



# Lehrveranstaltung für TI-Bachelor: "Vertiefung objektorientierte Programmierung in C++" Übungsblatt 2

## 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 **01. November 2012**.

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

---

**Aufgabe:** In der zweiten Übungsaufgabe des Projektes „Banken“ sollen die Klassen `TMoney`, `TCustomer` und `TAccount` erstellt werden.

Die Klasse `TMoney` erhält als private Eigenschaften den Betrag (`double`) und die Währung (`string`). Neben den Konstruktoren werden hier erst einmal `set`- und `get`-Methoden sowie eine `print`-Methode (Ausgabe entsprechend der Beispielausgabe) benötigt.

Die Klasse `TCustomer` erhält als private Eigenschaften folgende Daten vom Kunden: Name, Straße, Hausnr., PLZ und Ort (alles `strings`), Geburtstag (Klasse `TDate`), die Anzahl der Konten und ein Array von Zeigern auf Konten (Klasse `TAccount`). Die Größe des Arrays wird mit einer Konstanten (z.B. `#define MAXCACCOUNTS 10`) festgelegt. Auch hier werden erst einmal die `set`- und `get`-Methoden sowie eine `print`-Methode (siehe Beispielausgabe) benötigt.

Die Klasse `TAccount` erhält als private Eigenschaften folgende Daten: einen Zeiger auf den Kunden (Klasse `TCustomer`), die Kontonr. und die PIN als `strings`, die Anzahl der Buchungen (wird erst später benötigt und daher auf 0 gesetzt) sowie den Kontostand (Klasse `TMoney`). Die Buchungsliste und der Zeiger auf die Bank werden erst einmal weggelassen. Auch hier werden die `set`- und `get`-Methoden sowie eine `print`-Methode (siehe Beispielausgabe) benötigt.

Beim Setzen des Kunden muss beim Kunden der Zeiger auf das Konto in der Kontenliste des Kunden gesetzt werden.

Zum Testen der Klassen soll ein kleines Hauptprogramm geschrieben bzw. kann das vorgegebene Beispielpogramm verwendet werden, das einen Geldbetrag, einen Kunden sowie zwei Konten 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 "taccount.h"

int main()
{
    TDate Datum(7, 7, 1977);
    TMoney Betrag1(150.0);
    TCustomer Kundel("Egon Muster", Datum, "Mustergasse", "3a", "99889", "Musterstadt");
    TAccount Konto1(&Kundel, "1234567890", "9999");

    cout << "Betrag 1: " << endl;    Betrag1.print();    cout << endl << endl;
    cout << "Kunde 1:  " << endl;    Kundel.print();    cout << endl << endl;
    cout << "Konto 1;  " << endl;    Konto1.print();    cout << endl << endl;

    TAccount Konto2(&Kundel, "9876543210", "0101");

    cout << "Kunde 1:  " << endl;    Kundel.print();    cout << endl << endl;
    cout << "Konto 2;  " << endl;    Konto2.print();    cout << endl << endl;

    return 0;
}
```

### Beispiel-Ausgabe:

```
Betrag 1:
150.00 EUR

Kunde 1:
Egon Muster
Mustergasse 3a
99889 Musterstadt
geboren am: 07.07.1977
Konten:
- Kontonr.: 1234567890

Konto 1;
Kunde:
Egon Muster
Mustergasse 3a
99889 Musterstadt
geboren am: 07.07.1977
Konten:
- Kontonr.: 1234567890
Kontonr.      : 1234567890
Anz. Buchungen: 0
Kontostand    : 0.00 EUR

Kunde 1:
Egon Muster
Mustergasse 3a
99889 Musterstadt
geboren am: 07.07.1977
Konten:
- Kontonr.: 1234567890
- Kontonr.: 9876543210
```

Konto 2;  
Kunde:  
Egon Muster  
Mustergasse 3a  
99889 Musterstadt  
geboren am: 07.07.1977  
Konten:  
- Kontonr.: 1234567890  
- Kontonr.: 9876543210  
Kontonr. : 9876543210  
Anz. Buchungen: 0  
Kontostand : 0.00 EUR