FACHRICHTUNG 6.1
MATHEMATIK

Dr. M. Riplinger
Dipl. Math. Y. Johann

UNIVERSITÄT DES SAARLANDES

6. Übung zu „Modellierung/Programmierung“

Wintersemester 2012/2013

1. Aufgabe Arbeiten mit Dateien — 6 Punkte

a) Schreiben Sie eine Funktion `erzeugeDaten()`, die eine Wertetabelle für die Normalparabel im Intervall $[a, b]$ mit N Stützstellen erzeugt. Als Eingabeparameter erwartet die Funktion die Intervallgrenzen a und b, die Anzahl der Knoten N und Pointer auf dynamische Felder `x` und `y`, in welche die entsprechenden Werte eingetragen werden sollen, d.h. es soll das Prinzip *Call by reference* verwendet werden. Der Rückgabewert ist vom Typ `void`.
Hinweis: Es sei $h = \frac{b-a}{N}$ die Schrittweite zwischen den einzelnen Knotenpunkten $x_i = a + i \cdot h$, $i = 0, \ldots, N$.

b) Schreiben Sie eine Funktion `schreibeDaten()`, welche die erzeugte Wertetabelle in einer **Text**datei speichert. Die erste Spalte der Tabelle enthält die Stützstellen, die zweite Spalte die dazugehörigen Funktionswerte. Für die Werte soll in jeder Spalte Platz für mindestens 6 Zeichen bei einer Genauigkeit von 2 Nachkommastelle reserviert werden. Die Funktion erwartet als Eingabeparameter den File-Pointer, die Anzahl der Stützstellen und die beiden Felder. Sollten Fehler beim Öffnen auftreten, soll dies mit einer entsprechenden Fehlermeldung auf dem Bildschirm angezeigt werden. Der Rückgabewert ist vom Typ `void`.

c) Schreiben Sie ein Hauptprogramm, in dem Sie zunächst die Wertetabelle über `erzeugeDaten()` erstellen und anschließend mit der Funktion `schreibeDaten()` in der Datei `square.dat` abspeichern. Am Ende des Hauptprogramms soll die Datei `square.dat` wieder geschlossen werden.

2. Aufgabe Methode der kleinsten Fehlerquadrate — 6 Punkte

Ein Verfahren zur Ausgleichsrechnung ist die Methode der kleinsten Fehlerquadrate, die dazu verwendet wird, gemessene Datenpukte durch eine stetige Funktion darzustellen, deren Verlauf man analysieren kann. Hierbei sollen die Punkte möglichst nahe an der Kurve liegen. Dies wird erreicht, indem der quadratische Abstand zwischen der Kurve und den Messdaten, d. h. der Fehler, minimiert wird.
Im einfachsten Fall kann man den Verlauf der Daten durch eine Gerade $y(t) = at + b$ approximieren. Hierbei sind die Steigung a und der y-Achsenabschnitt b zu bestimmen. Die Parameter a und b lassen sich in diesem Fall explizit angeben. Es gilt

$$a = \frac{\sum_{i=1}^{N} (t_i - \bar{t})(y_i - \bar{y})}{\sum_{i=1}^{N} (t_i - \bar{t})^2} \quad \text{und} \quad b = \bar{y} - a\bar{t}.$$

Dabei sind $\bar{t}$ und $\bar{y}$ die arithmetischen Mittel von t bzw. y definiert, d.h. es gilt bspw. für $\bar{t}$:

$$\bar{t} = \frac{1}{N} \sum_{i=1}^{N} t_i.$$

Schreiben Sie ein Programm, das die in der Datei `leastsquare.dat` enthaltenen Daten (s. Homepage zur Vorlesung) mit Hilfe des oben beschriebenen Ansatzes durch eine Gerade approximiert. Die Datei enthält die Daten einer Studie, in der Körpergröße und Ruhepuls von 50 verschiedenen Patienten miteinander verglichen werden. Geben Sie die Gleichung der Geraden sowie die Fehler $err_i = |y - y_i|$ am Bildschirm aus.

3. Aufgabe Makros 6 Punkte

a) Schreiben Sie Makros `REAL`, `IMAG`, `SETZE_REAL` und `SETZE_IMAG` mit denen Sie auf Real- bzw. Imaginärteil komplexer Zahlen lesend bzw. schreibend zugreifen können. Die komplexen Zahlen sollen analog zur 9.Vorlesung (12.12.12) als Struktur definiert werden.

b) Schreiben Sie ein Makro zur Ausgabe komplexer Zahlen am Bildschirm.
Hinweis: Aufgabe 4.

c) Schreiben Sie die Funktionen `newComplexPolar()` und `complexProduct()` aus der Vorlesung vom 12.12.12 als Makros, wobei jedoch die Struktur selbst, statt ein Zeiger auf diese durch die Makros zurückgegeben werden soll.

d) Testen Sie Ihre Makros in einem Hauptprogramm, indem Sie zunächst eine komplexe Zahl definieren und diese überschreiben, eine zweite mit Hilfe des Makros `NEW_COMPLEX_POLAR` defnieren und anschliessend das Produkt dieser beiden Zahlen berechnen. Geben Sie die komplexen Zahlen anschliessend am Bildschirm aus.

4. Aufgabe Makros 2 Punkte

Welches der folgenden Makros liefert beim Aufruf

```
PRINT_COMPLEX("Zahl %d: ", z1, 1);
```

die Ausgabe

```
Zahl 1:  1.000000-2.000000i,
```

wenn z1 eine komplexe Zahl ist mit Realteil 1.0 und Imaginärteil -2.0?

a)
```
#define PRINT_COMPLEX(_FMT_STR, Z, ...) \
 printf(_FMT_STR "% lf%+lfi\n", REAL((Z)), IMAG((Z)), ##__VA_ARGS__)
```

b)
```
#define PRINT_COMPLEX(_FMT_STR, Z, ...) \
 printf(_FMT_STR "% lf%+lfi\n", #__VA_ARGS__, REAL((Z)), IMAG((Z)))
```

c)
```
#define PRINT_COMPLEX(_FMT_STR, Z, ...) \
 printf(_FMT_STR "% lf%+lfi\n", ##__VA_ARGS__, REAL((Z)), IMAG((Z)))
```

d)
```
#define PRINT_COMPLEX(_FMT_STR, Z, ...) \
 printf("% lf%+lfi\n", ##__VA_ARGS__, REAL((Z)), IMAG((Z)))
```

Abgabe bis zum 30.01.2013, 19.00 Uhr per E-Mail an Ihren Bremser.

Hinweis: Da die Anmeldung zur Klausur via Hispos nicht für Studenten aller Studiengänge funktioniert, melden Sie sich auf der Vorlesungs-Homepage zur Klausur an (gegebenfalls zusätzlich zu Hispos). Die Anmeldung auf der Homepage ist **relevant**!