

Dortmund, den 19. November 2010

# Übungen zur Vorlesung EINI (WS 2010)

## Blatt 6

Block rot

Es können 4 Punkte erreicht werden.

Abgabedatum: 28. November 2010 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\_06\_0.txt* und *Aufgabe\_06\_1.txt* 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)

**Jede Aufgabe benötigt doppelt so viel Zeit wie Sie ansetzen. Verdoppeln Sie die Zeit, dauert die Aufgabe viermal so lang.**

# Aufgaben

## Aufgabe 0: Grundlagen (1 Punkt)

Legen Sie für Ihre Antworten eine Text-Datei *Aufgabe\_6\_0.txt* an.

- Wo werden globale Variable definiert und wo sind sie sichtbar (0,2 Punkte) ?
- Warum sollten globale Variable nach Möglichkeit vermieden werden (0,2 Punkte) ?
- Welcher Operator ermöglicht einen Zugriff auf global bekannte Objekte (0,2 Punkte) ?
- Wozu dienen Namensräume (0,2 Punkte) ?
- Wann und wie oft werden statische Variable in Funktionen initialisiert (0,2 Punkte) ?

Speichern sie Ihre Ergebnisse in der Ergebnisdatei *Aufgabe\_6\_0.txt*.

## Aufgabe 1: Programmanalyse (3 Punkte)

**Hinweis:** Bei dieser Aufgabe handelt es sich um eine leicht modifizierte Prüfungsaufgabe aus dem Jahr 2008.

Legen sie eine Datei *Aufgabe\_06\_1.txt* an.

Lesen Sie den folgenden Quelltext sorgfältig und beantworten Sie diesbezüglich die Fragen a), b), c) und d). Schreiben Sie die Antworten in die Datei *Aufgabe\_06\_1.txt*.

```
1  #include <iostream>
2  using namespace std;
3  int fkt0(const int a[], int p, int s) {
4  int m = p;
5  for (int i = p; i < s; i++)
6  if( a[m] > a[i] ) m = i;
7  return m;
8  }
9  void fkt1(int a[], int s) {
10 for (int i = 0; i < s; i++){
11 int m = fkt0(a, i, s);
12 int tmp = a[i];
13 a[i] = a[m];
14 a[m] = tmp;
15 }
16 }
17 int fkt2(int a[], int i, int j, int k) {
18 while (i <= j) {
19 int m = (i + j) / 2;
20 if (k > a[m])
21 i = m + 1;
22 else if (k < a[m])
23 j = m - 1;
24 else
25 return m;
26 }
27 return -1;
```

```

28 }
29 void fkt3(int a[], int s) {
30     for (int i=0; i<s; i++)
31         cout << a[i] << " ";
32     cout << endl;
33 }
34 int main(){
35     int v[10]= {10, -4, 3, 9, 0, 7, 3, -5, 2, 8};
36     fkt1(v, 10);
37     fkt3(v, 10);
38     int p=fkt2(v, 0, 9, 7);
39     if ( p >= 0 )
40         cout << "7: " << p << endl;
41     else
42         cout << "7: nicht da." << endl;
43     return 0;
44 }

```

- a) Beschreiben Sie was jede der Funktionen `fkt0`, `fkt1`, `fkt2` und `fkt3` macht. Bitte verwenden Sie nicht mehr als 3 (drei) Zeilen für die Beschreibung jeder Funktion. Beschreibungen, die lediglich ausdrücken, was die einzelnen Zeilen des Quelltexts tun, werden nicht akzeptiert! Sie sollen die globale Aufgabe jeder Funktion beschreiben und nicht was die einzelnen Anweisungen bedeuten. (1 Punkt)
- b) Was zeigt der Bildschirm nach der Ausführung der `main`-Funktion? (1 Punkt)
- c) Wie ändert sich die Ausgabe des Programms, wenn die Zeilen 36 und 37 der `main`-Funktion vertauscht werden? Erklären Sie warum. (0,5 Punkte)
- d) Wie ändert sich die Ausgabe des Programms, wenn stattdessen die Zeilen 36 und 38 der `main`-Funktion vertauscht werden? Erklären Sie warum. (0,5 Punkte)

## (Präsenz-)Aufgabe 2: Gültigkeitsbereich (0 Punkte)

Kennzeichnen Sie die jeweilige Sichtbarkeit der verschiedenen Variablen mit dem Bezeichner `var_b` indem Sie die Variablen mit dem Bezeichner `var_b` folgendermassen umbenennen: `var_b_ZN`, wobei `ZN` der Zeilennummer im Listing entspricht und aus dem Bezeichner die Sichtbarkeit der Variablen abgelesen werden kann, siehe Zeile 4.

```

1  #include <iostream>
2  using namespace std ;
3
4  int var_b = 7; //int var_b_4 = 7;
5
6  void funktion1 () {
7      cout << " Funktion 1" << endl ;
8      cout << "var_b=" << var_b << endl ;
9      int var_b = 10;
10     cout << "var_b=" << var_b << endl ;
11 }
12
13 void funktion2 ( int var_b) {
14     cout << " Funktion 2" << endl ;
15     cout << "var_b=" << var_b << endl ;
16     var_b = var_b + 2;
17     cout << "var_b=" << var_b << endl ;
18 }
19
20 int funktion3 () {
21     cout << " Funktion 3" << endl ;
22     cout << "var_b=" << var_b << endl ;
23     var_b++;
24     if ( var_b > 7 ) {
25         int var_b;
26         cout << "var_b=" << var_b << endl ;
27     }
28     cout << "var_b=" << var_b << endl ;
29     return var_b;
30 }
31
32 int main () {
33     funktion1 ();
34     funktion2 (var_b);
35     var_b = 2 * funktion3 ();
36     cout << " main" << endl;
37     cout << "var_b=" << var_b << endl ;
38     return 0;
39 }

```