



Grundzüge der objektorientierten Programmierung

WS2001/2002 – Übungsblatt 3

Abgabetermin: 19. November 2001

Aufgabe 1. Überladung von Funktionsnamen, 5 Punkte

Schreiben Sie eine Familie von Funktionen, die alle Elemente in einem Array aufsummieren:

```
int sum(const int a[], int length);  
double sum(const double a[], int length);  
char *sum(const char *a[], int length);
```

Die ersten beiden Versionen addieren die numerischen Werte der Arrays. Die Version für `char *` soll die Strings im Array zu einem String aneinanderhängen (Speicher einmal mit `new` besorgen).

Testen Sie alle Versionen Ihrer überladenen Funktionen mit einem kurzen Hauptprogramm. Versuchen Sie auch, `sum` auf ein `float`-Array anzuwenden. Könnte man zusätzlich

```
double sum(const int a[], int length);
```

überladen?

Aufgabe 2. `math.h/cmath`, 5 Punkte

Sehen Sie sich die Datei `/usr/include/math.h` an. (Auf LINUX-Systemen betrachten Sie zusätzlich bitte auch die Datei `/usr/include/bits/mathcalls.h`.) Erstellen Sie eine Liste der dort bereitgestellten mathematischen Konstanten jeweils mit kurzer Erläuterung:

```
M_E      Eulersche Zahl e  
...  
M_PI_2   Pi/2  
...
```

Erstellen Sie eine ähnliche Liste der vorhandenen transzendenten Funktionen:

```
acos    arcus cosinus
...
log10   Logarithmus zur Basis 10
...
```

Die Bedeutung können Sie mittels `man acos,...` vom Hilfesystem des Unix-Betriebssystems ermitteln.

Schreiben Sie ein Programm, daß alle diese Konstanten und Funktionen mindesten einmal benutzt.