ArrayList<Request>
+ getRequest(int): Request
+ getNumberOfRequests(): int
- getIndex(String): int
- deleteAlerts(): void
+ getRequestSummary(): String
+ addRequest(UndoRequest): void
+ ignoreRequest(int): void
+ completeRequest(int): void
+ restockCashMachine(): void

package phase2.Display



EntryDisplay

+ main(String[] args): void

ATM

+ bankUsers: ArrayList<User>
+ undoRequests: ArrayList<UndoRequests>

+ main(String[] args): void
- generateRandomSuperhero(): String

ATMController

# incorrectAns(Scanner, String): void
# verifyBankManager(Scanner): void
# logOut(): void

UserDisplay

+ main(String[] args): void

ManagerDisplay

+ main(String[] args): void

TellerDisplay

+ main(String[] args): void

AccountDisplay

+ main(String[] args): void

TransferDisplay

+ main(String[] args): void

DepositDisplay

+ main(String[] args): void

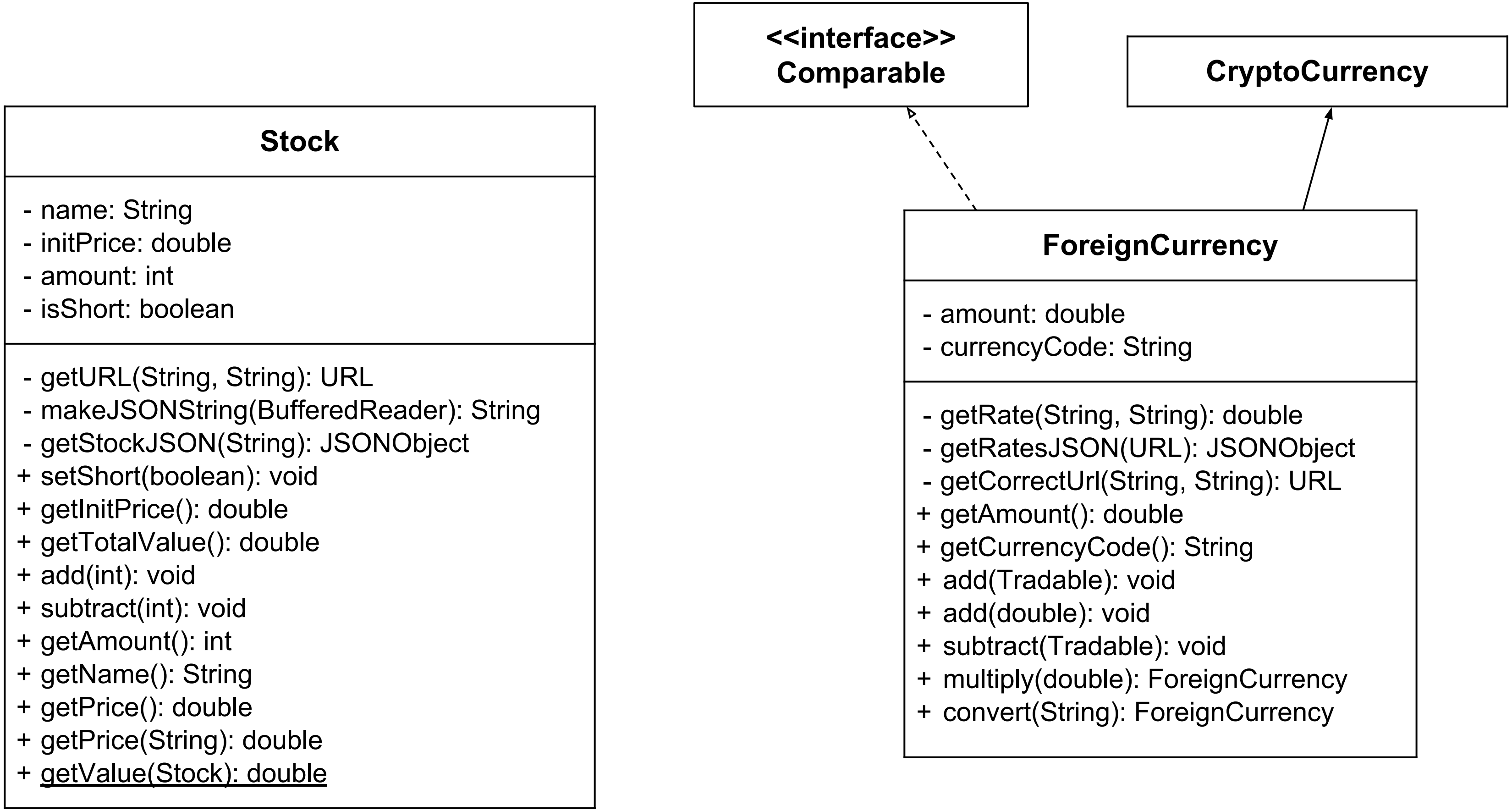
WithdrawDisplay

+ main(String[] args): void

BillDisplay

+ main(String[] args): void

package phase2.Tradable



Stock

- name: String
- initPrice: double
- amount: int
- isShort: boolean

- getURL(String, String): URL
- makeJSONString(BufferedReader): String
- getStockJSON(String): JSONObject
+ setShort(boolean): void
+ getInitPrice(): double
+ getTotalValue(): double
+ add(int): void
+ subtract(int): void
+ getAmount(): int
+ getName(): String
+ getPrice(): double
+ getPrice(String): double
+ getValue(Stock): double

<<interface>>
Comparable

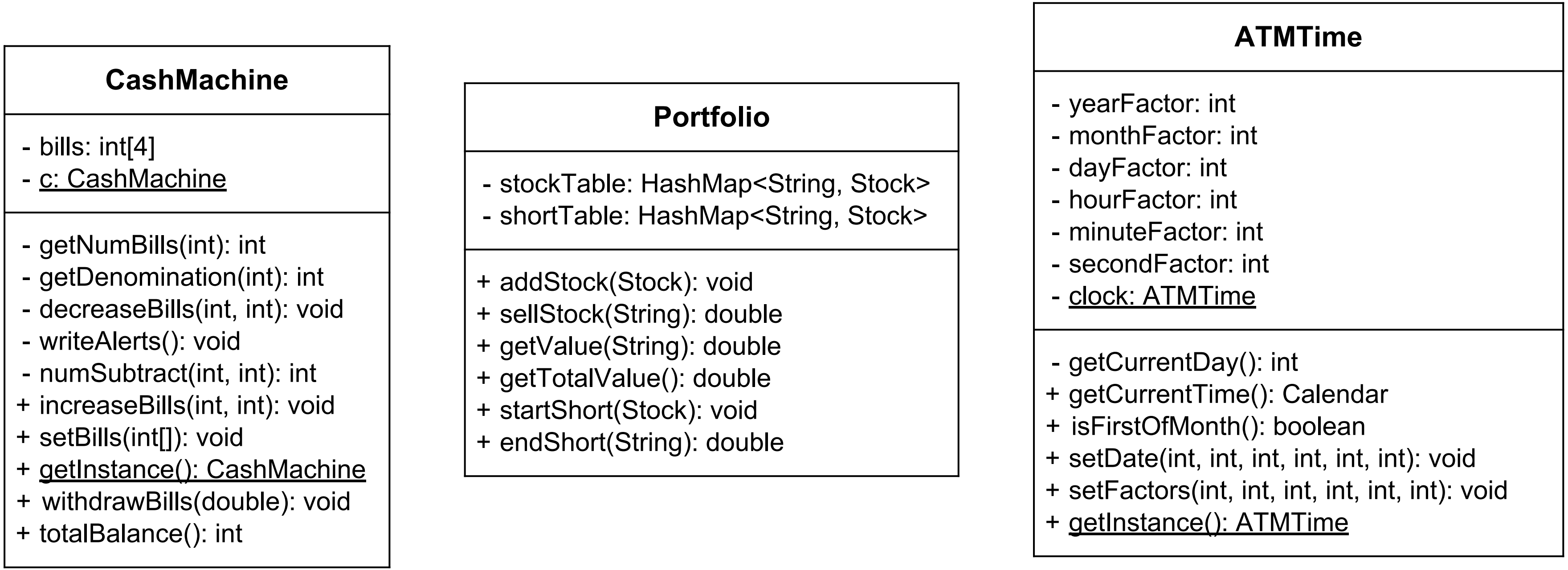
CryptoCurrency

ForeignCurrency

- amount: double
- currencyCode: String

- getRate(String, String): double
- getRatesJSON(URL): JSONObject
- getCorrectUrl(String, String): URL
+ getAmount(): double
+ getCurrencyCode(): String
+ add(Tradable): void
+ add(double): void
+ subtract(Tradable): void
+ multiply(double): ForeignCurrency
+ convert(String): ForeignCurrency

package phase2



CashMachine

- bills: int[4]
- c: CashMachine

- getNumBills(int): int
- getDenomination(int): int
- decreaseBills(int, int): void
- writeAlerts(): void
- numSubtract(int, int): int

+ increaseBills(int, int): void
+ setBills(int[]): void
+ getInstance(): CashMachine
+ withdrawBills(double): void
+ totalBalance(): int

Portfolio

- stockTable: HashMap<String, Stock>
- shortTable: HashMap<String, Stock>

+ addStock(Stock): void
+ sellStock(String): double
+ getValue(String): double
+ getTotalValue(): double
+ startShort(Stock): void
+ endShort(String): double

ATMTime

- yearFactor: int
- monthFactor: int
- dayFactor: int
- hourFactor: int
- minuteFactor: int
- secondFactor: int
- clock: ATMTime

- getCurrentDay(): int

+ getCurrentTime(): Calendar
+ isFirstOfMonth(): boolean
+ setDate(int, int, int, int, int, int): void
+ setFactors(int, int, int, int, int, int): void
+ getInstance(): ATMTime