

Vorname	Nachname	Matrikel-Nr
---------	----------	-------------

Schreiben Sie jede Lösung einer Aufgabe auf die Vorderseite eines neuen Blatts (**nicht** mehrere Lösungen auf ein Blatt) und lassen Sie die **Rückseiten** aller Blätter **leer**. Kennzeichnen Sie jedes Blatt, das Sie abgeben, mit Ihrem **Nachnamen** (Matrikel-Nr, Schuhgröße etc. sind **nicht** nötig).

Diese Klausur ist mein **dritter Versuch** (bitte ankreuzen): **Ja** ☐ **Nein** ☐

Aufgabe 1 (15 Punkte): Schreiben Sie (in C++) ein Unterprogramm entsprechend der folgenden Deklaration:

```
1  int  anzZiffern(int n, int b);
2      // Wieviele Ziffern braucht man, um die Zahl n als b-er-Zahl (d.h.
3      // als Zahl im Zahlensystem mit der Basis b, z.B. als 2-er-Zahl oder
4      // als 8-er-Zahl oder als 10-er-Zahl oder als 16-er-Zahl etc.)
5      // darzustellen? Diese Funktion liefert die Antwort.
6      // Zur Darstellung der Zahl 0 braucht man (in jedem Zahlensystem)
7      // *eine* Ziffer (naemlich eine 0). Alle anderen Zahlen denkt man
8      // sich natuerlich ohne fuehrende Nullen dargestellt.
9      // Diese Funktion liefert 0 als Ergebnis, wenn die Basiszahl b nicht
10     // zwischen 2 und 16 (einschliesslich) liegt.
11
12     // anzZiffern( 0, 10) ist gleich 1
13     // anzZiffern( 0, 2) ist gleich 1
14     // anzZiffern( 8, 10) ist gleich 1
15     // anzZiffern( 8, 2) ist gleich 4
16     // anzZiffern( 716, 10) ist gleich 3
17     // anzZiffern(-716, 10) ist gleich 3
18     // anzZiffern(0055, 10) ist gleich 2
19     // anzZiffern(-055, 10) ist gleich 2
20     // anzZiffern(0xA0, 10) ist gleich 3
21     // anzZiffern(0xA0, 16) ist gleich 2
```

Aufgabe 2 (15 Punkte): Betrachten Sie die folgende Definition einer Klasse:

```
1 class Name {
2     static int anz;
3 public:
4     static int getAnz();
5     Name(string name) : name(name), NR(anz++) {};
6 private:
7     int const NR;
8     string name;
9 public:
10    int getNR();
11    string getName();
12 }; // class Name
```

Definieren Sie die Elemente der Klasse Name, die (in der Klassendefinition) noch nicht definiert wurden. Die get-Methoden sollen den Wert des entsprechenden Attributs als Ergebnis liefern (z.B. soll die Methode getAnz den Wert des Attributs anz als Ergebnis liefern, etc.). Jedes Objekt der Klasse Name enthält offenbar ein Attribut NR. In dem Name-Objekt, welches als erstes erzeugt wird, soll das Attribut NR den Wert 1 haben.

Aufgabe 3 (15 Punkte): Stellen Sie sich ein Programm vor, welches aus der Klasse Name (siehe vorige Aufgabe) und der folgenden main-Funktion besteht

```
1 int main() {
2     Name ob1("Anna Blume");
3     Name ob2("Otto Meyer");
4     ...
5     return 0;
6 } // main
```

Stellen Sie sich vor, dass der Ausführer die main-Funktion bis zur Zeile 4 ausgeführt hat. Wieviele **Module** existieren in diesem Moment? Zeichnen Sie jeden dieser Module als rechteckigen Kasten, in dem die Namen der Elemente stehen, die der Modul enthält. Schreiben Sie den Namen des Moduls über den Kasten, etwa so:

```
Modul Otto:
+-----+
| summe  |
| gibAus |
| x      |
+-----+
```

Die Namen in diesem Beispiel-Modul (Otto, summe, gibAus und x) sind frei erfunden und haben keinen Bezug zu dieser Aufgabe.

Aufgabe 4 (15 Punkte): Schreiben Sie (in C++) ein Unterprogramm entsprechend der folgenden Deklaration:

```
1 int  anzKleineBuchstaben(string * sr, int const LEN);
2     // Verlaesst sich darauf, dass sr die Anfangsadresse einer Reihung
3     // der Laenge LEN ist.
4     // Ermittelt die Anzahl der kleinen Buchstaben ('a' bis 'z'), die
5     // in den Komponenten der Reihung sr vorkommen, und liefert sie
6     // als Ergebnis.
7     // Diese Funktion kann insbesondere auch Reihungen der Laenge 0
8     // und Reihungen, die string-Objekte der Laenge 0 enthalten,
9     // korrekt bearbeiten.
```

Aufgabe 5 (15 Punkte): Beantworten Sie die folgenden Fragen **kurz**, aber **genau** (möglichst so ähnlich, wie es in den Wiederholungen zu Beginn der Vorlesungen geschah):

1. Kurzdefinition des Begriffs **Modul**?
2. Kurzdefinition des Begriffs **Klasse**?
3. Kurzdefinition des Begriffs **binärer Baum**?
4. Was muss gelten, damit ein binärer Baum **sortiert** ist?
5. Geben Sie die Namen aller **Adresstypen** an, die es zum Typ string * gibt.
6. Welche Elemente einer Klasse muss man **innerhalb** der Klassendefinition **definieren**?
7. Welche Elemente einer Klasse muss man **ausserhalb** der Klassendefinition **definieren**?
8. Geben Sie die physikalische und die logische Länge des folgenden C-Strings an:

```
10 char cs[20] = "Hallo!";
```

Aufgabe 6 (15 Punkte): Stellen Sie die folgende Variable tab als **Boje** dar:

```
11 string * tab[3] = {new string("ABC"), new string("DE"), new string("")};
```

Lösung 1: Die Funktion anzZiffern:

```

1 int anzZiffern(int n, int b) {
2
3     if (b < 2 || 16 < b) return 0;
4     int erg = 0;
5     while (true) {
6         erg++;
7         n /= b;
8         if (n==0) break;
9     }
10    return erg;
11 } // anzZiffern

```

Lösung 2: Die Definitionen der noch nicht definierten Elemente der Klasse Name:

```

1 int    Name::anz = 1;
2
3 int    Name::getAnz () {return anz;}
4
5 int    Name::getNR  () {return NR;}
6 string Name::getName() {return name;}

```

Lösung 3: Anzahl und Inhalte der Module:

Modul Name:	Modul ob1:	Modul ob2:
<pre> +-----+ anz getAnz Name +-----+ </pre>	<pre> +-----+ NR name getNR getName +-----+ </pre>	<pre> +-----+ NR name getNR getName +-----+ </pre>

Lösung 4: Die Funktion anzKleineBuchstaben:

```

12 int anzKleineBuchstaben(string * sr, int const LEN) {
13
14     int anz = 0;
15     for (int i=0; i<LEN; i++) {
16         for (int j=sr[i].size()-1; j>=0; j--) {
17             char c = sr[i].at(j); // Entweder so
18             // char c = sr[i][j]; // oder so!
19             if ('a'<=c && c<='z') anz++;
20         }
21     }
22     return anz;
23 } // anzKleineBuchstaben

```

Lösung 5: Ausgabe des Programms:

5.1. Ein Modul ist ein Behälter für Variablen, Unterprogramme, Typen, Module etc., der aus mindestens zwei Teilen besteht, einem öffentlichen (ungeschützten) und einem privaten (geschützten Teil). Von Stellen ausserhalb des Moduls kann man nur auf die Elemente im öffentlichen Teil zugreifen, aber nicht auf die Elemente im privaten Teil.

5.2. Eine Klasse ist ein Modul und ein Bauplan für Module.

5.3. Ein binärer Baum ist entweder leer oder er besteht aus einem Knoten, an dem zwei Bäume hängen (ein rechter Unterbaum und ein linker Unterbaum).

5.4. Ein binärer Baum ist sortiert, wenn für jeden Knoten k und seinen Schlüssel s gilt:

Alle Schlüssel im linken Unterbaum von k sind kleiner als s und

alle Schlüssel im rechten Unterbaum von k sind grösser als s.

5.5. Zum Typ `string *` („Adresse von string“) gibt es die folgende 4 Adresstypen:

```

string * const * const
string * const *
string *         * const
string *         *

```

5.6. Objektattribute muss man innerhalb der Klassendefinition definieren.

5.7. Klassenattribute muss man ausserhalb der Klassendefinition definieren.

5.8. Der C-String namens cs hat die physikalische Länge 20 und die logische Länge 6.

Lösung 6 (15 Punkte):

```

| tab | --<10>-- | +-----+
|      |         | |<10>--[<50>]--+--+"ABC" | |
|      |         | |         |         |
|      |         | |         |         |
|      |         | |<14>--[<60>]--+--+"DE"  |
|      |         | |         |         |
|      |         | |         |         |
|      |         | |<18>--[<70>]--+--+" "    |
|      |         | |         |         |

```