

## Übungen zu Graphikprogrammierung in C++

### Aufgabe 9 (H) Zeiger (Pointer) und Felder (Arrays)

Ein Zeiger vom Typ  $T$  wird durch  $T^*$  ausgedrückt. Ein einfaches Beispiel hierfür ist:

```
int no = 23;
int* p = &no; // p holds the address of no
int no2 = *p; // no2 == 23
```

Dieses einfache Programm verdeutlicht den Unterschied zwischen einem Objekt von einem bestimmten Typ  $T$  und einem Pointer auf das Objekt. Der Pointer gibt die Speicheradresse an, in der das Objekt dieses Datentyps liegt.

Mit Hilfe von Pointern soll die Funktion

```
bool sortList(int* list, int length){ ... }
```

geschrieben werden, die eine Liste von Integerwerten aufsteigend sortiert. Der einfachste Algorithmus hierfür ist der BubbleSort. Er vergleicht jeweils zwei benachbarte Listenelemente. Falls der linke Wert kleiner ist, so muss nichts gemacht werden. Falls der linke Wert größer ist, so müssen die beiden Werte vertauscht werden.

Das Programm soll mit Beispieldaten getestet werden. Eine Liste mit 7 Elementen kann z.B. durch

```
int listToSort [] = {3,7,2,4,1,9,5};
```

erstellt werden. Nach dem Aufruf der Funktion `sortList` soll genau diese Liste sortiert sein.

### Aufgabe 10 (H) Einfache Bildmanipulation

Ein Bild wird im Folgenden als eindimensionales Feld von Integerwerten dargestellt. Zusätzlich werden Variablen für die Dimensionen des Bildes benötigt. Jeder Wert zwischen 0 und 255 drückt die Intensität des Grauwertes aus, wobei 0 schwarz und 255 weiß ist. Z.B. stellt

```
int width = 5; // Breite des Bildes
int height = 4; // Höhe des Bildes
int* image = {2 , 7 , 8 , 3 , 5 ,
              6 , 100, 120, 121, 4 ,
              222, 212, 122, 120, 34,
              56 , 78 , 67 , 89 , 101};
```

ein kleines 5x4 Pixel großes Bild dar.

In zwei Funktionen soll das Bild horizontal und vertikal geflippt werden.

## **Aufgabe 11 (H) Bildmanipulation am echten Beispiel**

Nachdem das Programm mit dem einfachen Zahlenbeispiel getestet wurde, sollen die beiden implementierten Funktionen auf echte Bitmaps angewendet werden.

Im SVN ist unter `solution/Aufgabe11` ein Beispielprogramm (`Aufgabe11.cpp`), das auf der Basis von `solution/imagelib` ein `.bmp` Bild einliest und dieses wieder speichert. Der Inhalt von `solution` soll in das jeweilige Team Verzeichnis kopiert werden. Es ist sowohl ein Makefile als auch eine VS solution vorhanden, womit das Beispielprogramm und die lib erzeugt werden können.

Die Datei `Bitmap.cpp` muss um die flip-Funktionen `flipH()` sowie `flipV()` erweitert werden. Dabei ist zu beachten, dass die Bild-Daten in der Variablen `*m_data` gespeichert werden, welche manipuliert werden muss. Der Datentyp eines Bitmaps ist `byte`. Die Höhe und Breite des Bildes stehen in `m_dim`. Dabei ist `m_dim[0]` die Breite und `m_dim[1]` die Höhe des Bildes.