

Dortmund, den 31. Dezember 2010

Übungen zur Vorlesung EINI (WS 2010)

Blatt 9

Block blau

Es können 4 Punkte erreicht werden.

Abgabedatum: 09. Januar 2011 23:59 Uhr

Hinweise

1) Bitte beachten Sie die aktuellen Hinweise unter

<http://www4.cs.uni-dortmund.de/home/maeter/UeEiniET10W/>

Für die Abgabe sind die Dateien: *Aufgabe_09_0.txt* und *Aufgabe_09_1.cpp* zu erstellen.

- 2) Es ist ratsam, die Programme vor der Abgabe zu kompilieren und auszuführen.
- 3) Verwenden Sie für die Textaufgaben reine Texteditoren. Das Abgabesystem erkennt **keine** Word- oder PDF-Dateien!!!
- 4) In der optionalen Datei *Anmerkungen.txt* können Sie allgemeine Anmerkungen bezüglich Ihrer Lösungen notieren.
- 5) Machen Sie sich im Vorhinein Gedanken, wie Sie die zu erstellenden Programme zu implementieren sind. Machen Sie sich also zuerst einen Programmentwurf.
- 6) Kommentieren Sie Ihre Quelltexte. Das gehört zur Aufgabenstellung und wird mitbewertet.
- 7) Verwenden Sie bei Ihrer Implementierung den C++-Standard (insbesondere bei der Deklaration von Feldern).

Spruch des Tages (frei nach Murphy)

Die Ergebnisse eines Programms sollten immer reproduzierbar sein. Sie sollten alle auf die gleiche Art danebengehen.

Aufgaben

Aufgabe 0: Grundlagen (1 Punkt)

Legen Sie für Ihre Antworten eine Text-Datei *Aufgabe_09_0.txt* an.

- Welche Ziele werden bei der Verwendung von Klassen verfolgt? (0,2 Punkte)
- Welches ist der wesentliche Unterschied zwischen einer `struct`- und `class`-Definition? (0,2 Punkte)
- Erklären Sie das Prinzip des 'information hiding'. (0,2 Punkte)
- Welche Aufgaben haben Konstruktoren? (0,2 Punkte)
- Welche Aufgaben haben Destruktoren? (0,2 Punkte)

Speichern Sie Ihre Ergebnisse in der Ergebnisdatei *Aufgabe_09_0.txt*.

Aufgabe 1: Klassen (3 Punkte)

Modellieren Sie eine Klasse `Bericht`, mit der Berichte in einer Bibliothek dargestellt werden können. `Bericht` hat dabei (hier für die Aufgabenstellung vereinfacht) folgende Daten:

- einen Titel, bestehend aus einer Zeichenkette
- einen Umfang, als Seitenanzahl
- eine Berichtsnummer
- eine Markierung, ob der Bericht momentan ausgeliehen ist
- die Ausweisnummer der ausleihenden Person

`Bericht` soll folgende Operationen unterstützen:

- Erzeugen eines Berichtes mit Titel, Berichtsnummer und Seitenanzahl
- Ausgabe (auf dem Bildschirm) des Titels, Berichtsnummer und Seitenanzahl des Berichtes
- Ausleihen des Berichtes mit einer Ausweisnummer
- Abfrage, ob der Bericht ausgeliehen ist
- Abfrage, wer den Bericht ausgeliehen hat
- Rückgabe des Berichtes

Legen Sie eine cpp-Datei *Aufgabe_09_1.cpp* an und gehen Sie in folgenden Schritten vor:

- Deklariieren Sie eine Klasse `Bericht`. (0,5 Punkte)
- Geben sie der Klasse `Bericht` entsprechend der obigen Aufzählung passende Attribute. (0,5 Punkte)

- c) Geben Sie der Klasse `Bericht` entsprechend der obigen Aufzählung einen passenden Konstruktor und implementieren Sie diesen. (0,25 Punkte)
- d) Fügen Sie der Klasse die passenden Methoden hinzu und implementieren Sie diese. (0,5 Punkte)
- e) Achten Sie darauf, welche Teile der Klasse öffentlich zugänglich sein sollen und welche nicht. (0,25 Punkte)
- f) Schreiben Sie in der `main`-Funktion einen Testfall in dem zuerst zwei Berichte erstellt und auf dem Bildschirm ausgegeben werden, dann versucht wird einen der Berichte von zwei Personen auszuleihen. Danach gibt die Person, die den Bericht erfolgreich ausgeliehen hat den Bericht zurück und die zweite Person versucht es nochmals den Bericht auszuleihen. Kopieren Sie die Ergebnisse als Kommentar an das Ende der cpp-Datei `Aufgabe_09_1.cpp`. (1 Punkt)

(Präsenz-)Aufgabe 2: Fehlersuche 0 Punkte)

Hinweis: Bei dieser Aufgabe handelt es sich um eine leicht modifizierte ehemalige Prüfungsaufgabe.

Das folgende Programm führt den Umgang mit zweidimensionalen Feldern vor. Jedes dieser Felder stellt eine $m \times n$ Matrix dar. Es werden die Matrizen `x` und `y` aufaddiert. Das Ergebnis wird in Matrix `z` abgespeichert. Dazu wird die - zugegebenermaßen recht einfache - Funktion `Plus` genutzt. Die Ausgabe eines Testlaufes finden Sie am Ende der Aufgabe.

Leider haben sich beim Abtippen des Programms einige Fehler eingeschlichen. Geben Sie die Zeilennummer der Fehler und eine geeignete Korrektur an. Beachten Sie, dass ein Tippfehler nicht unbedingt zu einem Syntaxfehler führt, sondern möglicherweise eine Änderung der Semantik zur Folge hat.

Markieren Sie die Fehler und korrigieren Sie die Zeilen.

```

/* 1*/ #include <string>
/* 2*/ using namespace std;
/* 3*/
/* 4*/ int Plus(int a; int b) {
/* 5*/     int ergebnis = a + b;
/* 6*/     return;
/* 7*/ }
/* 8*/
/* 9*/ int main() {
/*10*/     const int m=2, n=3;
/*11*/     int x[m][n], y[m][n], z[m][n]
/*12*/     / Matrizen x und y einlesen
/*13*/     cout << "Geben Sie die Werte fuer x und y ein:\n";
/*14*/     For (int i=0; i<m; i++)
/*15*/         for (int j=0; j<n; j) {
/*16*/             cout << "Wert x[" << i << ", " << j << "]? :";
/*17*/             cin >> x[i][j];
/*18*/             cout << "Wert y[" << i << ", " << j << "]? :";
/*19*/             cin << y[i][j];
/*20*/         }

```

```

/*21*/ // Summe von x+y berechnen
/*22*/ for (int v=0; v<n; v++)
/*23*/     for (int u=0; u<m; u+=1)
/*24*/         z[u][v]=Plus(x[u][v],y[v][u]);
/*25*/ // Ergebnis ausgeben
/*26*/ cout << Das Ergebnis z lautet:\n;
/*27*/ for (int a=m-1; a>=0; a--)
/*28*/     for (int b=n-1; b>=-1; b--) {
/*29*/         cout << "z[" << a << ", ";
/*30*/         cout << b << "]=" << z[a][b] << Entlein;
/*31*/     }
/*32*/ return 0;
/*33*/ }

```

Ein Testlauf des **fehlerfreien** Programms hat zu der folgenden Ausgabe geführt:

Geben Sie die Werte fuer x und y ein :

```

Wert x[0,0]? :1
Wert y[0,0]? :2
Wert x[0,1]? :3
Wert y[0,1]? :4
Wert x[0,2]? :5
Wert y[0,2]? :6
Wert x[1,0]? :7
Wert y[1,0]? :8
Wert x[1,1]? :9
Wert y[1,1]? :10
Wert x[1,2]? :11
Wert y[1,2]? :12

```

Das Ergebnis z lautet :

```

z[1,2]=23
z[1,1]=19
z[1,0]=15
z[0,2]=11
z[0,1]=7
z[0,0]=3

```