

Dortmund, den 12. November 2010

Übungen zur Vorlesung EINI (WS 2010)

Blatt 5

Block rot

Es können 4 Punkte erreicht werden.

Abgabedatum: 21. November 2009 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_05_0.txt* und *Aufgabe_05_1.cpp* zu erstellen.

- 2) In der optionalen Datei *Anmerkungen.txt* können Sie allgemeine Anmerkungen bezüglich Ihrer Lösungen notieren.
- 3) Es ist ratsam, die Programme vor der Abgabe zu kompilieren und auszuführen.
- 4) Verwenden Sie für die Textaufgaben reine Texteditoren. Das Abgabesystem erkennt **keine** Word- oder PDF-Dateien!!!

Spruch des Tages (frei nach Murphy)

**Würden Architekten Häuser bauen, wie
Programmierer ihre Programme, dann könnte
ein Specht die ganze Zivilisation zerstören.**

Aufgaben

Aufgabe 0: Grundlagen (1 Punkt)

Legen Sie für Ihre Antworten eine Text-Datei *Aufgabe_05_0.txt* an.

- Welche Vorteile haben Funktionen (0,2 Punkte) ?
- Was versteht man unter Übergabe eines Wertes oder `call by value` (0,2 Punkte) ?
- Wo wird hauptsächlich die Übergabe einer Referenz oder `call by reference` eingesetzt (0,2 Punkte) ?
- Von welchem Typ ist der Rückgabewert der Funktion `void function1 (int a){...}` (0,2 Punkte) ?
- In welcher Bibliothek ist die Standardfunktion `sin (x)` abgelegt (0,2 Punkte) ?

Speichern Sie Ihre Ergebnisse in der Ergebnisdatei *Aufgabe_05_0.txt*.

Aufgabe 1: Statistiken (3 Punkte)

Auf Übungsblatt 2 haben Sie bereits ein Programm geschrieben, welches das *arithmetische Mittel* und das *harmonische Mittel* berechnet. Ändern Sie Ihren Programmcode nun so, dass diese beiden Werte jeweils in einer Funktion ermittelt werden. Diesen Funktionen soll ein Array mit Elementen vom Typ `double` übergeben werden. Der Rückgabewert soll jeweils vom Typ `double` sein.

Legen Sie eine cpp-Datei *Aufgabe_05_1.cpp* an und ergänzen Sie darin folgendes Programm:

```
1  #include <iostream>
2  using namespace std;
3
4  // Funktion arith fuer das arithmetische Mittel
5
6  // Funktion harm fuer der harmonische Mittel
7
8
9  int main() {
10     double a[11] = {12.6,33.77,51.32,95.11,19.78,
11                   35.34,85.18,89.0,37.5,13.5,24.1};
12
13     cout << "Arithmetisches Mittel: " << arith(11, a) << endl;
14     // Ergebnis =
15     cout << "Harmonisches Mittel: " << harm(11, a) << endl;
16     // Ergebnis =
17     return 0;
18 }
```

Kompilieren und testen Sie Ihr Programm. Ergänzen Sie die Ergebnisse, die Sie für die aufgeführten Funktionsaufrufe erhalten haben, in der Datei *Aufgabe_05_1.cpp*.

Zur Erinnerung: Seien a_1, \dots, a_n reelle Zahlen. Das *arithmetische Mittel* dieser Zahlen ist durch

$$\bar{a}_{arith} = \frac{1}{n} \sum_{i=1}^n a_i = \frac{a_1 + \dots + a_n}{n}$$

und das *harmonische Mittel* dieser Zahlen durch

$$\bar{a}_{\text{harm}} = \frac{n}{\sum_{i=1}^n \frac{1}{a_i}} = \frac{n}{\frac{1}{a_1} + \dots + \frac{1}{a_n}}$$

definiert.

(Präsenz-)Aufgabe 2: Parameterübergabe (0 Punkte)

In der Vorlesung wurden die beiden Konzepte *call by value* und *call by reference* vorgestellt. Welcher Art sind die Aufrufe der folgenden vier Funktionen? Welche Ausgaben erwarten Sie und warum? Vermerken Sie neben den Funktionen den erwarteten Effekt auf die Variablen `x` und `y` und auf die Zeiger `z1` und `z2`.

```
1  #include <iostream>
2  using namespace std;
3
4  void function_1(int x, int y) {
5      y = x;
6  }
7
8  void function_2(int *x, int *y) {
9      y = x;
10 }
11
12 void function_3(int *x, int *y) {
13     *y = *x;
14 }
15
16 void function_4(int x, int& y) {
17     y = x;
18 }
19
20 int main() {
21     int y = 33;
22     int x = 19;
23     int *z1, *z2;
24
25     z1 = &x;
26     z2 = &y;
27
28     function_1(x, y);
29     cout << "x: " << x << " y: " << y << endl;
30     //x:    y:
31
32     y = 33;
33     function_2(z1, z2);
34     cout << "x: " << x << " y: " << y << endl;
35     //x:    y:
36
37     y = 33;
38     function_3(z1, z2);
39     cout << "x: " << x << " y: " << y << endl;
40     //x:    y:
41
42     y = 33;
43     function_4(x, y);
44     cout << "x: " << x << " y: " << y << endl;
45     //x:    y:
46
47     return 0;
48 }
```