

---

## Lehrveranstaltung "Objektorientierte Programmierung" Übungsblatt 2

### 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 **03. November 2023**.

---

**Aufgabe:** In der zweiten Übungsaufgabe des Projektes „Banken“ sollen die Klassen `CMoney`, `CAddress`, `CCustomer` und `CAccount` erstellt werden.

Die Klasse `CMoney` 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 benötigt; die Ausgabe soll der Beispielausgabe entsprechen.

Die Klasse `CAddress` erhält als private Eigenschaften Straße mit Hausnummer, Postleitzahl und Ort (jeweils `string`). Auch hier werden erst einmal der Konstruktor (kein Standardkonstruktor! d.h. es sollen keine „leeren“ Adressen existieren), `set`- und `get`- sowie eine `print`-Methode (siehe Beispielausgabe) benötigt.

Die Klasse `CCustomer` erhält als private Eigenschaften folgende Daten vom Kunden: Kundennummer (`long`), Name (`string`), Adresse (Klasse `CAddress`), Geburtstag (Klasse `CDate`) und einen Vektor von Zeigern auf Konten (Klasse `CAccount`). Als Methoden sollen auch hier ein Konstruktor (kein Standardkonstruktor!), mit dem die Kundennummer, der Name und die Adresse gesetzt werden (die Kontenliste ist anfangs leer), sowie `set`, `get` und `print` implementiert werden. Zusätzlich wird noch eine Methode `addAccount` benötigt, mit der dem Kunden ein Zeiger auf ein Konto hinzugefügt werden kann.

Die Klasse `CAccount` erhält als private Eigenschaften folgende Daten: die Kontonummer (IBAN; `string`), einen Zeiger auf den Kontoinhaber (Klasse `CCustomer`) sowie den Kontostand (Klasse `CMoney`). Der Zeiger auf die Bank sowie die Buchungsliste werden erst in einer der nächsten Übungsaufgaben nachgetragen, wenn die entsprechenden Klassen existieren. Wie üblich werden hier auch Konstruktor, `set`, `get` und `print`-Methoden benötigt. Bei der Ausgabe der IBAN soll diese in 4er-

Gruppen ausgegeben werden.

Beim Setzen des Zeigers auf den Kunden muss zusätzlich beim Kunden der Zeiger auf das Konto in der Kontenliste des Kunden gesetzt werden (Methode `addAccount` in der Klasse `CCustomer`).

Allgemeine Hinweise:

Es steht in den Aufgaben immer, dass die `set-` und `get-`Methoden implementiert werden sollen, aber Sie erstellen diese Methoden immer nur dann, wenn sie tatsächlich benötigt werden.

In den `print-`Methoden wird am Ende grundsätzlich kein Zeilenumbruch ausgegeben. Wenn der Aufrufer der `print-`Methode anschließend einen Zeilenumbruch haben möchte, muss er es selber machen. Aber vielleicht soll nach der Ausgabe in der gleichen Zeile weitergeschrieben werden (z.B. bei Tabellenausgaben), dann wäre ein abschließender Zeilenumbruch in der `print-`Methode nicht hilfreich.

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 "ccustomer.h"

int main()
{
    CTime Jetzt, SUBeginn(16, 0);
    CDate Heute, Geburtsdatum(7, 7, 1977);
    CMoney Startkapital(150.0);
    CAddress Adresse("Mustergasse 3a", "D - 99889", "Musterstadt");
    CCustomer Kunde(4711, "Egon Muster", Geburtsdatum, Adresse);
    CAccount Konto1("DE99123456781234567890", &Kunde, Startkapital);

    cout << "Dieses Programm wurde am ";
    Heute.print();
    cout << " um ";
    Jetzt.print();
    cout << " gestartet." << endl << endl;

    cout << "Das Startkapital fuer alle Kunden betraegt ";
    Startkapital.print();
    cout << "." << endl << endl;

    cout << "Personenbezogene Daten des Bank-Kunden:" << endl;
    Kunde.print();
    cout << endl << endl;

    cout << "Daten des Konto 1:" << endl;
    Konto1.print();
    cout << endl << endl;

```

```
CAccount Konto2("DE99876543210987654321", &Kunde, CMoney(100.0, "$"));

cout << "Personenbezogene Daten des Bank-Kunden (ein Konto mehr):" <<
endl;
Kunde.print();
cout << endl << endl;

cout << "Daten des Konto 2:" << endl;
Konto2.print();
cout << endl << endl;

return 0;
}
```

### Beispiel-Ausgabe:

Dieses Programm wurde am 12.08.2023 um 12:27:08 gestartet.

Das Startkapital fuer alle Kunden betraegt 150.00 EUR.

Personenbezogene Daten des Bank-Kunden:

Egon Muster (Kd-Nr. 4711)

Mustergasse 3a

D - 99889 Musterstadt

geboren am: 07.07.1977

Konten:

- Kontonr.: DE99 1234 5678 1234 5678 90

Daten des Konto 1:

Kunde: Egon Muster (Kd-Nr. 4711)

IBAN: DE99 1234 5678 1234 5678 90

Kontostand: 150.00 EUR

Personenbezogene Daten des Bank-Kunden (ein Konto mehr):

Egon Muster (Kd-Nr. 4711)

Mustergasse 3a

D - 99889 Musterstadt

geboren am: 07.07.1977

Konten:

- Kontonr.: DE99 1234 5678 1234 5678 90

- Kontonr.: DE99 8765 4321 0987 6543 21

Daten des Konto 2:

Kunde: Egon Muster (Kd-Nr. 4711)

IBAN: DE99 8765 4321 0987 6543 21

Kontostand: 100.00 \$