

---

## Lehrveranstaltung "Objektorientierte Programmierung" Übungsblatt 5

### Hinweise:

Dieses Übungsblatt ist zur Zulassung zu der Klausur erfolgreich zu bearbeiten ("Erfolgreich" bedeutet: Keine Programmabstürze bzw. Endlosschleifen, Aufgabenstellung einschließlich der Nebenbedingungen müssen eingehalten sowie Kommentierung und Einrückung korrekt sein! Compilerwarnungen sollen möglichst vermieden werden.).

Die Aufgaben werden überwiegend in den Übungszeiten bearbeitet und dort auch abgegeben. Allerdings genügt die Zeit hierfür unter Umständen nicht, so dass Sie auch außerhalb dieser Zeiten die Aufgaben bearbeiten müssen. Der Abgabetermin für diese Aufgabe ist der **15. Dezember 2023**.

---

**Aufgabe:** In der fünften Übungsaufgabe des Projektes „Banken“ müssen einige Klassen angepasst werden und es kommen wieder neue Klassen hinzu (alle neuen Klassen erhalten wie immer – sofern nötig – Getter und Setter sowie eine `print`-Methode):

Die Klasse `CTime` soll analog zur Klasse `CDate` um eine `load`-Methode erweitert werden.

Neu ist die Klasse `CTimestamp`; sie erbt von den Klassen `CDate` und `CTime` und hat zusätzlich noch die private Eigenschaft `Wochentag` (`Weekday`) vom Typ `CWeekday` (ein Aufzählungstyp, der auch noch an geeigneter Stelle definiert werden muss). Im Konstruktor muss der `Wochentag` des Datums ermittelt werden (Stichwort `Wochentagsberechnung`).

Ferner kommt die Klasse `CTransaction` als Buchung hinzu: Sie erhält als private Eigenschaften eine `ID` (`long int`), je einen Zeiger auf Konto und Gegenkonto (Zeiger auf `CAccount`), den Betrag und eine Bearbeitungsgebühr (beides Objekte der Klasse `CMoney`) sowie den Verwendungszweck (`string`).

Außerdem wird die Klasse `CTransactionBatch` als Buchungstapel benötigt: Sie erhält einen Zeitstempel (Objekt der Klasse `CTimestamp`; Zeitpunkt des Einlesens der Daten) sowie einen Vektor von Zeigern auf Objekte der Klasse `CTransaction`. Dem Konstruktor dieser Klasse wird ein Dateiname als Parameter mitgegeben; diese angegebene Datei soll vom Konstruktor eingelesen werden und auf die Konten angewendet werden.

Die Klasse `CBankManager` muss dann noch um einen Vektor von Zeigern auf `CTransactionBatch` erweitert werden. Das Hinzufügen eines `TransactionBatch`-Objektes erfolgt über den „+=“-Operator. Die Ausgabe

der Transaktionen soll mittels der Methode `printTransactions` erfolgen (siehe Beispiel-Ausgabe).

Zum Testen der Klassen soll das vorgegebene Hauptprogramm verwendet werden, das Objekte der einzelnen Klassen erzeugt, auf verschiedene Werte setzt und wieder auf dem Bildschirm ausgibt. Die Ausgabe kann je nachdem, wie die einzelnen Probleme gelöst werden, etwas variieren.

### vorgegebenes Hauptprogramm:

```
#include <iostream>
#include <vector>

using namespace std;

#include "ctimestamp.h"
#include "ctransactionbatch.h"
#include "cbankmanager.h"

void printNow()
{
    CTimestamp Now;

    cout << "Aktueller Zeitstempel: ";
    Now.print();
    cout << endl << endl;
}

int main()
{
    printNow();

    CBankManager BankManager("data.xml");

    cout << endl;
    BankManager.printTransactions(); // sollte "keine Buchungstapel vorhanden" ausgeben
    cout << endl;

    BankManager += new CTransactionBatch("transactions.xml", BankManager);

    cout << endl;
    BankManager.printTransactions(); // gibt alle Stapel mit allen Transaktionen aus
    cout << endl;

    cout << endl;
    BankManager.printBankList(); // Uebersicht der Konten mit geaenderten Kontostaenden
    cout << endl;

    printNow();

    return 0;
}
```

**Beispiel-Ausgabe:**

Aktueller Zeitstempel: So, 13.08.2023; 22:32:23

Datei 'data.xml' wurde erfolgreich geöffnet und wird jetzt eingelesen ...  
Datei wurde eingelesen.

keine Buchungstapel vorhanden

Datei 'transactions.xml' wurde erfolgreich geöffnet und wird jetzt eingelesen ...  
Datei wurde eingelesen.

Buchung ueber 50.00 EUR  
--- von ---  
Kunde : Egon Muster (Kd-Nr. 4711)  
IBAN / BIC: DE99876543210987654321 / DEHOBADExXX  
Kontostand: 150.00 EUR  
--- an ---  
Kunde : Anton Muster (Kd-Nr. 815)  
IBAN / BIC: DE11223344556677889900 / BESPKADEXXX  
Kontostand: 150.00 EUR  
--- Verwendungszweck: ReNr. 123 ---

Buchung ueber 80.00 EUR  
--- von ---  
Kunde : Anton Muster (Kd-Nr. 815)  
IBAN / BIC: DE11223344556677889900 / BESPKADEXXX  
Kontostand: 200.00 EUR  
--- an ---  
Kunde : Paul Muster (Kd-Nr. 1234)  
IBAN / BIC: DE99887766554433221100 / DEHOBADExXX  
Kontostand: 150.00 EUR  
--- Verwendungszweck: Miete Dezember 2019 ---

Berliner Spasskasse

BIC BESPKADEXXX

Anzahl Konten: 2

Kontenliste:

IBAN	Kundenname	Anz.Buchungen	Kontostand
DE99 1234 5678 1234 5678 90	Egon Muster	0	150.00 EUR
DE11 2233 4455 6677 8899 00	Anton Muster	0	120.00 EUR

Deutsche Hochschul-Bank

BIC DEHOBADExXX

Anzahl Konten: 2

Kontenliste:

IBAN	Kundenname	Anz.Buchungen	Kontostand
DE99 8765 4321 0987 6543 21	Egon Muster	0	100.00 EUR
DE99 8877 6655 4433 2211 00	Paul Muster	0	230.00 EUR

Aktueller Zeitstempel: So, 13.08.2023; 22:32:23

Kunde Egon Muster (Kundennr. 4711) wird vernichtet.  
Kunde Anton Muster (Kundennr. 815) wird vernichtet.  
Kunde Paul Muster (Kundennr. 1234) wird vernichtet.  
CAccount: Konto (DE99 1234 5678 1234 5678 90) wird vernichtet!  
CSavingsAccount: Konto (DE11 2233 4455 6677 8899 00) wird vernichtet!  
CAccount: Konto (DE11 2233 4455 6677 8899 00) wird vernichtet!  
Bank Berliner Spasskasse wird vernichtet.  
CCurrentAccount: Konto (DE99 8765 4321 0987 6543 21) wird vernichtet!  
CAccount: Konto (DE99 8765 4321 0987 6543 21) wird vernichtet!  
CFixedDepositAccount: Konto (DE99 8877 6655 4433 2211 00) wird vernichtet!  
CSavingsAccount: Konto (DE99 8877 6655 4433 2211 00) wird vernichtet!  
CCurrentAccount: Konto (DE99 8877 6655 4433 2211 00) wird vernichtet!  
CAccount: Konto (DE99 8877 6655 4433 2211 00) wird vernichtet!  
Bank Deutsche Hochschul-Bank wird vernichtet.