

## 2. Übungsblatt

(Abgabe: 1. - 5. November 1999)

1. **Aufgabe:** KORREKTHEIT (Punkte: 3)

Korrigieren Sie die Programme `loop1.C`, `loop2.C` und `loop3.C`.

2. **Aufgabe:** COLOR KLASSE (Punkte: 3)

Entwerfen und Implementieren Sie eine Klasse `Color` für RGB-Farbwerte. Es soll zumindest Memberfunktionen `red()`, `green()` und `blue()` geben, die Rot-, Grün- bzw. Blauanteil liefern. Schreiben Sie ferner eine Funktion, die als Argument ein Objekt vom Typ `Color` nimmt und eine der `glColor3--()` Funktionen aus OpenGL aufruft (-- steht hier für `b`, `d`, `f`, `i`, `s`, `ui` oder `us`). Setzen Sie die Funktion im Programm `circle_main8.C` ein.

3. **Aufgabe:** SEGMENT KLASSE (Punkte: 4)

Entwerfen und Implementieren Sie eine Klasse `Segment` für Strecken in der Ebene. Eine Strecke soll aus ihren Eckpunkten konstruiert werden können. Fügen Sie eine Memberfunktion `draw()` hinzu, mit der Sie eine Strecke in ein OpenGL Fenster zeichnen können. Schreiben Sie ein Programm `segment_main.C`, das Koordinaten zweier Punkte einliest und die Strecke zwischen den Punkten zeichnet.

Hinweis: Seien `x1`, `y1`, `x2`, `y2` Variablen vom Typ `float`; Das Codefragment

```
glBegin( GL_LINES);
  glVertex2f( x1, y1);
  glVertex2f( x2, y2);
glEnd();
```

zeichnet die Strecke zwischen den Punkten  $(x_1, y_1)$  und  $(x_2, y_2)$ .

4. **Aufgabe:** LEERZEILEN ENTFERNEN (Punkte: 5)

Schreiben Sie ein Programm, das Mehrfachvorkommen von Leerzeilen durch eine einzelne Leerzeile ersetzt. Das Programm soll stets nach Standardausgabe schreiben. Wenn beim Programmaufruf ein Argument mitgegeben wird, so soll das Argument als Filename interpretiert werden. Ihr Programm soll dann das File mit diesem Namen lesen. Gibt es kein Argument, so soll das Programm von Standardeingabe lesen. Orientieren Sie sich am Programm `strip_of_comments.C`.

Hinweis: Mit der Compileroption `-E` können Sie sich anschauen, was der C Präprozessor aus Ihren Files macht. Das Compiler schreibt das Resultat auf Standardausgabe. Die Ausgabe ist recht unleserlich, da es viel Leerraum gibt. Sie können nun die Compilerausgabe durch Ihr Programm pipen und dann anschauen (z.B. `g++ -E file.C | strip_of_empty_lines | more`).

5. **Aufgabe:** LÖSEN QUADRATISCHER GLEICHUNGEN (Punkte: 5)

Schreiben sie eine Funktion `void roots_of_quadratic_eq(const Number& a, const Number& b, const Number& c, complex< Number>& root1, complex< Number>& root2)`, die die Nullstellen der quadratischen Gleichung  $ax^2 + bx + c = 0$  berechnet und über die Parameter `root1` und `root2` zurückliefert. Implementieren sie die Funktion für die Zahltypen `float`, `double` und `leda_real`. Beim Zahltyp `leda_real` müssen Sie `leda_real_complex.o` zum Wurzelziehen aus komplexen Zahlen hinzulinken.

Zur Erinnerung: Die Nullstellen der Gleichung  $ax^2 + bx + c = 0$  sind von der Form  $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ .