/\*

\* Bank.cpp

\*/

#include "Bank.h"

#include <algorithm>

#include <string>

Bank::Bank() {}

void Bank::addAccount(Account \*a) {

accounts.push\_back(a);

}

void Bank::addBankOfficer(BankOfficer b){

bankOfficers.push\_back(b);

}

vector<BankOfficer> Bank::getBankOfficers() const {

return bankOfficers;

}

vector<Account \*> Bank::getAccounts() const {

return accounts;

}

// ----------------------------------------------------------------------------------------------

// a alterar

double Bank::getWithdraw(string cod1) const {

double result = 0;

for(auto &x: accounts) {

if (x->getCodH() == cod1)

result += x->getWithdraw();

}

return result;

}

// a alterar

vector<Account \*> Bank::removeBankOfficer(string name){

vector<Account \*> res;

for (vector<BankOfficer>::iterator it = bankOfficers.begin(); it != bankOfficers.end(); it++) {

if ((\*it).getName() == name) {

res = (\*it).getAccounts();

bankOfficers.erase(it);

break;

}

}

return res;

}

// a alterar

const BankOfficer & Bank::addAccountToBankOfficer(Account \*ac, string name) {

for (vector<BankOfficer>::iterator it = bankOfficers.begin(); it != bankOfficers.end(); it++) {

if ((\*it).getName() == name) {

(\*it).addAccount(ac);

return (\*it);

}

}

throw NoBankOfficerException(name);

}

bool compareAccounts(Account \*a1, Account \*a2) {

if (a1->getBalance() == a2->getBalance())

return a1->getCodIBAN() < a2->getCodIBAN();

return a1->getBalance() < a2->getBalance();

}

// a alterar

void Bank::sortAccounts() {

sort(accounts.begin(), accounts.end(), compareAccounts);

}