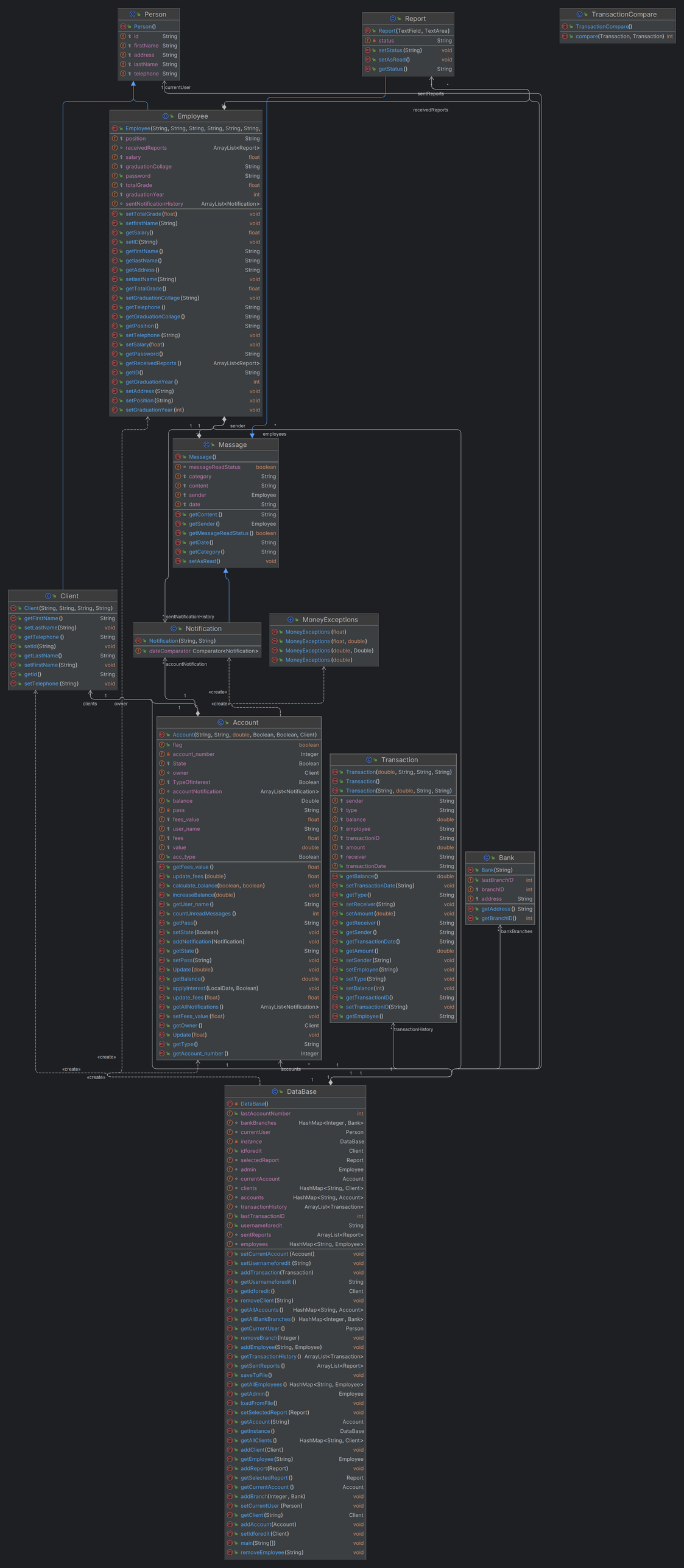
Person
Person()
id String
firstName String
address String
lastName String
telephone String
Report
Report(TextField, TextArea)
status String
setStatus(String) void
setAsRead() void
getStatus() String
TransactionCompare
TransactionCompare()
compare(Transaction, Transaction) int
currentUser
sentReports
receivedReports
Employee
Employee(String, String, String, String, String, String,
position String
receivedReports ArrayList<Report>
salary float
graduationCollage String
password String
totalGrade float
graduationYear int
sentNotificationHistory ArrayList<Notification>
setTotalGrade(float) void
setfirstName(String) void
getSalary() float
setID(String) void
getfirstName() String
getlastName() String
getAddress() String
setlastName(String) void
getTotalGrade() float
setGraduationCollage(String) void
getTelephone() String
getGraduationCollage() String
getPosition() String
setTelephone(String) void
setSalary(float) void
getPassword() String
getReceivedReports() ArrayList<Report>
getID() String
getGraduationYear() int
setAddress(String) void
setPosition(String) void
setGraduationYear(int) void
sender
employees
Message
Message()
messageReadStatus boolean
category String
content String
sender Employee
date String
getContent() String
getSender() Employee
getMessageReadStatus() boolean
getDate() String
getCategory() String
setAsRead() void
Client
Client(String, String, String, String)
getFirstName() String
setLastName(String) void
getTelephone() String
setId(String) void
getLastName() String
setFirstName(String) void
getId() String
setTelephone(String) void
sentNotificationHistory
Notification
Notification(String, String)
dateComparator Comparator<Notification>
MoneyExceptions
MoneyExceptions(float)
MoneyExceptions(float, double)
MoneyExceptions(double, Double)
MoneyExceptions(double)
accountNotification
«create»
«create»
clients
owner
Account
Account(String, String, double, Boolean, Boolean, Client)
flag boolean
account_number Integer
State Boolean
owner Client
TypeOfInterest Boolean
accountNotification ArrayList<Notification>
balance Double
pass String
fees_value float
user_name String
fees float
value double
acc_type Boolean
getFees_value() float
update_fees(double) float
calculate_balance(boolean, boolean) void
increaseBalance(double) void
getUser_name() String
countUnreadMessages() int
getPass() String
setState(Boolean) void
addNotification(Notification) void
getState() String
setPass(String) void
Update(double) void
getBalance() double
applyInterest(LocalDate, Boolean) void
update_fees(float) float
getAllNotifications() ArrayList<Notification>
setFees_value(float) void
getOwner() Client
Update(float) void
getType() String
getAccount_number() Integer
Transaction
Transaction(double, String, String, String)
Transaction()
Transaction(String, double, String, String)
sender String
type String
balance double
employee String
transactionID String
amount double
receiver String
transactionDate String
getBalance() double
setTransactionDate(String) void
getType() String
setReceiver(String) void
setAmount(double) void
getReceiver() String
getSender() String
getTransactionDate() String
getAmount() double
setSender(String) void
setEmployee(String) void
setType(String) void
setBalance(int) void
getTransactionID() String
setTransactionID(String) void
getEmployee() String
Bank
Bank(String)
lastBranchID int
branchID int
address String
getAddress() String
getBranchID() int
bankBranches
transactionHistory
accounts
«create»
«create»
«create»
DataBase
DataBase()
lastAccountNumber int
bankBranches HashMap<Integer, Bank>
currentUser Person
instance DataBase
idforedit Client
selectedReport Report
admin Employee
currentAccount Account
clients HashMap<String, Client>
accounts HashMap<String, Account>
transactionHistory ArrayList<Transaction>
lastTransactionID int
usernameforedit String
sentReports ArrayList<Report>
employees HashMap<String, Employee>
setCurrentAccount(Account) void
setUsernameforedit(String) void
addTransaction(Transaction) void
getUsernameforedit() String
getIdforedit() Client
removeClient(String) void
getAllAccounts() HashMap<String, Account>
getAllBankBranches() HashMap<Integer, Bank>
getCurrentUser() Person
removeBranch(Integer) void
addEmployee(String, Employee) void
getTransactionHistory() ArrayList<Transaction>
getSentReports() ArrayList<Report>
saveToFile() void
getAllEmployees() HashMap<String, Employee>
getAdmin() Employee
loadFromFile() void
setSelectedReport(Report) void
getAccount(String) Account
getInstance() DataBase
getAllClients() HashMap<String, Client>
addClient(Client) void
getEmployee(String) Employee
addReport(Report) void
getSelectedReport() Report
getCurrentAccount() Account
addBranch(Integer, Bank) void
setCurrentUser(Person) void
getClient(String) Client
addAccount(Account) void
setIdforedit(Client) void
main(String[]) void
removeEmployee(String) void