**Dipl.Phys. Gerald Kempfer**

Berliner Hochschule für Technik
- University of Applied Sciences -
Fachbereich VI – Informatik und Medien
Studiengang Technische Informatik Bachelor

# Lehrveranstaltung "Objektorientierte Programmierung"
# Übungsblatt 3

**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 **17. November 2023**.

---

## Aufgabe:

In der dritten Übungsaufgabe des Projektes „Banken“ soll die Klasse `CBank` erstellt, die Klasse `CAccount` um einen Zeiger auf eine Bank (Klasse `CBank`) erweitert sowie die Klassen `CCurrentAccount` und `CSavingsAccount` von der Klasse `CAccount` abgeleitet werden.

Die Klasse `CBank` erhält als private Eigenschaften einen Namen sowie die BIC (jeweils `string`) sowie einen Vektor von Zeigern auf Konten (Klasse `CAccount`). Als Methoden sollen Konstruktor (kein Standardkonstruktor!), mit dem der Name und die BIC gesetzt werden (die Kontenliste ist anfangs leer), sowie `set`, `get` und `print` implementiert werden. Zusätzlich wird noch eine Methode `addAccount` benötigt, mit der der Bank ein Zeiger auf ein Konto hinzugefügt werden kann. Hier kann einiges von der Klasse `CCustomer` (siehe letzte Aufgabe) abgeguckt werden.

Die Klasse `CAccount` wird um die private Eigenschaft Zeiger auf Bank erweitert. Entsprechend müssen Konstruktur, `get`- und `set`- sowie `print`-Methode angepasst werden. Damit die privaten Eigenschaften vererbt werden können, müssen diese auf `protected` gesetzt werden.

Die Klassen `CCurrentAccount` und `CSavingsAccount` sollen von der Klasse `CAccount` abgeleitet werden. Jede der beiden Klassen erhält eine zusätzliche private Eigenschaft: In der Klasse `CCurrentAccount` kommt der Dispo-Betrag (als Zeiger auf `CMoney`) und in der Klasse `CSavingsAccount` der Zinssatz (`double`) hinzu. Bei beiden Klassen soll jeweils eine `print`-Methode implementiert werden, die die Ausgabe entsprechend der unten stehenden Beispiel-Ausgabe erstellt. Außerdem sollen alle drei Kontenklassen jeweils einen Destruktor erhalten, der die Vernichtung der entsprechenden Konten auf dem Bildschirm verkündet (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.

## vorgegebenes Hauptprogramm:

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

using namespace std;

#include "cdate.h"
#include "ctime.h"
#include "cmoney.h"
#include "caddress.h"
#include "caccount.h"
#include "ccurrentaccount.h"
#include "csavingsaccount.h"
#include "ccustomer.h"
#include "cbank.h"

int main()
{
   CDate Geburtsdatum(7, 7, 1977);
   CMoney Startkapital(150.0);
   CMoney Dispo(250.0);
   double Zinsen = -2.5;
   CAddress Adresse("Mustergasse 3a", "D - 99889", "Musterstadt");
   // Egon, Anton und Paul sind Drillinge, die noch bei Mutti wohnen
   CCustomer Egon(4711, "Egon Muster", Geburtsdatum, Adresse);
   CCustomer Anton(815, "Anton Muster", Geburtsdatum, Adresse);
   CCustomer Paul(1234, "Paul Muster", Geburtsdatum, Adresse);
   CBank Spasskasse("Berliner Spasskasse", "BESPKADEXXX");
   CBank HochschulBank("Deutsche Hochschul-Bank", "DEHOBADEXXX");
   CAccount Konto1(&Spasskasse, "DE99123456781234567890", &Egon,  Startkapital);
   CCurrentAccount Konto2(&HochschulBank, "DE99876543210987654321", &Egon,  Startkapital, &Dispo);
   CSavingsAccount Konto3(&Spasskasse, "DE11223344556677889900", &Anton, Startkapital, Zinsen);
   CAccount Konto4(&HochschulBank, "DE99887766554433221100", &Paul,  CMoney(100.0, "$"));

   cout << "Daten der Konten:" << endl << "=================" << endl << endl;
   cout << "Konto 1:" << endl;   Konto1.print();         cout << endl << endl;
   cout << "Konto 2:" << endl;   Konto2.print();         cout << endl << endl;
   cout << "Konto 3:" << endl;   Konto3.print();         cout << endl << endl;
   cout << "Konto 4:" << endl;   Konto4.print();         cout << endl << endl;

   cout << "Daten der Banken:" << endl << "=================" << endl << endl;
   cout << "Bank 1:"  << endl;   Spasskasse.print();     cout << endl << endl;
   cout << "Bank 2:"  << endl;   HochschulBank.print();  cout << endl << endl;

   return 0;
}
```

## Beispiel-Ausgabe:

```
Daten der Konten:
=================

Konto 1:
Kunde     : Egon Muster (Kd-Nr. 4711)
IBAN / BIC: DE99 1234 5678 1234 5678 90 / BESPKADEXXX
Kontostand: 150.00 EUR

Konto 2:
Kunde     : Egon Muster (Kd-Nr. 4711)
IBAN / BIC: DE998765432109876543 21 / DEHOBADEXXX
Kontostand: 150.00 EUR
Dispo     : 250.00 EUR

Konto 3:
Kunde     : Anton Muster (Kd-Nr. 815)
IBAN / BIC: DE11223344556677889900 / BESPKADEXXX
Kontostand: 150.00 EUR
Sparzinsen: -2.50 %

Konto 4:
Kunde     : Paul Muster (Kd-Nr. 1234)
IBAN / BIC: DE99 8877 6655 4433 2211 00 / DEHOBADEXXX
Kontostand: 100.00 $

Daten der Banken:
=================

Bank 1:
Berliner Spasskasse
BLZ 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 |     150.00 EUR


Bank 2:
Deutsche Hochschul-Bank
BLZ DEHOBADEXXX
Anzahl Konten: 2
Kontenliste:
IBAN                         | Kundenname           | Anz.Buchungen | Kontostand
-----------------------------|----------------------|---------------|----------------
DE99 8765 4321 0987 6543 21 | Egon Muster          |             0 |     150.00 EUR
DE99 8877 6655 4433 2211 00 | Paul Muster          |             0 |     100.00 $


CAccount:             Konto (DE99 8877 6655 4433 2211 00) wird vernichtet!
CSavingsAccount:      Konto (DE11 2233 4455 6677 8899 00) wird vernichtet!
CAccount:             Konto (DE11 2233 4455 6677 8899 00) wird vernichtet!
CCurrentAccount:      Konto (DE99 8765 4321 0987 6543 21) wird vernichtet!
CAccount:             Konto (DE99 8765 4321 0987 6543 21) wird vernichtet!
CAccount:             Konto (DE99 1234 5678 1234 5678 90) wird vernichtet!
CAccount:             Konto (DE99 8765 4321 0987 6543 21) wird vernichtet!
```