
Lehrveranstaltung "Objektorientierte Programmierung"

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 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 **28. April 2023**.

Aufgabe: In der zweiten Übungsaufgabe des Projektes „Bücherei“ sollen die Klassen `CAddress`, `CPerson` und `CLocation` erstellt werden.

Die Klasse `CAddress` erhält als private Eigenschaften die Daten einer Anschrift: `Street`, `Number`, `Zipcode` und `Town` (alles `strings`). Neben dem Konstruktor (**keinen Standardkonstruktor!** - siehe Initialisierungslisten) werden hier erst einmal `set`- und `get`-Methoden sowie eine `print`-Methode (Ausgabe der Adresse in zwei Zeilen; erste Zeile Straße und Hausnr., zweite Zeile PLZ und Ort; siehe Beispielausgabe) benötigt.

Die Klasse `CPerson` erhält einen Namen (`Name`; `string`), eine Anschrift (`Address`; Klasse `CAddress`) und ein Geburtsdatum (`Birthday`; Klasse `CDate`). Auch hier werden erst einmal Konstruktor, `set`- und `get`-Methoden sowie eine `print`-Methode, die die Daten der Person optisch wertvoll (siehe Beispielausgabe) ausgibt, benötigt.

Analog kann die Klasse `CLocation` erstellt werden; hier sind als private Eigenschaften die Abteilung und das Regal (`Section` und `Rack`) als `strings` zu verwenden. Benötigt werden hier zwei Konstrukturen (Standardkonstruktor setzt die Eigenschaften auf den Raum „Buero“ und „Fach zum Einsortieren“ - sozusagen der erste Ort für neue Medien zwecks Registrierung), die üblichen `set`- und `get`-Methoden sowie eine `print`-Methode (siehe Beispielausgabe).

Hinweise:

1. In der Aufgabe steht wohl immer, dass `set`- und `get`-Methoden erstellt werden sollen, aber dies nur dann, wenn diese auch verwendet werden. D.h. diese Methoden sollen nicht auf Verdacht bzw. auf Vorrat programmiert werden.

2. Die `print`-Methoden sollen am Ende der Methode keinen Zeilenumbruch ausgeben. Wird nach der Ausgabe ein Zeilenumbruch gewünscht, muss der Aufrufer dieser Methode den Zeilenumbruch anschließend selber ausgeben (siehe auch Hauptprogramm).

Zum Testen der Klassen soll wieder das vorgegebene Hauptprogramm verwendet werden, das Objekte der drei Klassen erzeugt, auf verschiedene Werte setzt und wieder auf dem Bildschirm ausgibt.

Hauptprogramm:

```
#include <iostream>

using namespace std;

#include "cdate.h"
#include "clocation.h"
#include "caddress.h"
#include "cperson.h"

int main()
{
    CLocation L1, L2("Technik", "Computer"); // Technik-Abteilung im Computer-Regal
    CDate Geburtstag(7, 11, 1984);
    CAddress A1("Luxemburger Str.", "10", "D-13351", "Berlin");
    CAddress Adresse("Berliner Str.", "21-23", "10876", "Berlin");
    CPerson Egon("Egon Mustermann", Adresse, Geburtstag);

    cout << "Klasse CLocation:" << endl;
    cout << "Standardkonstruktor fuer neue Buecher:    ";
    L1.print();
    cout << endl;
    cout << "Konstruktor Technik-Abt., Computer-Regal: ";
    L2.print();
    cout << endl;

    cout << "\nKlasse CAddress:" << endl;
    cout << "Konstruktor Adresse der Berliner Hochschule fuer Technik:\n";
    A1.print();
    cout << endl;

    cout << "\nKlasse CPerson:" << endl;
    cout << "Konstruktor Egon Mustermann:\n";
    Egon.print();
    cout << endl;

    return 0;
}
```

Beispiel-Ausgabe:

```
Klasse CLocation:
Standardkonstruktor fuer neue Buecher:    Abt.: Buero; Regal: Fach zum Einsortieren
Konstruktor Technik-Abt., Computer-Regal: Abt.: Technik; Regal: Computer

Klasse CAddress:
Konstruktor Adresse der Berliner Hochschule fuer Technik:
Luxemburger Str. 10
D-13351 Berlin

Klasse CPerson:
Konstruktor Egon Mustermann:
Egon Mustermann
Berliner Str. 21-23
10876 Berlin
* 07.11.1984
```