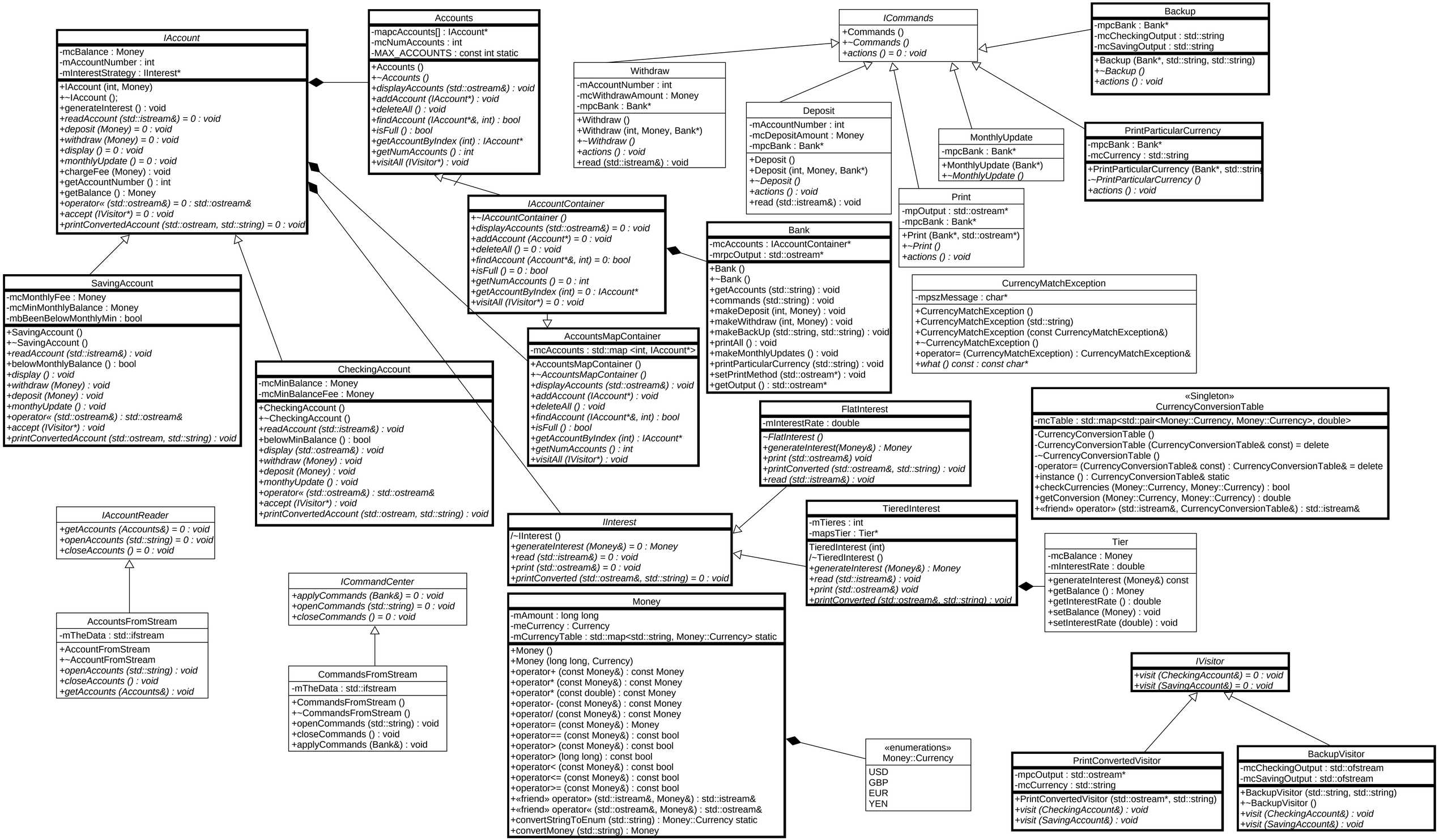
IAccount
-mcBalance : Money
-mAccountNumber : int
-mInterestStrategy : IInterest*
+IAccount (int, Money)
+~IAccount ();
+generateInterest () : void
+readAccount (std::istream&) = 0 : void
+deposit (Money) = 0 : void
+withdraw (Money) = 0 : void
+display () = 0 : void
+monthlyUpdate () = 0 : void
+chargeFee (Money) : void
+getAccountNumber () : int
+getBalance () : Money
+operator« (std::ostream&) = 0 : std::ostream&
+accept (IVisitor*) = 0 : void
+printConvertedAccount (std::ostream, std::string) = 0 : void
Accounts
-mapcAccounts[] : IAccount*
-mcNumAccounts : int
-MAX_ACCOUNTS : const int static
+Accounts ()
+~Accounts ()
+displayAccounts (std::ostream&) : void
+addAccount (IAccount*) : void
+deleteAll () : void
+findAccount (IAccount*&, int) : bool
+isFull () : bool
+getAccountByIndex (int) : IAccount*
+getNumAccounts () : int
+visitAll (IVisitor*) : void
ICommands
+Commands ()
+~Commands ()
+actions () = 0 : void
Backup
-mpcBank : Bank*
-mcCheckingOutput : std::string
-mcSavingOutput : std::string
+Backup (Bank*, std::string, std::string)
+~Backup ()
+actions () : void
Withdraw
-mAccountNumber : int
-mcWithdrawAmount : Money
-mpcBank : Bank*
+Withdraw ()
+Withdraw (int, Money, Bank*)
+~Withdraw ()
+actions () : void
+read (std::istream&) : void
Deposit
-mAccountNumber : int
-mcDepositAmount : Money
-mpcBank : Bank*
+Deposit ()
+Deposit (int, Money, Bank*)
+~Deposit ()
+actions () : void
+read (std::istream&) : void
MonthlyUpdate
-mpcBank : Bank*
+MonthlyUpdate (Bank*)
+~MonthlyUpdate ()
PrintParticularCurrency
-mpcBank : Bank*
-mcCurrency : std::string
+PrintParticularCurrency (Bank*, std::string
-~PrintParticularCurrency ()
+actions () : void
Print
-mpOutput : std::ostream*
-mpcBank : Bank*
+Print (Bank*, std::ostream*)
+~Print ()
+actions () : void
IAccountContainer
+~IAccountContainer ()
+displayAccounts (std::ostream&) = 0 : void
+addAccount (Account*) = 0 : void
+deleteAll () = 0 : void
+findAccount (Account*&, int) = 0: bool
+isFull () = 0 : bool
+getNumAccounts () = 0 : int
+getAccountByIndex (int) = 0 : IAccount*
+visitAll (IVisitor*) = 0 : void
Bank
-mcAccounts : IAccountContainer*
-mrpcOutput : std::ostream*
+Bank ()
+~Bank ()
+getAccounts (std::string) : void
+commands (std::string) : void
+makeDeposit (int, Money) : void
+makeWithdraw (int, Money) : void
+makeBackUp (std::string, std::string) : void
+printAll () : void
+makeMonthlyUpdates () : void
+printParticularCurrency (std::string) : void
+setPrintMethod (std::ostream*) : void
+getOutput () : std::ostream*
CurrencyMatchException
-mpszMessage : char*
+CurrencyMatchException ()
+CurrencyMatchException (std::string)
+CurrencyMatchException (const CurrencyMatchException&)
+~CurrencyMatchException ()
+operator= (CurrencyMatchException) : CurrencyMatchException&
+what () const : const char*
SavingAccount
-mcMonthlyFee : Money
-mcMinMonthlyBalance : Money
-mbBeenBelowMonthlyMin : bool
+SavingAccount ()
+~SavingAccount ()
+readAccount (std::istream&) : void
+belowMonthlyBalance () : bool
+display () : void
+withdraw (Money) : void
+deposit (Money) : void
+monthyUpdate () : void
+operator« (std::ostream&) : std::ostream&
+accept (IVisitor*) : void
+printConvertedAccount (std::ostream, std::string) : void
AccountsMapContainer
-mcAccounts : std::map <int, IAccount*>
+AccountsMapContainer ()
+~AccountsMapContainer ()
+displayAccounts (std::ostream&) : void
+addAccount (IAccount*) : void
+deleteAll () : void
+findAccount (IAccount*&, int) : bool
+isFull () : bool
+getAccountByIndex (int) : IAccount*
+getNumAccounts () : int
+visitAll (IVisitor*) : void
CheckingAccount
-mcMinBalance : Money
-mcMinBalanceFee : Money
+CheckingAccount ()
+~CheckingAccount ()
+readAccount (std::istream&) : void
+belowMinBalance () : bool
+display (std::ostream&) : void
+withdraw (Money) : void
+deposit (Money) : void
+monthyUpdate () : void
+operator« (std::ostream&) : std::ostream&
+accept (IVisitor*) : void
+printConvertedAccount (std::ostream, std::string) : void
FlatInterest
-mInterestRate : double
~FlatInterest ()
+generateInterest(Money&) : Money
+print (std::ostream&) void
+printConverted (std::ostream&, std::string) : void
+read (std::istream&) : void
«Singleton»
CurrencyConversionTable
-mcTable : std::map<std::pair<Money::Currency, Money::Currency>, double>
-CurrencyConversionTable ()
-CurrencyConversionTable (CurrencyConversionTable& const) = delete
-~CurrencyConversionTable ()
-operator= (CurrencyConversionTable& const) : CurrencyConversionTable& = delete
+instance () : CurrencyConversionTable& static
+checkCurrencies (Money::Currency, Money::Currency) : bool
+getConversion (Money::Currency, Money::Currency) : double
+«friend» operator» (std::istream&, CurrencyConversionTable&) : std::istream&
IAccountReader
+getAccounts (Accounts&) = 0 : void
+openAccounts (std::string) = 0 : void
+closeAccounts () = 0 : void
IInterest
/~IInterest ()
+generateInterest (Money&) = 0 : Money
+read (std::istream&) = 0 : void
+print (std::ostream&) = 0 : void
+printConverted (std::ostream&, std::string) = 0 : void
TieredInterest
-mTieres : int
-mapsTier : Tier*
TieredInterest (int)
/~TieredInterest ()
+generateInterest (Money&) : Money
+read (std::istream&) : void
+print (std::ostream&) void
+printConverted (std::ostream&, std::string) : void
Tier
-mcBalance : Money
-mInterestRate : double
+generateInterest (Money&) const
+getBalance () : Money
+getInterestRate () : double
+setBalance (Money) : void
+setInterestRate (double) : void
ICommandCenter
+applyCommands (Bank&) = 0 : void
+openCommands (std::string) = 0 : void
+closeCommands () = 0 : void
AccountsFromStream
-mTheData : std::ifstream
+AccountFromStream
+~AccountFromStream
+openAccounts (std::string) : void
+closeAccounts () : void
+getAccounts (Accounts&) : void
CommandsFromStream
-mTheData : std::ifstream
+CommandsFromStream ()
+~CommandsFromStream ()
+openCommands (std::string) : void
+closeCommands () : void
+applyCommands (Bank&) : void
Money
-mAmount : long long
-meCurrency : Currency
-mCurrencyTable : std::map<std::string, Money::Currency> static
+Money ()
+Money (long long, Currency)
+operator+ (const Money&) : const Money
+operator* (const Money&) : const Money
+operator* (const double) : const Money
+operator- (const Money&) : const Money
+operator/ (const Money&) : const Money
+operator= (const Money&) : Money
+operator== (const Money&) : const bool
+operator> (const Money&) : const bool
+operator> (long long) : const bool
+operator< (const Money&) : const bool
+operator<= (const Money&) : const bool
+operator>= (const Money&) : const bool
+«friend» operator» (std::istream&, Money&) : std::istream&
+«friend» operator« (std::ostream&, Money&) : std::ostream&
+convertStringToEnum (std::string) : Money::Currency static
+convertMoney (std::string) : Money
«enumerations»
Money::Currency
USD
GBP
EUR
YEN
IVisitor
+visit (CheckingAccount&) = 0 : void
+visit (SavingAccount&) = 0 : void
PrintConvertedVisitor
-mpcOutput : std::ostream*
-mcCurrency : std::string
+PrintConvertedVisitor (std::ostream*, std::string)
+visit (CheckingAccount&) : void
+visit (SavingAccount&) : void
BackupVisitor
-mcCheckingOutput : std::ofstream
-mcSavingOutput : std::ofstream
+BackupVisitor (std::string, std::string)
+~BackupVisitor ()
+visit (CheckingAccount&) : void
+visit (SavingAccount&) : void