



Lehrveranstaltung für TI-Bachelor: "Vertiefung objektorientierte Programmierung in C++" Übungsblatt 4

Hinweise:

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

Die Aufgaben werden überwiegend in den Übungszeiten bearbeitet. 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 **spätestens** der **29. November 2012**.

Nutzen Sie die Übungen auch, um ggf. Fragen, die sich in den Vorlesungen ergeben haben, anzusprechen.

Aufgabe: In der vierten Übungsaufgabe des Projektes "Banken" sollen folgende Klassen hinzugefügt werden: `TCurrentAccount` und `TSavingsAccount` (beide abgeleitet von der Klasse `TAccount`) sowie `TFixedDepositAccount` (abgeleitet von den beiden Klassen `TCurrentAccount` und `TSavingsAccount`). Ferner sollen die Klassen `TAccount` und `TBank` erweitert werden.

Die Klasse `TCurrentAccount` wird von der Klasse `TAccount` abgeleitet und erhält als zusätzliche Eigenschaft einen Zeiger auf den Geldbetrag vom Dispokredit; es können nach Belieben weitere Eigenschaften hinzugefügt werden. Ferner muss beim Buchen geprüft werden, ob der Dispokredit überschritten wird. Durch die Verwendung eines Zeigers muss im Konstruktor der Speicherplatz für den Geldbetrag reserviert und im Destruktor der Speicherplatz wieder freigegeben werden.

Auch die Klasse `TSavingsAccount` wird von der Klasse `TAccount` abgeleitet; sie erhält als zusätzliche Eigenschaft einen Zeiger auf den Zinssatz in Prozent. Auch hier soll beim Buchen überprüft werden, ob genügend Geld auf dem Sparsbuch ist (negative Kontostände sind nicht zulässig!). Außerdem muss der Zeiger für den Zinssatz wieder im Konstruktor und Destruktor berücksichtigt werden.

Die Klasse `TFixedDepositAccount` soll von den beiden Klassen `TCurrentAccount` und `TSavingsAccount` abgeleitet werden (Mehrfachvererbung). Zusätzliche Eigenschaften sind freiwillig. Alle Eigenschaften der Basisklasse `TAccount` werden nur einmal benötigt. Für alle drei Klassen werden mindestens eine eigene `print` und `printAccountStatement`-Methode benötigt.

Sind die Klassen richtig polymorph programmiert, kann nun im Hauptprogramm in einer Schleife über alle Konten der Bank (Array von Zeigern auf `TAccount`) jeweils der Kontoauszug aufgerufen werden. Dabei muss für jedes Konto automatisch die richtige Kontoauszugs-Methode aufgerufen werden.

Außerdem sollen die Destruktoren der Klasse `TAccount` und deren abgeleiteten Klassen jeweils auf dem Bildschirm melden, von welcher Klasse sie sind und welches Konto gerade vernichtet wird (siehe Beispielausgabe).

In der Klasse `TAccount` soll das Array von Buchungen (`TBooking`) in ein Array von Zeigern auf Buchungen geändert werden. Beim Hinzufügen einer Buchung zu einem Konto soll die Buchung im Array mittels `new` erzeugt werden. Entsprechend müssen beim Vernichten des Kontos alle Buchungen wieder freigegeben werden.

Die Klasse `TBank` soll im Destruktor alle Konten vernichten (, da diese im Hauptprogramm mit `new` erzeugt werden).

Zum Testen der Klassen soll wieder ein kleines Hauptprogramm geschrieben bzw. kann das vorgegebene Beispielpogramm verwendet werden, dass Objekte der Klassen erzeugt, auf verschiedene Werte setzt und wieder auf dem Bildschirm ausgibt.

Beispiel für ein Hauptprogramm:

```
#include <iostream>

using namespace std;

#include "tdate.h"
#include "tmoney.h"
#include "tcustomer.h"
#include "tbank.h"
#include "taccount.h"
#include "tbooking.h"

int main()
{
    TDate Datum1(7, 7, 1977);
    TDate Datum2(8, 8, 1988);

    TCustomer Bundesdruckerei("Geldquelle", TDate(), "", "", "", "");
    TCustomer Kunde1("Egon Muster", Datum1, "Mustergasse", "3a", "99889", "Musterstadt");
    TCustomer Kunde2("Rudolph Rednose", Datum2, "Berliner Str.", "17", "10999", "Berlin");

    TBank *Bank1 = new TBank("Berliner Bank", "10020000");

    TAccount *Geldquelle = new TAccount(&Bundesdruckerei, Bank1, "0", "0000");
    TCurrentAccount *Konto1 = new TCurrentAccount(&Kunde1, Bank1, "1234567890", "9999", TMoney(100.0));
    TSavingsAccount *Konto2 = new TSavingsAccount(&Kunde2, Bank1, "9876543120", "0101", 1.5);
    TFixedDepositAccount *Konto3 = new TFixedDepositAccount(&Kunde1, Bank1, "111333555", "1357", TMoney(100.0), 1.5);

    TBooking Buchung1(TMoney(150.0), Konto1, Geldquelle, TDate(), TTime(), string("Startguthaben"));
    TBooking Buchung2(TMoney(50.0), Konto3, Konto1, TDate(), TTime(), string("Umbuchung"));
    TBooking Buchung3(TMoney(39.9), Konto2, Konto1, TDate(), TTime(), string("Rechnung 4711"));

    // Ausgaben:
    cout << "Kunde 1:" << endl; Kunde1.print(); cout << endl;
    cout << "Kunde 2:" << endl; Kunde2.print(); cout << endl;
    cout << "Bank 1: " << endl; Bank1->print(); cout << endl;

    for (int i = 0; i < Bank1->getAccountCounter(); i++)
    {
        (Bank1->getAccount(i))->printAccountStatement();
        cout << endl;
    }

    cout << "Jetzt wird die Bank vernichtet und damit auch die Konten der Bank:" << endl;
    delete Bank1;

    return 0;
}
```

Beispiel-Ausgabe:

Kunde 1:

Egon Muster

Mustergasse 3a

99889 Musterstadt

geboren am: 07.07.1977

Konten:

- Kontonr.: 1234567890 (Kontostand: 60.10 EUR)

- Kontonr.: 111333555 (Kontostand: 50.00 EUR)

Kunde 2:

Rudolph Rednose

Berliner Str. 17

10999 Berlin

geboren am: 08.08.1988

Konten:

- Kontonr.: 9876543120 (Kontostand: 39.90 EUR)

Bank 1:

Berliner Bank

BLZ 10020000

Anzahl Konten: 4

Kontenliste:

Kontonr.	Kundenname	Anz. Buchungen	Kontostand
0	Geldquelle	1	- 150.00 EUR
1234567890	Egon Muster	3	60.10 EUR
9876543120	Rudolph Rednose	1	39.90 EUR
111333555	Egon Muster	1	50.00 EUR

Kontoauszug vom 14.11.2012; 19:53:06

Kontonr.: 0; BLZ 10020000

Kontoinhaber: Geldquelle

Datum	Betrag	Absender / Empfaenger	Buchungstext
14.11.2012	- 150.00 EUR	Egon Muster	Startguthaben

aktueller Kontostand: -150.00 EUR

Kontoauszug vom 14.11.2012; 19:53:06

Kontonr.: 1234567890; BLZ 10020000

Kontoinhaber: Egon Muster

Datum	Betrag	Absender / Empfaenger	Buchungstext
14.11.2012	150.00 EUR	Geldquelle	Startguthaben
14.11.2012	- 50.00 EUR	Egon Muster	Umbuchung
14.11.2012	- 39.90 EUR	Rudolph Rednose	Rechnung 4711

aktueller Kontostand: 60.10 EUR

max. Dispo: 100.00 EUR

Kontoauszug vom 14.11.2012; 19:53:06

Kontonr.: 9876543120; BLZ 10020000

Kontoinhaber: Rudolph Rednose

Datum	Betrag	Absender / Empfaenger	Buchungstext
14.11.2012	39.90 EUR	Egon Muster	Rechnung 4711

aktueller Kontostand: 39.90 EUR

Zinssatz: 1.50 %

Kontoauszug vom 14.11.2012; 19:53:06

Kontonr.: 111333555; BLZ 10020000

Kontoinhaber: Egon Muster

Datum	Betrag	Absender / Empfaenger	Buchungstext
14.11.2012	50.00 EUR	Egon Muster	Umbuchung

aktueller Kontostand: 50.00 EUR

max. Dispo: 100.00 EUR

Zinssatz: 1.50 %

Jetzt wird die Bank vernichtet und damit auch die Konten der Bank:

TAccount: Konto (KtoNr. 0) wird vernichtet!
 TCurrentAccount: Girokonto (KtoNr. 1234567890) wird vernichtet!
 TAccount: Konto (KtoNr. 1234567890) wird vernichtet!
 TSavingsAccount: Sparbuchkonto (KtoNr. 9876543120) wird vernichtet!
 TAccount: Konto (KtoNr. 9876543120) wird vernichtet!
 TFixedDepositAccount: Tagesgeldkonto (KtoNr. 111333555) wird vernichtet!
 TSavingsAccount: Sparbuchkonto (KtoNr. 111333555) wird vernichtet!
 TCurrentAccount: Girokonto (KtoNr. 111333555) wird vernichtet!
 TAccount: Konto (KtoNr. 111333555) wird vernichtet!