

# ”Informatik - Grundlagen”

## Klausur

3. August 2005

### 1. Programmanalyse (6 Punkte)

Gegeben sei folgendes C++-Programm:

```
#include <iostream>
using namespace std;

int p(int a, int b)
{ int x;
  x = a + b;
  cout << " x: " << x << endl;
  if ( x < b) return ++x; else return x--;
}

main()
{int a=3, b=5;
  int feld[16] = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16};
  cout << p(b,-a) << endl << p(-a,b) << endl << feld[p(a,b)] << endl;
}
```

Welche Ausgabe liefert das C++-Programm?

## 2. Dynamische Datenstrukturen (21 Punkte)

Gegeben sei folgendes C++-Programm:

```
#include <iostream>
using namespace std;
struct lelem{
    int a;
    lelem * pNext;
};

main()
{
    lelem * pAnker = new lelem;
    lelem * pNeu;
    lelem * pLast;
    int i,j;
    cin >> j;
    pAnker -> a = j;
    pAnker -> pNext = pAnker;
    pLast = pAnker;
    for ( i = 1; i < 10; i++)
    {
        pNeu = new lelem;
        cin >> j;
        pNeu -> a = i % j;
        pNeu -> pNext = pAnker;
        pLast -> pNext = pNeu;
        pLast = pNeu;
    }
}
```

- a. Stellen Sie die Datenstruktur grafisch dar, die in diesem Programm aufgebaut wird (inklusive aller vorkommenden Zeiger nach Abarbeitung des Programms) bei folgender Eingabefolge: 4 5 6 7 8 9 10 11 12 13 (die Reihenfolge des Aufbaus der dynamischen Datenstruktur muß erkennbar sein).

5 Punkte

- b. Ergänzen Sie das Programm um eine Funktion

```
int count_wert_lower (int wert, int & count, lelem * pAnker1)
```

mit folgender Funktionalität:

In der Funktion **count\_wert** wird die Anzahl der Elemente der Datenstruktur, die durch den Zeiger **pAnker1** adressiert wird, ermittelt, deren Datenwerte **a** einen kleineren Wert haben als der Parameter **wert**. Diese Anzahl

wird der rufenden Funktion über den Referenzparameter **count** mitgeteilt. Die Funktion liefert als Rückgabewert die Anzahl der Elemente, die in der Datenstruktur, die über den Zeiger **pAnker1** adressiert wird, enthalten sind. Es ist nicht davon auszugehen, daß die Anzahl der Elemente in der Datenstruktur, die durch den Zeiger **pAnker1** adressiert wird, bekannt ist. Demonstrieren Sie die Verwendung von **count\_wert\_lower** im Hauptprogramm zur Bestimmung und Anzeige des prozentualen Anteils von Elementen, die einen kleineren Wert haben als ein vom Nutzer einzugebender Wert an der Gesamtanzahl aller Elemente der Datenstruktur, die durch den Zeiger **pAnker** adressiert wird.

11 Punkte

- c. Ergänzen Sie das Hauptprogramm derart, daß alle a-Instanzen (Elemente a) jeweils mit einer Breite von 7 Bildschirmzeichen auf dem Bildschirm angezeigt werden. Dabei sollen 4 Elemente auf einer Bildschirmzeile ausgegeben werden (in der letzten Zeile sind möglicherweise weniger als 4 Elemente auszugeben). Es ist nicht davon auszugehen, daß die Anzahl der Elemente bekannt ist.

5 Punkte

3. Objektorientierte Programmierung (15 Punkte) **ACHTUNG ! nicht für Studenten des Studienganges Mikrosystemtechnik / Mechatronik**

Geben Sie eine Klassendefinition für eine Klasse **kreis** an, welche einen privaten Member (Datenelement) **radius** vom Typ **float** verwaltet. Folgende Methoden sollen in der Klasse enthalten sein:

- Konstruktor mit keinem Parameter, der Member soll mit 0 initialisiert werden (Standardkonstruktor)
- Konstruktor mit einem Parameter vom Typ **float**. Der Member soll mit dem Werten des Parameters initialisiert werden
- Methode **set** mit einem Parameter. Diese Methode setzt den Wert des Members mit dem Wert des Parameters.
- Methode **flaeche**. Diese Methode gibt den Wert der Fläche des durch den privaten Member beschriebenen Kreises zurück.
- Methode **flaeche\_sektor** mit einem Parameter vom Typ **float**. Diese Methode gibt den Wert der Fläche des durch den privaten Member und dem Parameter, welcher als Winkel  $\alpha$  interpretiert werden soll, beschriebenen Kreissegmentes zurück.
- Methode **umfang**. Diese Methode gibt den Wert des Umfanges des durch den privaten Member beschriebenen Kreises zurück.
- Methode **print**. Diese Methode gibt den Wert des Members auf den Bildschirm aus.

Geben Sie die Implementierung der Methoden an (gegebenenfalls inline-Code). Demonstrieren Sie die Verwendung der Klasse **kreis** in einem kurzen Hauptprogramm, in dem **alle** Methoden verwendet werden.

**Hinweis:** Es gelten folgende Formeln:

Kreisfläche:  $f = \pi r^2$

Kreisumfang:  $u = 2\pi r$

Kreissegment:  $s = \frac{\pi r^2 \alpha}{360}$

Pi:  $\pi = 3.141592653589793$ , ist vom Nutzer als Konstante zu definieren

#### 4. Matrixoperationen (12 Punkte)

Für eine Rechteckmatrix  $rm_{8,12}$  (8 \* 12 ganzzahlige Matrixelemente) soll ein C++-Programm mit folgendem Ablauf geschrieben werden:

- Einlesen der Elemente der 1. Spalte der Matrix (die am weitesten links befindliche Spalte) aus einer Datei mit dem Namen *zahlen*. Wenn diese Datei nicht vorhanden ist, so soll das Programm beendet werden.
- Bilden der restlichen Elemente nach folgendem Muster:

$$rm_{ij} = \begin{cases} rm_{i,j-1} & \text{für } i = 1 \\ rm_{i,j-1} + rm_{i-1,j} & \text{sonst} \end{cases}$$

- Bestimmen der Matrixzeile mit der größten Anzahl von negativen Werten (wenn mehrere Zeilen mit gleicher Anzahl ermittelt werden, so soll die Zeile ausgegeben werden, bei der die Maximalzahl zuletzt auftritt). Ausgabe der Zahlen der eben bestimmten Matrixzeile in eine Datei mit dem Namen *maxneg*.

*Hinweis:* Die hier angegebenen Indizes beziehen sich auf die in der Mathematik üblichen Werte, nicht auf die in C++ gebräuchlichen.

#### 5. Funktionen (6 Punkte)

Zur Zeit  $t = 0$  sind in einer Probe  $10^{22}$  radioaktiver Atome eines bestimmten Nuklides vorhanden. Im Verlauf einer Sekunde entstehen durch Zerfall anderer Nuklide  $10^{10}$  dieser Atome neu. Außerdem zerfallen in jeder Sekunde 6% der zu Beginn dieser Sekunde vorhandenen Atome. Gesucht ist die Anzahl der nach  $n$  Sekunden vorhandenen Atome des Nuklides.

- Schreiben Sie eine C++-Funktion **double atome(int n)** die die Zahl der Atome nach  $n$  Sekunden **rekursiv** berechnet.
- Schreiben Sie dazu eine Funktion **main()**, die unter Verwendung der Funktion eine Tabelle ausgibt, in der für jede abgelaufene Minute im Intervall 0 ... 15 Minuten die aktuelle Minutenzahl sowie die Anzahl der Atome aus- gibt.