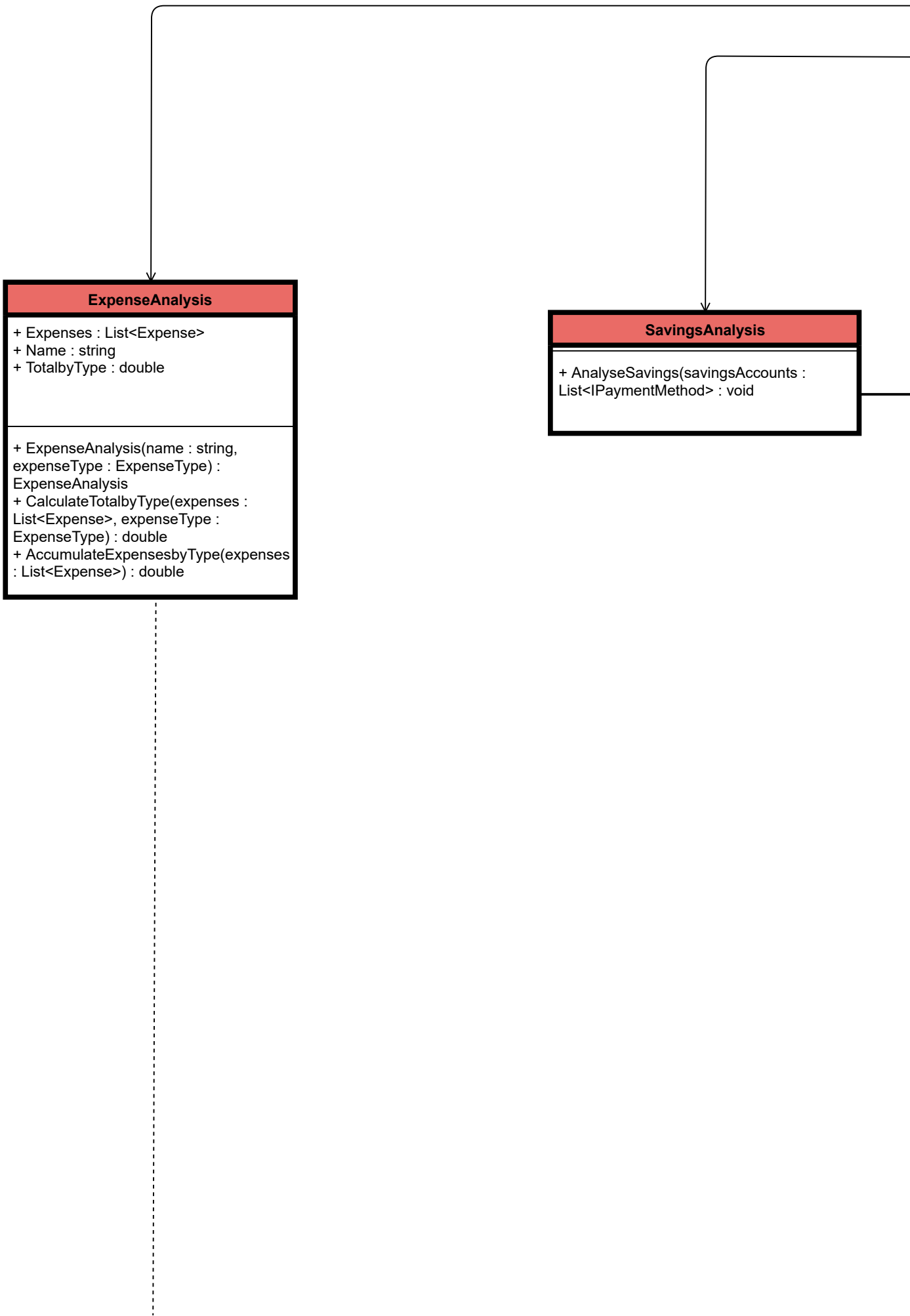
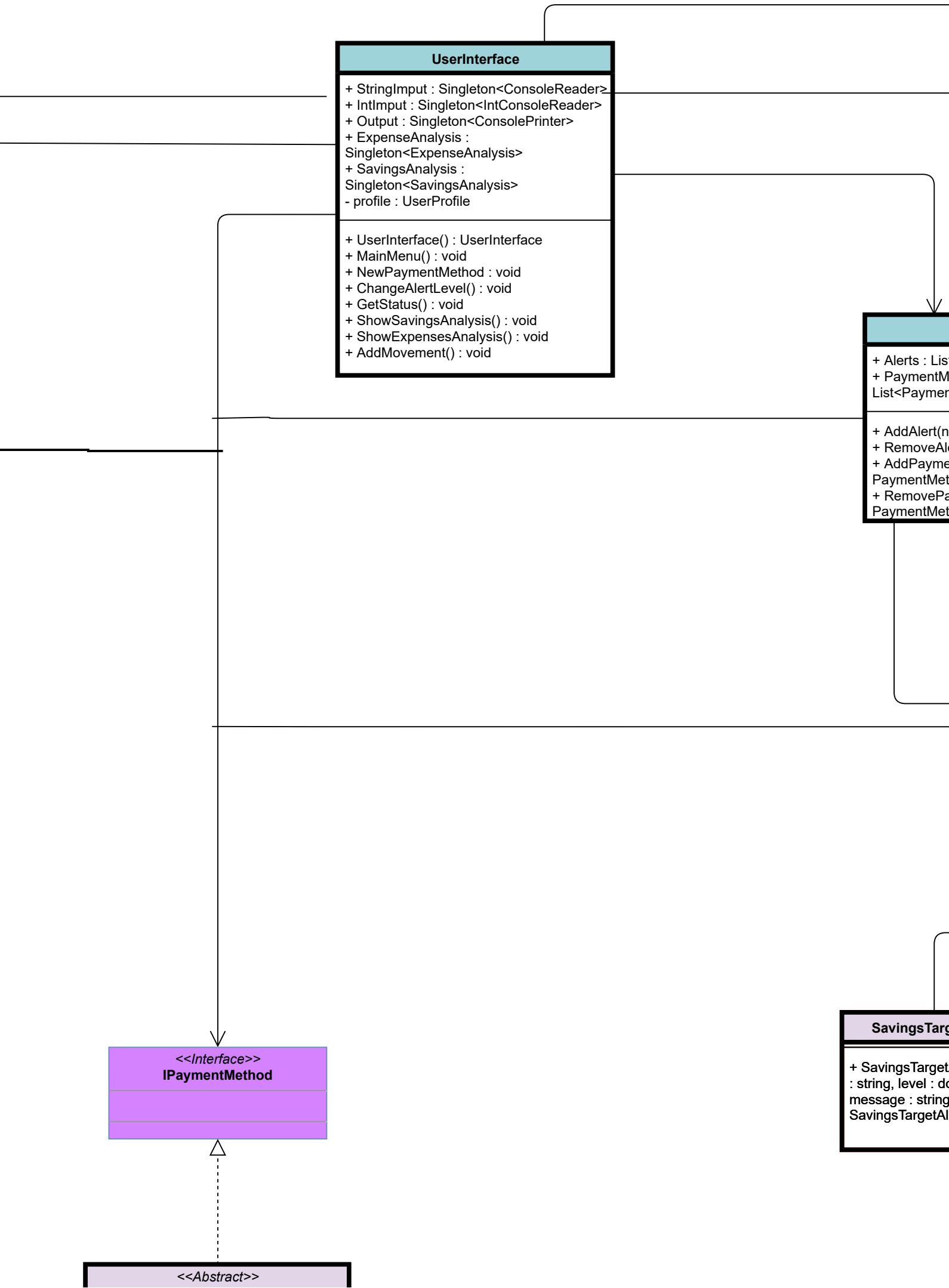
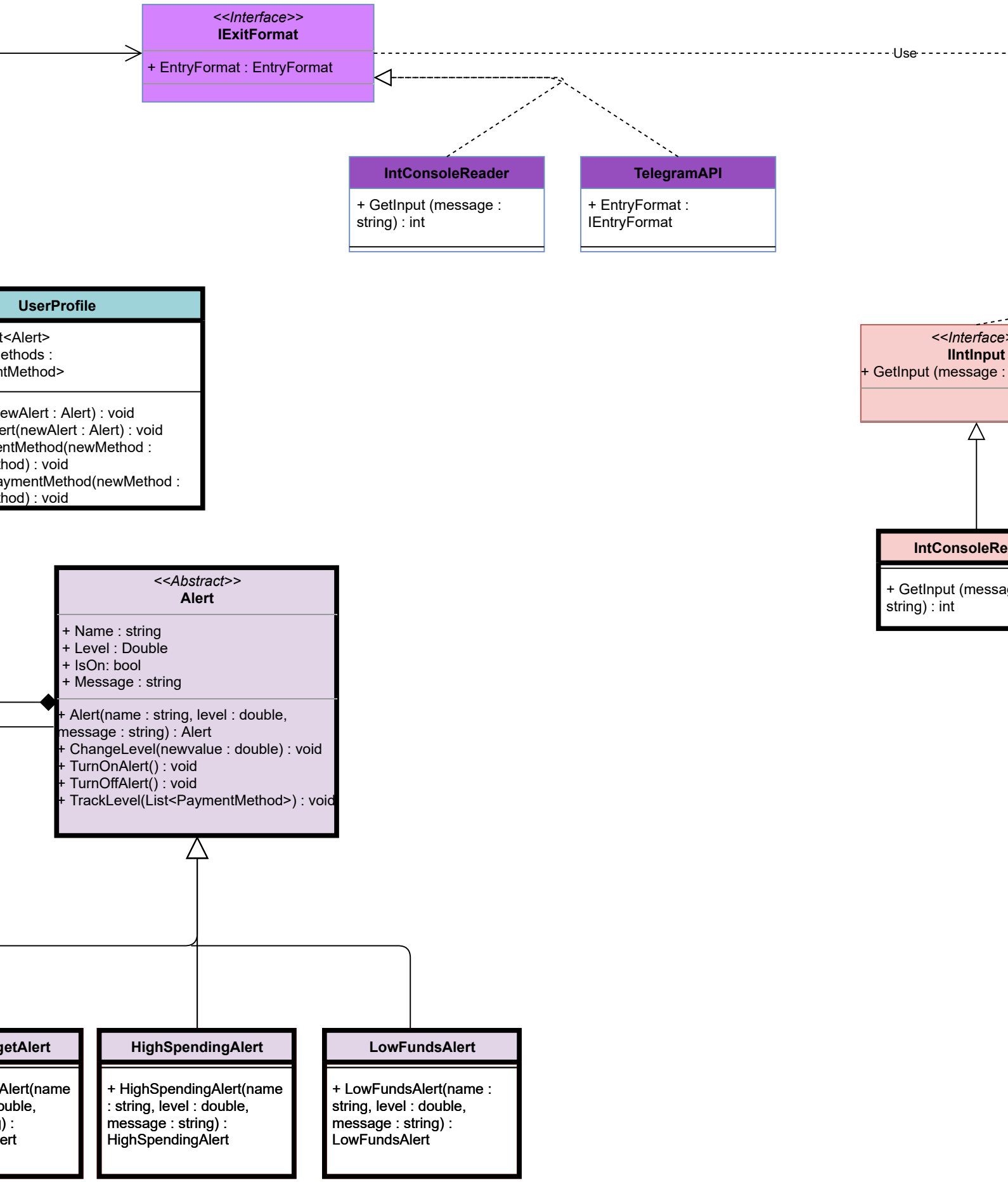
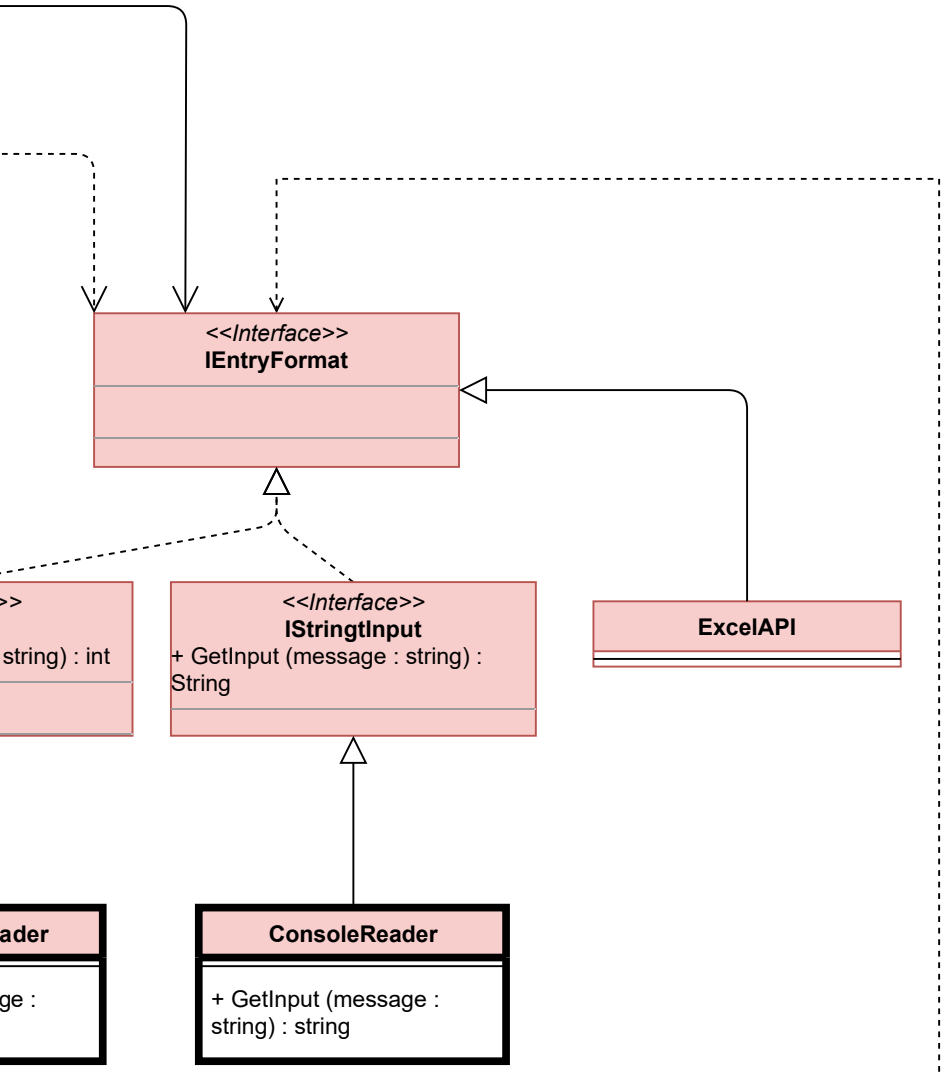
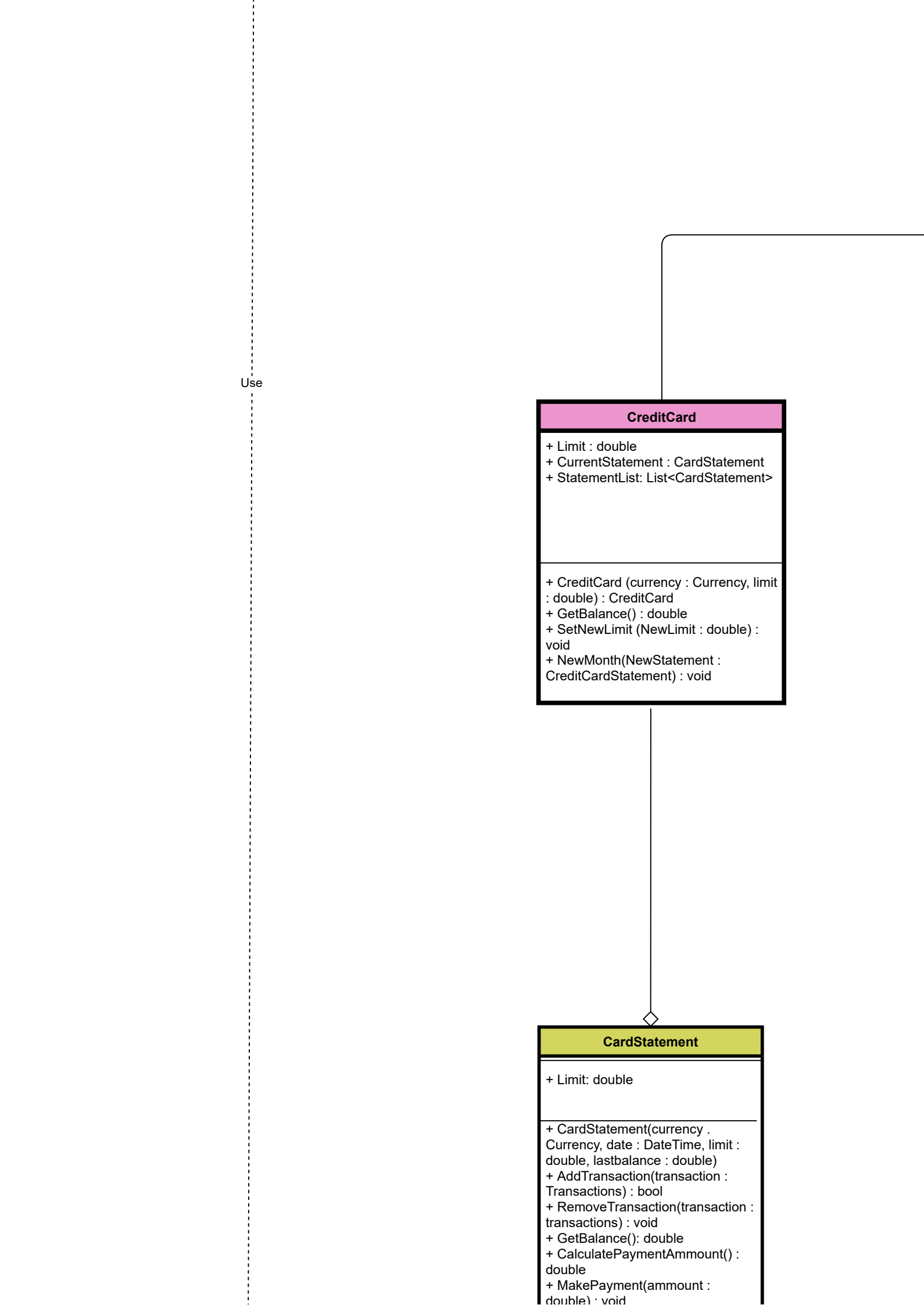
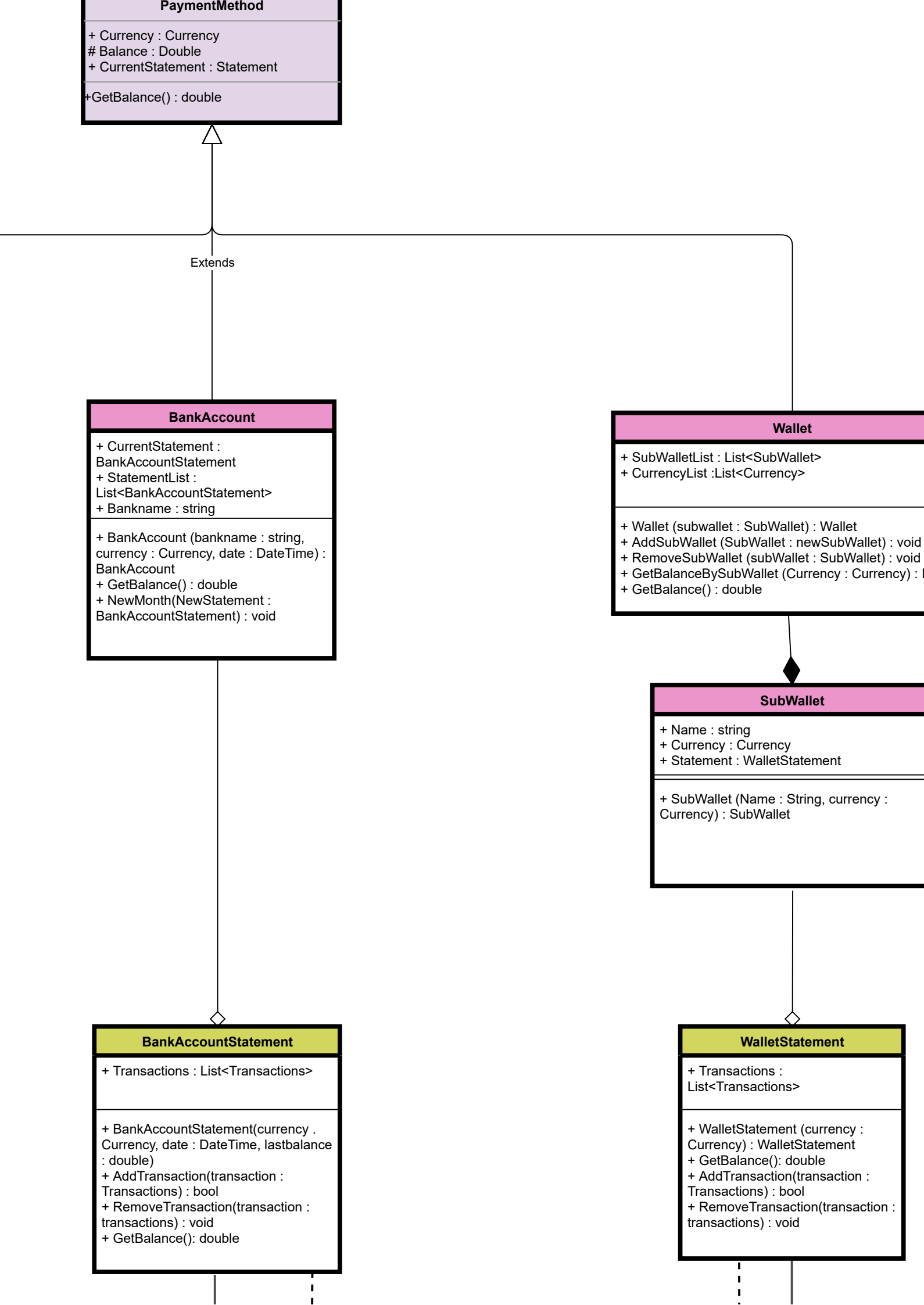

ExpenseAnalysis
+ Expenses : List<Expense>
+ Name : string
+ TotalbyType : double
+ ExpenseAnalysis(name : string, expenseType : ExpenseType) : ExpenseAnalysis
+ CalculateTotalbyType(expenses : List<Expense>, expenseType : ExpenseType) : double
+ AccumulateExpensesbyType(expenses : List<Expense>) : double
SavingsAnalysis
+ AnalyseSavings(savingsAccounts : List<IPaymentMethod> : void


UserInterface
+ StringImput : Singleton<ConsoleReader>
+ IntImput : Singleton<IntConsoleReader>
+ Output : Singleton<ConsolePrinter>
+ ExpenseAnalysis :
Singleton<ExpenseAnalysis>
+ SavingsAnalysis :
Singleton<SavingsAnalysis>
- profile : UserProfile
+ UserInterface() : UserInterface
+ MainMenu() : void
+ NewPaymentMethod : void
+ ChangeAlertLevel() : void
+ GetStatus() : void
+ ShowSavingsAnalysis() : void
+ ShowExpensesAnalysis() : void
+ AddMovement() : void
<<Interface>>
IPaymentMethod
<<Abstract>

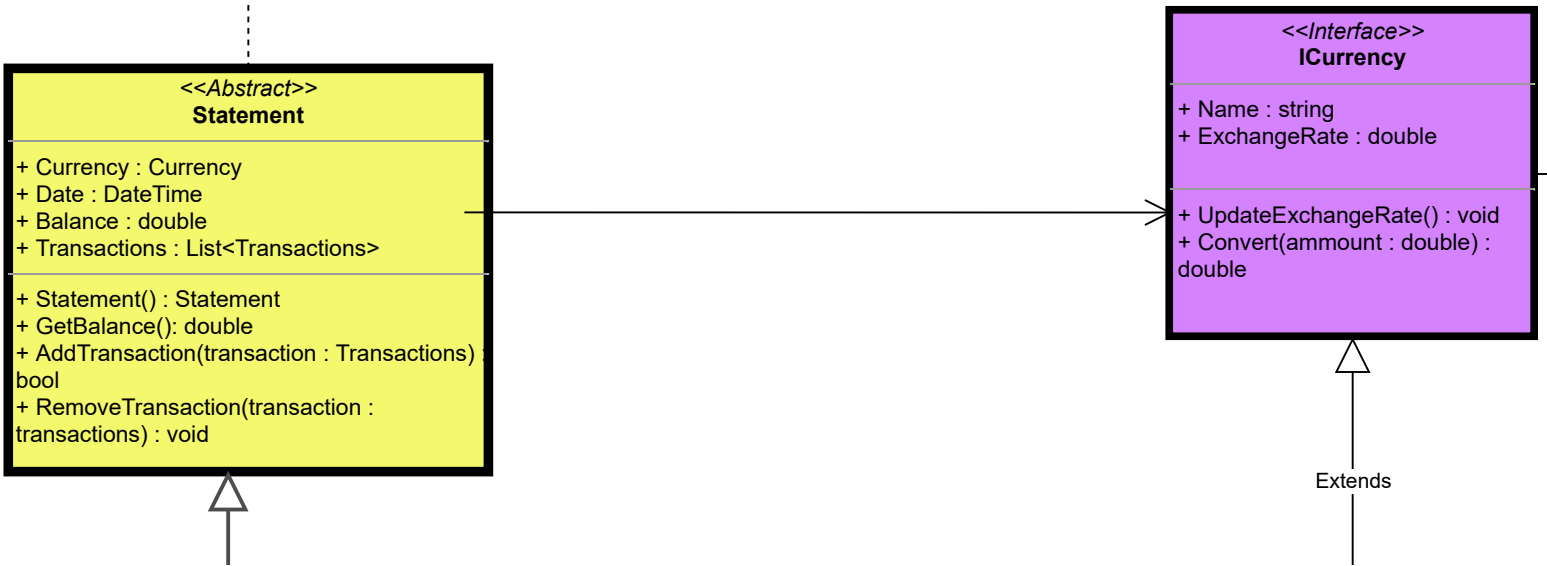

<<Interface>>
IExitFormat
+ EntryFormat : EntryFormat
Use
IntConsoleReader
+ GetInput (message : string) : int
TelegramAPI
+ EntryFormat : IEntryFormat
UserProfile
ewAlert : Alert) : void
ert(newAlert : Alert) : void
ntMethod(newMethod :
hod) : void
aymentMethod(newMethod :
hod) : void
<<Interface>
IIntInput
+ GetInput (message :
IntConsoleRe
+ GetInput (messa
string) : int
<<Abstract>>
Alert
+ Name : string
+ Level : Double
+ IsOn: bool
+ Message : string
+ Alert(name : string, level : double, message : string) : Alert
+ ChangeLevel(newvalue : double) : void
+ TurnOnAlert() : void
+ TurnOffAlert() : void
+ TrackLevel(List<PaymentMethod>) : void
getAlert
Alert(name
ouble,
) :
ert
HighSpendingAlert
+ HighSpendingAlert(name : string, level : double, message : string) : HighSpendingAlert
LowFundsAlert
+ LowFundsAlert(name : string, level : double, message : string) : LowFundsAlert

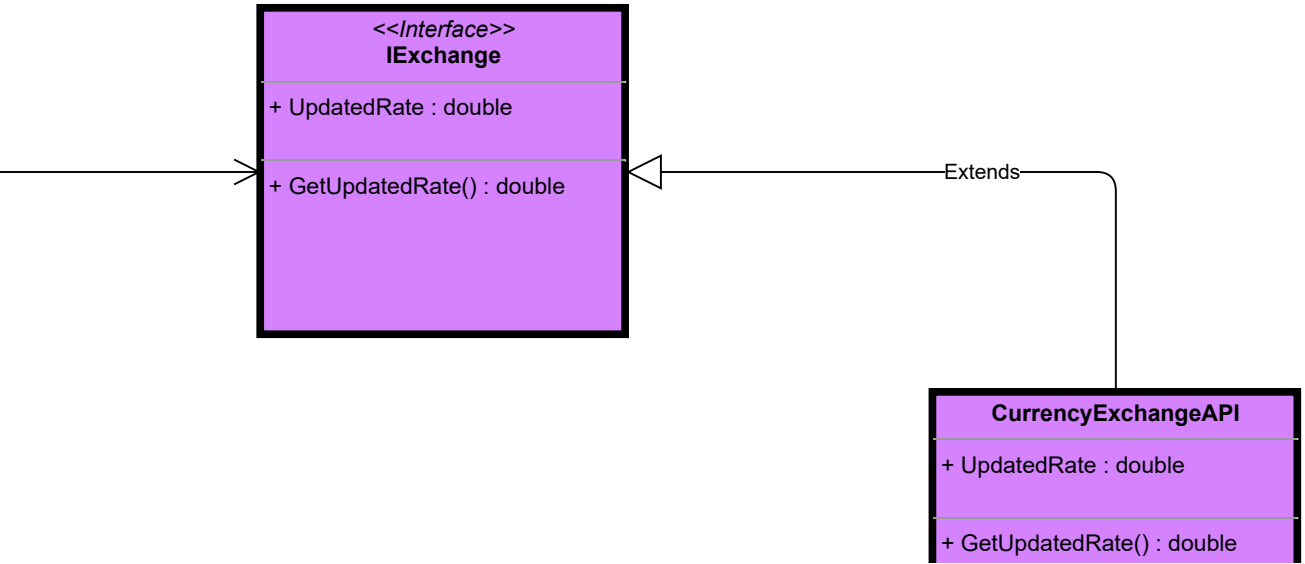

<<Interface>>
IEntryFormat
string) : int
<<Interface>>
IStringtInput
+ GetInput (message : string) :
String
ExcelAPI
ader
ge :
ConsoleReader
+ GetInput (message :
string) : string

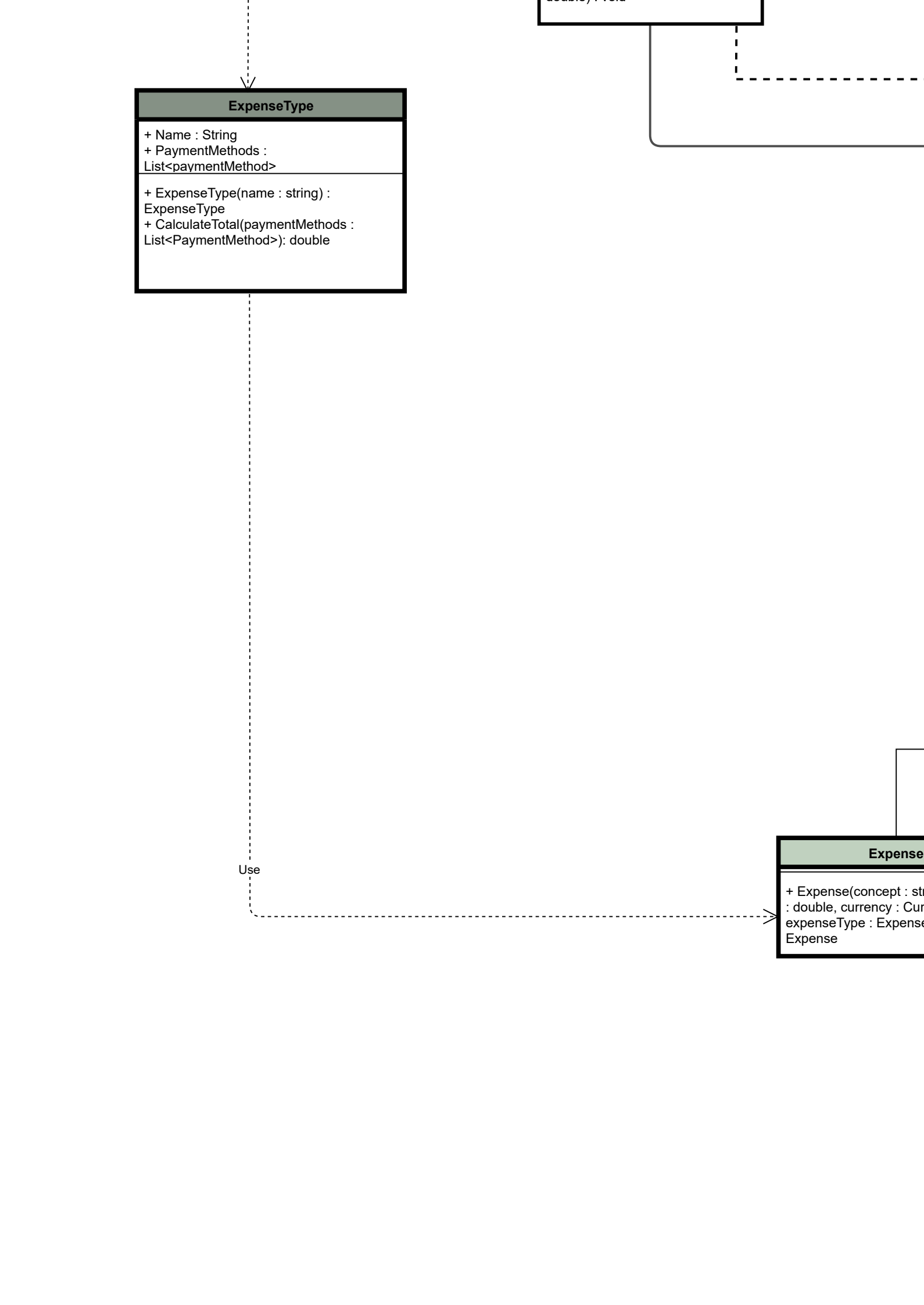

Use
CreditCard
+ Limit : double
+ CurrentStatement : CardStatement
+ StatementList: List<CardStatement>
+ CreditCard (currency : Currency, limit : double) : CreditCard
+ GetBalance() : double
+ SetNewLimit (NewLimit : double) : void
+ NewMonth(NewStatement : CreditCardStatement) : void
CardStatement
+ Limit: double
+ CardStatement(currency . Currency, date : DateTime, limit : double, lastbalance : double)
+ AddTransaction(transaction : Transactions) : bool
+ RemoveTransaction(transaction : transactions) : void
+ GetBalance(): double
+ CalculatePaymentAmmount() : double
+ MakePayment(ammount :

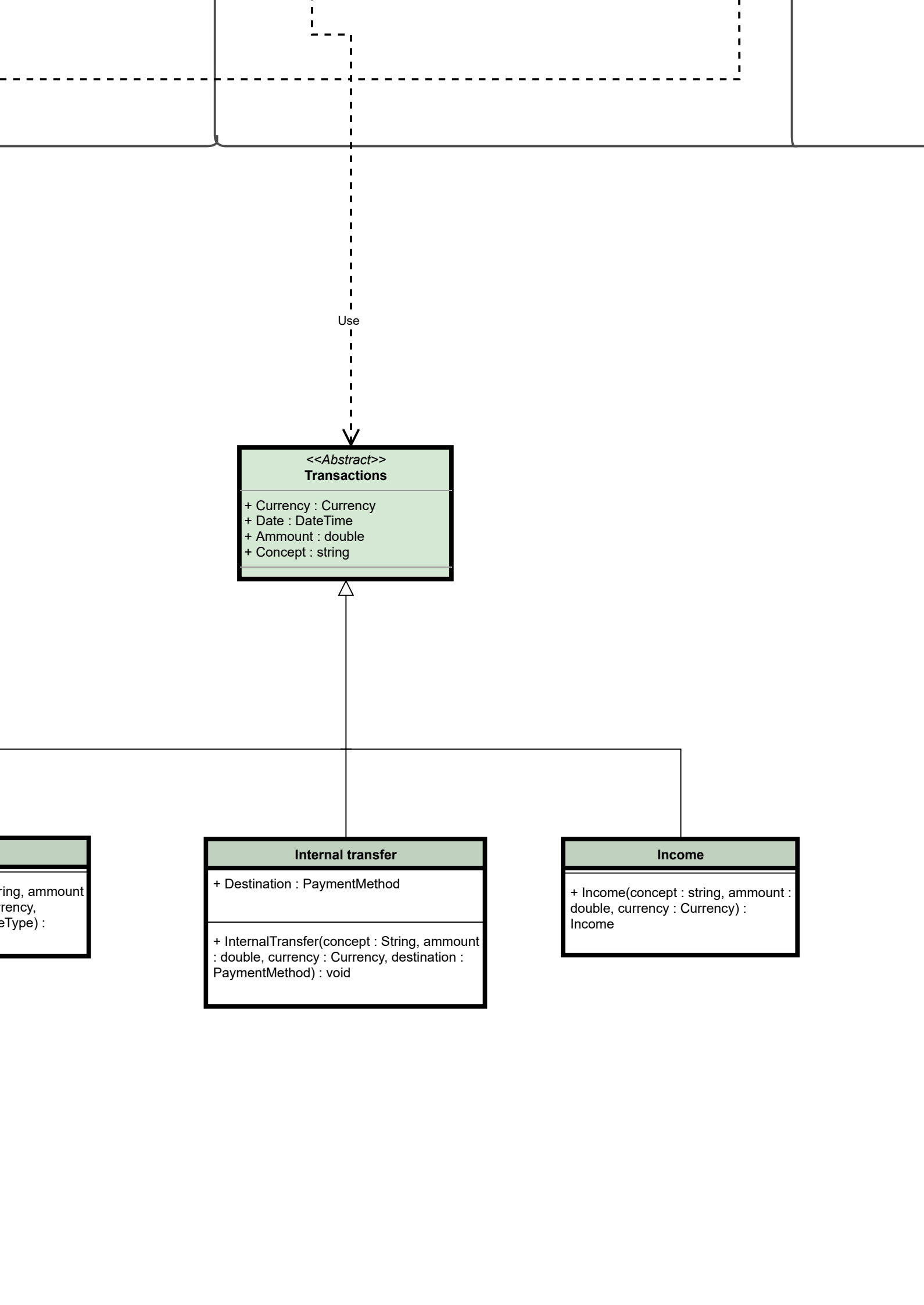

PaymentMethod
+ Currency : Currency
# Balance : Double
+ CurrentStatement : Statement
+GetBalance() : double
Extends
BankAccount
+ CurrentStatement : BankAccountStatement
+ StatementList : List<BankAccountStatement>
+ Bankname : string
+ BankAccount (bankname : string, currency : Currency, date : DateTime) : BankAccount
+ GetBalance() : double
+ NewMonth(NewStatement : BankAccountStatement) : void
Wallet
+ SubWalletList : List<SubWallet>
+ CurrencyList :List<Currency>
+ Wallet (subwallet : SubWallet) : Wallet
+ AddSubWallet (SubWallet : newSubWallet) : void
+ RemoveSubWallet (subWallet : SubWallet) : void
+ GetBalanceBySubWallet (Currency : Currency) :
+ GetBalance() : double
SubWallet
+ Name : string
+ Currency : Currency
+ Statement : WalletStatement
+ SubWallet (Name : String, currency : Currency) : SubWallet
BankAccountStatement
+ Transactions : List<Transactions>
+ BankAccountStatement(currency . Currency, date : DateTime, lastbalance : double)
+ AddTransaction(transaction : Transactions) : bool
+ RemoveTransaction(transaction : transactions) : void
+ GetBalance(): double
WalletStatement
+ Transactions : List<Transactions>
+ WalletStatement (currency : Currency) : WalletStatement
+ GetBalance(): double
+ AddTransaction(transaction : Transactions) : bool
+ RemoveTransaction(transaction : transactions) : void

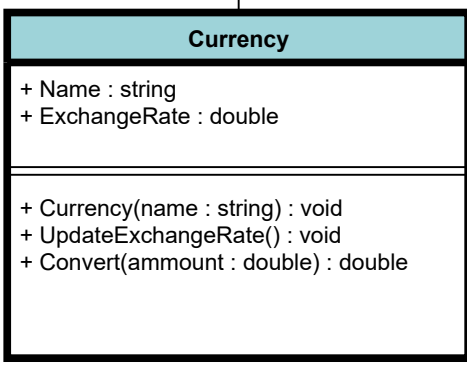
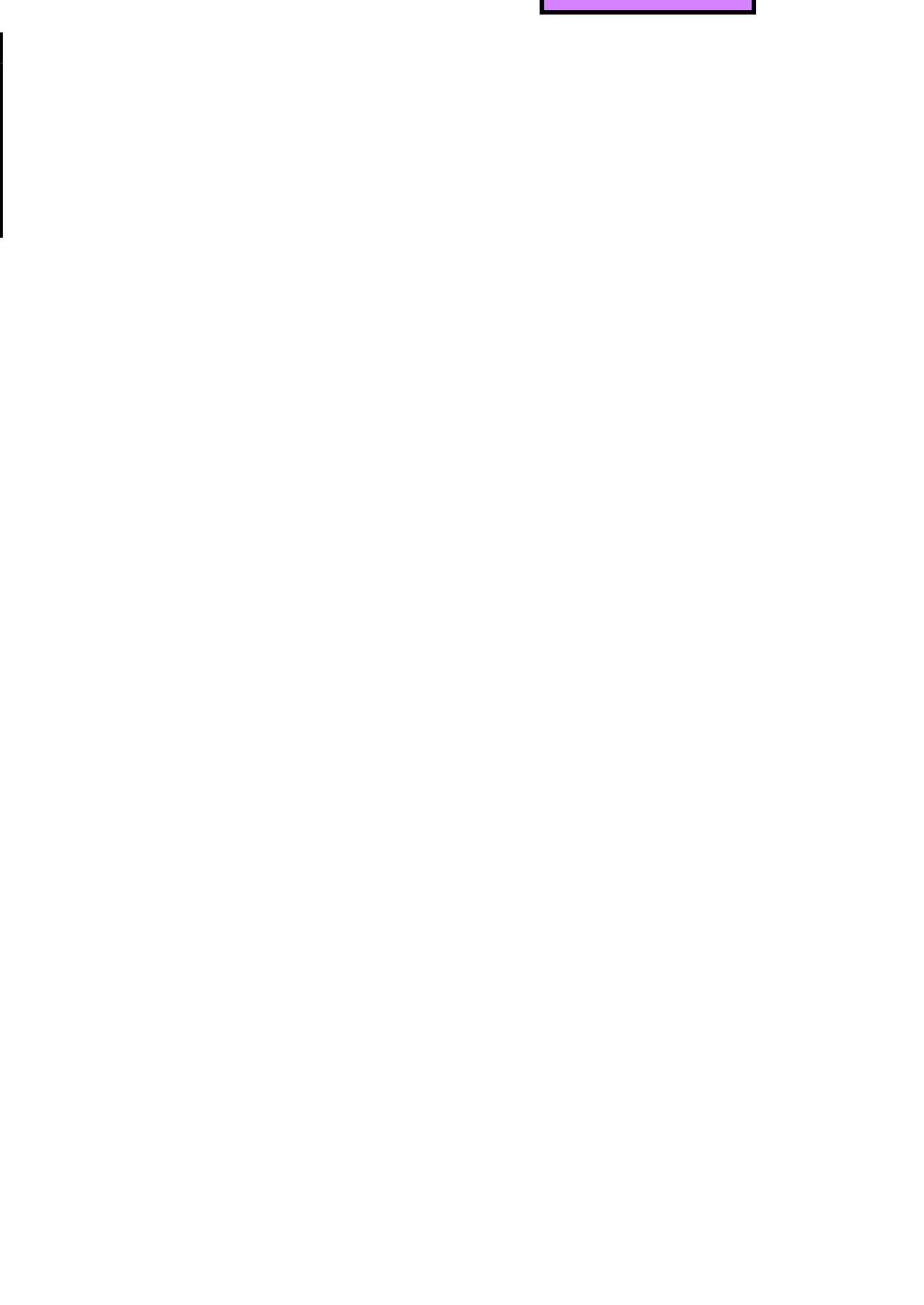
lList


<<Abstract>>
Statement
+ Currency : Currency
+ Date : DateTime
+ Balance : double
+ Transactions : List<Transactions>
+ Statement() : Statement
+ GetBalance(): double
+ AddTransaction(transaction : Transactions) : bool
+ RemoveTransaction(transaction : transactions) : void
<<Interface>>
ICurrency
+ Name : string
+ ExchangeRate : double
+ UpdateExchangeRate() : void
+ Convert(ammount : double) : double
Extends


<<Interface>>
IExchange
+ UpdatedRate : double
+ GetUpdatedRate() : double
Extends
CurrencyExchangeAPI
+ UpdatedRate : double
+ GetUpdatedRate() : double


ExpenseType
+ Name : String
+ PaymentMethods :
List<paymentMethod>
+ ExpenseType(name : string) :
ExpenseType
+ CalculateTotal(paymentMethods :
List<PaymentMethod>): double
Use
Expense
+ Expense(concept : str
: double, currency : Cur
expenseType : Expense
Expense


Use
<<Abstract>>
Transactions
+ Currency : Currency
+ Date : DateTime
+ Ammount : double
+ Concept : string
ring, ammount
rrency,
eType) :
Internal transfer
+ Destination : PaymentMethod
+ InternalTransfer(concept : String, ammount : double, currency : Currency, destination : PaymentMethod) : void
Income
+ Income(concept : string, ammount : double, currency : Currency) : Income


Currency
+ Name : string
+ ExchangeRate : double
+ Currency(name : string) : void
+ UpdateExchangeRate() : void
+ Convert(ammount : double) : double