

# Kompaktkurs Einführung in die Programmierung

## Übungsblatt 6: Zeiger

### 1. Zeiger und Referenzen (1)

Was ist nach Ausführung der folgenden Anweisungen jeweils der Inhalt der Variablen  $x$  und  $y$ ?

a) Referenzieren und Dereferenzieren:

```
int x = 3, y = 4;
int *p;
p = &x;
*p = *p + y;
```

b) Zeiger zuweisen:

```
int x = 3, y = 4;
int *p1, *p2;
p1 = &x;
p2 = &y;
*p1 = *p1 + y;
p1 = p2;
*p1 = *p1 + y;
```

c) Zeiger auf Zeiger:

```
int x = 3, y = 4;
int *p1 = &x, *p2 = &y;
int **pp = &p1;
(**pp)--;
*pp = p2;
(**pp)++;
```

d) Variablenparameter:

```
int x = 3, y = 4;
zz(&x, &y);
```

mit der Funktion `zz`:

```
void zz(int *a, int *b) {
    int c = *a;
    *a = *b;
    *b = c;
}
```

e) char Zeiger:

```
char x = 'a', y = 'k';
char *ppc = &x;
*ppc += 3;
y += 2;
```

f) Zeiger und Zeiger:

```
float x = 1.5, y = -0.5;
float *z = &x;
z = &y;
float **ppf = &z;
foo(ppf);
```

mit der Funktion `foo`:

```
void foo(float **pf) {
    **pf *= 4.0;
}
```

## 2. Zeiger und Referenzen (2)

Gegeben sei folgendes Programm

```
int main()
{
    int a = 2, b = 3, *pi = &a, **ppi, ***pppi = &ppi;
    char c='a', *pc;
}
```

Welche folgender Zuweisungen sind zulässig und welche sind wegen Typ-Konflikten bedenklich bzw. nicht möglich/sinnvoll? Begründen Sie ihre Antworten!

- |                                 |                               |
|---------------------------------|-------------------------------|
| a) <code>pi = &amp;b;</code>    | d) <code>pi = &amp;c;</code>  |
| b) <code>pc = pi;</code>        | e) <code>*pi = *pc;</code>    |
| c) <code>**pi = &amp;pi;</code> | f) <code>***pi = pppi;</code> |

## 3. Variablenparameter (Satz von Bezout)

Zu zwei ganzen Zahlen  $a$  und  $b$  gibt es zwei ganze Zahlen  $n$  und  $m$ , so dass  $ggT(a, b) = n \cdot a + m \cdot b$ . Bei der Durchführung des euklidischen Algorithmus bekommt man solche Zahlen  $n$  und  $m$  beinahe geschenkt, daher soll die  $ggT$ -Funktion von Blatt 5 um die Berechnung dieser Werte erweitert werden. Da eine Funktion immer nur eine Größe als Ergebnis haben kann, arbeiten wir mit Variablenparametern, so dass die Funktion die berechneten Werte für den  $ggT$ ,  $n$  und  $m$  in Variablen der aufrufenden Funktion speichern kann. Gesucht ist also eine Funktion folgender Signatur:

```
void ggT(int a, int b, int *ggT, int *n, int *m)
```

Nach Aufruf von `ggT(a, b, &ggT, &n, &m)` sollen in der Variablen `ggT` der  $ggT$  von  $a$  und  $b$ , und in den Variablen `n` und `m` die oben spezifizierten Zahlen  $n$  und  $m$  enthalten sein.

*Hinweis:* Die (rekursive!) Berechnungsvorschrift für die Zahlen  $n$  und  $m$  erschließt sich aus der Identität

$$b \% a == b - (b/a) * a \quad (\text{für } a \neq 0).$$

Die Arbeitsweise kann wieder über ein Programmprotokoll erkannt werden. Hier wird wie bei Blatt 5 über einen entsprechenden Zusatzparameter nach Rekursionstiefe eingerückt:

```
| ggT(72, 30) aufgerufen
| | ggT(30, 72) aufgerufen
| | | ggT(12, 30) aufgerufen
| | | | ggT(6, 12) aufgerufen
| | | | Der ggT ist 6
| | | | 6 = 1*6 + 0*12
| | | | 6 = -2*12 + 1*30
| | | 6 = 5*30 + -2*72
| | 6 = -2*72 + 5*30
```