
Lehrveranstaltung "Objektorientierte Programmierung"

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 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 **25. November 2022**.

Aufgabe: In der 4. Übungsaufgabe des Projektes „Belegungen“ sollen die Klassen `CStudent` und `CTeacher` (jeweils von der Klasse `CPerson` abgeleitet), `CStudy` (Studiengang), `CSubject` (Studienfach) und `CBooking` (Belegung) erstellt werden.

Die private Eigenschaft `ID` der Klasse `CPerson` hatten wir bisher nicht genutzt. Dies soll sich jetzt ändern: Die `ID` soll eine laufende Nummer (`unsigned`) sein und automatisch vergeben werden. Dazu muss eine `static`-Eigenschaft in dieser Klasse angelegt werden, die mit 0 initialisiert wird. Diese wird bei jedem Konstruktoraufwurf um 1 erhöht und dann der neuen Person als ID zugewiesen.

Die Klasse `CStudent` erbt von der Klasse `CPerson` und erhält zusätzlich als private Eigenschaften die Matrikelnummer (`unsigned int`), das aktuelle Fachsemester (`unsigned short`), die Anzahl der erreichten Credits (`unsigned`) und einen Zeiger auf den Studiengang (Klasse `CStudy`). Neben einem Konstruktor und eventuell nötigen `set`- und `get`-Methoden wird eine `print`-Methode benötigt, die den Namen des Studenten sowie in Klammern dessen Geburtstag und Matrikelnummer ausgibt (siehe Beispiel-Ausgabe).

Die Klasse `CTeacher` erbt ebenfalls von der Klasse `CPerson` und erhält zusätzlich als private Eigenschaft die Personalnummer (`unsigned`). Auch hier werden Konstruktor und gegebenenfalls `set`- und `get`-Methoden sowie eine `print`-Methode benötigt, die den Namen des Dozenten sowie in Klammern die Personalnummer ausgibt (siehe Beispiel-Ausgabe).

Die Klasse `CStudy` erhält als private Eigenschaften den Namen des Studiengangs, einen Wahrheitswert, ob es einen Numerus clausus für diesen Studiengang gibt, und die Semesteranzahl der Regelstudienzeit. Die `print`-Methode dieser Klasse soll nur den Namen des Studiengangs ausgeben.

Die Klasse `CSubject` erhält als private Eigenschaften den Namen des Studienfachs, eine Nummer (`unsigned`), einen Zeiger auf den Studiengang (Klasse `CStudy`) und einen Vektor von Veranstaltungen (Klasse `CEvent`). Hauptsächlich wird hier (ähnlich wie in der Klasse `CEvents`) eine `addEvent`- und eine `print`-Methode benötigt. Die Ausgabe soll den Namen des Studienfachs und in Klammern die Nummer und den Namen des Studiengangs ausgeben (siehe Beispiel-Ausgabe).

Die Klasse `CBooking` erhält als private Eigenschaften eine Belegungsnummer (soll genauso wie die `ID` in der Klasse `CPerson` automatisch vergeben werden), je einen Zeiger auf das Fach (Klasse `CSubject`) und auf den Studenten (Klasse `CStudent`) sowie Datum und Uhrzeit der Belegung. Die `print`-Methode soll die 4 Zeilen einer Belegung ausgeben – siehe Beispiel-Ausgabe.

Die fünf neuen Klassen sowie die Klasse `CPerson` sollen jetzt außerdem einen Destruktor erhalten. In diesen soll jeweils ausgegeben werden, welches Objekt gerade vernichtet wird. Hiermit kann sehr gut kontrolliert werden, ob die Vererbung korrekt implementiert wurde. Auch für die kommenden Aufgaben ist es eine gute Kontrolle, ob alle Objekte korrekt vernichtet werden.

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

vorgegebenes Hauptprogramm:

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

using namespace std;

#include "ctime.h"
#include "cblock.h"
#include "croom.h"
#include "cperson.h"
#include "cstudent.h"
#include "cteacher.h"
#include "cevent.h"
#include "cstudy.h"
#include "csubject.h"
#include "cbooking.h"

int main()
{
    unsigned i;

    // Uhrzeiten
    CTime *Zeit0800 = new CTime( 8, 0);
    CTime *Zeit1000 = new CTime(10, 0);
    CTime *Zeit1215 = new CTime(12, 15);
    CTime *Zeit1415 = new CTime(14, 15);
    CTime *Zeit1600 = new CTime(16, 0);
    CTime *Zeit1745 = new CTime(17, 45);
    CTime *Zeit1930 = new CTime(19, 30);

    // Bloecke
    CBlock *Block1 = new CBlock( 1, *Zeit0800);
    CBlock *Block2 = new CBlock( 2, *Zeit1000);
    CBlock *Block3 = new CBlock( 3, *Zeit1215);
    CBlock *Block4 = new CBlock( 4, *Zeit1415);
    CBlock *Block5 = new CBlock( 5, *Zeit1600);
    CBlock *Block6 = new CBlock( 6, *Zeit1745);
    CBlock *Block7 = new CBlock( 7, *Zeit1930);
```

```

// Raeume
CRoom *D113 = new CRoom("D 113", "Haus Bauswesen", 24);
CRoom *D114 = new CRoom("D 114", "Haus Bauswesen", 28);
CRoom *D117 = new CRoom("H3", "Haus Bauwesen", 104);
CRoom *D136 = new CRoom("H5", "Haus Bauwesen", 104);
CRoom *D209 = new CRoom("D 209", "Haus Bauwesen", 35);
CRoom *D419 = new CRoom("D 419", "Haus Bauswesen", 40);
CRoom *B301 = new CRoom("B 301", "Haus Gauss", 100);

// Studiengaenge
CStudy *TechInfo = new CStudy("Technische Informatik", false, 8);
CStudy *Physik = new CStudy("Physik", true, 10);
CStudy *ETechnik = new CStudy("Elektrotechnik", false, 8);

// Personen
CStudent *Student1 = new CStudent("Anna Albrecht", "Albrechtstr.", "15a", 12045, "Berlin",
15, 9, 1989, 191278, 2, 28, TechInfo);
CStudent *Student2 = new CStudent("Detlef Dicht", "Dichterweg", "27", 29031, "Vordertupfingen",
27, 5, 1991, 191115, 3, 31, Physik);
CStudent *Student3 = new CStudent("Frank Faust", "Goethestr.", "3-5", 12345, "Berlin",
3, 11, 1987, 191492, 2, 24, TechInfo);
CStudent *Student4 = new CStudent("Max Muster", "Meistergasse", "23", 19885, "Hintertupfingen",
19, 7, 1985, 191368, 3, 30, Etechnik);
CPerson *Dozent1 = new CTeacher("Kevin Kaufmann", "Kaiserstr.", "38", 12075, "Berlin",
15, 6, 1959, 12);
CPerson *Dozent2 = new CTeacher("Nathan Neuling", "Neue Str.", "7", 10777, "Berlin",
9, 9, 1969, 97);
CPerson *Dozent3 = new CTeacher("Sabine Sauber", "Saubere Gasse", "1", 17001, "Auf dem Land",
3, 1, 1962, 68);
CPerson *Dozent4 = new CTeacher("Willi Witzig", "Wunderweg", "99", 15911, "Nauen",
9, 11, 1976, 34);

// Veranstaltungen
CEvent *Event1 = new CEvent("SU OOP", "", Dozent1, D136, Block4, Fr, 1);
CEvent *Event2 = new CEvent("Ueb OOP", "", Dozent1, D113, Block5, Fr, 2);
CEvent *Event3 = new CEvent("Ueb OOP", "", Dozent1, D113, Block6, Fr, 2);
CEvent *Event4 = new CEvent("SU Mathe III", "", Dozent2, B301, Block2, Do, 1);
CEvent *Event5 = new CEvent("SU Mathe III", "", Dozent2, B301, Block3, Do, 1);
CEvent *Event6 = new CEvent("Ueb Mathe III", "", Dozent2, D209, Block1, Do, 1);
CEvent *Event7 = new CEvent("Englisch", "", Dozent3, D419, Block5, Mi, 1);
CEvent *Event8 = new CEvent("SU Systemprogrammierung", "", Dozent4, D117, Block3, Di, 1);
CEvent *Event9 = new CEvent("Ueb Systemprogrammierung", Dozent4, D114, Block4, Di, 1);

// Faecher einschl. der Zuordnung zu den Veranstaltungen
CSubject *Fach1 = new CSubject("Mathe III", 31103, TechInfo);
CSubject *Fach2 = new CSubject("Systemprogrammierung", 33102, Etechnik);
CSubject *Fach3 = new CSubject("OOP", 31101, TechInfo);
CSubject *Fach4 = new CSubject("Englisch", 31209, TechInfo);
CSubject *Fach5 = new CSubject("Quantenmechanik", 32502, Physik);
Fach1->addEvent(Event4);
Fach1->addEvent(Event5);
Fach1->addEvent(Event6);
Fach2->addEvent(Event8);
Fach2->addEvent(Event9);
Fach3->addEvent(Event1);
Fach3->addEvent(Event2);
Fach3->addEvent(Event3);
Fach4->addEvent(Event7);
// Quantenmechanik faellt dieses Semester wegen Personalmangel aus

// Belegungen
vector<CBooking*> Belegungen;
Belegungen.push_back(new CBooking(Fach3, Student1, 4, 10, 2022, 5, 17, 29));
Belegungen.push_back(new CBooking(Fach2, Student1, 4, 10, 2022, 5, 19, 43));
Belegungen.push_back(new CBooking(Fach1, Student1, 4, 10, 2022, 5, 23, 7));
Belegungen.push_back(new CBooking(Fach1, Student2, 4, 10, 2022, 13, 28, 56));
Belegungen.push_back(new CBooking(Fach1, Student3, 4, 10, 2022, 17, 3, 31));
Belegungen.push_back(new CBooking(Fach1, Student4, 5, 10, 2022, 1, 52, 11));

for (i = 0; i < Belegungen.size(); i++)
{
    Belegungen[i]->print();
    cout << endl << endl;
}
cout << endl;

// Speicherbereiche freigeben
for (i = 0; i < Belegungen.size(); i++)
    delete Belegungen[i];

delete Fach1;      delete Fach2;      delete Fach3;
delete Fach4;      delete Fach5;

delete Event1;     delete Event2;     delete Event3;
delete Event4;     delete Event5;     delete Event6;
delete Event7;     delete Event8;     delete Event9;

delete Student1;   delete Student2;
delete Student3;   delete Student4;
delete Dozent1;    delete Dozent2;
delete Dozent3;    delete Dozent4;

```

```

delete TechInfo;      delete Physik;      delete Etechnik;

delete D113;          delete D114;          delete D117;
delete D136;          delete D209;          delete D419;
delete B301;

delete Block1;        delete Block2;        delete Block3;
delete Block4;        delete Block5;        delete Block6;
delete Block7;

delete Zeit0800;       delete Zeit1000;       delete Zeit1215;
delete Zeit1415;       delete Zeit1600;       delete Zeit1745;
delete Zeit1930;

return 0;
}

```

Beispiel-Ausgabe:

Belegung Nr. 1:
 Am 04.10.2022 um 05:17:29
 hat Anna Albrecht (* 15.09.1989; MatrNr. 191278)
 das Fach 'OOP (31101; Technische Informatik)' belegt.

Belegung Nr. 2:
 Am 04.10.2022 um 05:19:43
 hat Anna Albrecht (* 15.09.1989; MatrNr. 191278)
 das Fach 'Systemprogrammierung (33102; Elektrotechnik)' belegt.

Belegung Nr. 3:
 Am 04.10.2022 um 05:23:07
 hat Anna Albrecht (* 15.09.1989; MatrNr. 191278)
 das Fach 'Mathe III (31103; Technische Informatik)' belegt.

Belegung Nr. 4:
 Am 04.10.2022 um 13:28:56
 hat Detlef Dicht (* 27.05.1991; MatrNr. 191115)
 das Fach 'Mathe III (31103; Technische Informatik)' belegt.

Belegung Nr. 5:
 Am 04.10.2022 um 17:03:31
 hat Frank Faust (* 03.11.1987; MatrNr. 191492)
 das Fach 'Mathe III (31103; Technische Informatik)' belegt.

Belegung Nr. 6:
 Am 05.10.2022 um 01:52:11
 hat Max Muster (* 19.07.1985; MatrNr. 191368)
 das Fach 'Mathe III (31103; Technische Informatik)' belegt.

Belegung Nr. 1 wird vernichtet.
 Belegung Nr. 2 wird vernichtet.
 Belegung Nr. 3 wird vernichtet.
 Belegung Nr. 4 wird vernichtet.
 Belegung Nr. 5 wird vernichtet.
 Belegung Nr. 6 wird vernichtet.
 Studienfach Mathe III wird vernichtet.
 Studienfach Systemprogrammierung wird vernichtet.
 Studienfach OOP wird vernichtet.
 Studienfach Englisch wird vernichtet.
 Studienfach Quantenmechanik wird vernichtet.
 Student*in Anna Albrecht wird vernichtet.
 Person Anna Albrecht wird vernichtet.
 Student*in Detlef Dicht wird vernichtet.
 Person Detlef Dicht wird vernichtet.
 Student*in Frank Faust wird vernichtet.
 Person Frank Faust wird vernichtet.
 Student*in Max Muster wird vernichtet.
 Person Max Muster wird vernichtet.
 Lehrer*in Kevin Kaufmann wird vernichtet.
 Person Kevin Kaufmann wird vernichtet.
 Lehrer*in Nathan Neuling wird vernichtet.
 Person Nathan Neuling wird vernichtet.
 Lehrer*in Sabine Sauber wird vernichtet.
 Person Sabine Sauber wird vernichtet.
 Lehrer*in Willi Witzig wird vernichtet.
 Person Willi Witzig wird vernichtet.
 Studiengang Technische Informatik wird vernichtet.
 Studiengang Physik wird vernichtet.
 Studiengang Elektrotechnik wird vernichtet.