Prof. Dr. Franz Schweiggert — Sektion Angewandte Informationsverarbeitung — 11. Januar 1999
Dr. Andreas Borchert Blatt 10
Ingo Melzer

Allgemeine Informatik I (WS 1998/99)

Abgabetermin 20. Januar 1999

12 Kantige Kurven (15 Punkte)

Erstellen Sie eine kleine Anwendung zum Plotten von Funktionen, die mit einfacher auf ASCII-Zeichen basierender Graphik auskommen darf. Auf die Ausgabe von Koordinatensystemen kann verzichtet werden. So könnte ein Aufruf aussehen:

```
oberon$ fplot -h 25 -w 77 -x '-pi' 'pi' 'sin(x)'
                                                  XXXXXXXXX
                                                XXXXXXXXXXXXX
                                               XXXXXXXXXXXXXXXX
                                              XXXXXXXXXXXXXXXXXX
                                             XXXXXXXXXXXXXXXXXXXX
                                            XXXXXXXXXXXXXXXXXXXXXX
                                           XXXXXXXXXXXXXXXXXXXXXXXX
                                          XXXXXXXXXXXXXXXXXXXXXXXXXX
                                         XXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X                                    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XX                                  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX                                XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXX                              XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXX                            XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXX                          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXX                        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXX                      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXX                  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXX                XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXX            XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXX        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
oberon$
```

Die zu unterstützenden Kommandozeilenparameter im einzelnen:

-d *xdelta* Abstand zwischen den einzelnen Punkten auf der X-Achse. Dies wird voreinstellungsgemäß in Abhängigkeit des Intervalls und der gewünschten Weite berechnet.

-h *höhe* spezifiziert die zur Verfügung stehende Höhe. Die Voreinstellung ist bei 23.

-v *varname* Der Variablenname, der beim Funktionsausdruck verwendet wird. Dies ist normalerweise "x".

-w *weite* spezifiziert die zur Verfügung stehende Weite. Die Voreinstellung liegt bei 80. Diese Angabe wird hinfällig, wenn *xdelta* angegeben wird.

-x *min max* gibt das darzustellende Intervall auf der X-Achse an. Die Voreinstellung liegt bei $[-1..1]$. Die Angabe von Ausdrücken ist dabei zulässig (siehe obiges Beispiel).

Als letztes folgt auf der Kommandozeile die darzustellende Funktion.

Hinweise:

Zur Bearbeitung der Kommandozeilenparameter sollte das Modul *Arguments* verwendet werden.
Für die Bewertung von Funktionen empfiehlt sich das Modul *Functions*. Mit *ParseFunction* kann eine Zeichenkette parsiert und als bewertbare Funktion abgelegt werden, mit *SetFuncParam* ist es möglich, einem der Parameter in der Funktion einen Wert zuzuweisen (hier insbesondere für "x") und mit *EvalFunction* kann dann der entsprechende Wert ermittelt werden. Wenn Sie zusätzlich noch das Modul *StdFuncs* importieren, stehen Ihnen zahlreiche Standardfunktionen wie z.B. $sin(x)$ oder $sqrt(x)$ zur Verfügung nebst den Konstanten pi und e. Sehen Sie sich bitte dazu die zugehörigen Manualseiten an. Sie finden dort nicht nur die zulässige Eingabesyntax, sondern auch ein kleines Anwendungsbeispiel.
Die Aufgabe besteht im Grunde genommen aus drei einzelnen Teilen, bei denen es sich auch empfiehlt, sie hintereinander zu testen. Wenn Sie die Argumentverarbeitung fertig haben, können Sie testweise alle Argumente wieder ausgeben. Teilen Sie die Plotfunktionalität in zwei Bereiche auf: eine Prozedur, die eine Folge von Funktionswerten generiert, und eine Prozedur, die jeweils ein Funktionswert entgegennimmt und zur späteren Aufbereitung abspeichert. Hier können Sie testweise zunächst alle Funktionswerte ausdrucken und diese z.B. mit *gnuplot* betrachten. Erst wenn dies funktioniert, macht es Sinn, sich mit dem eigentlichen Plotten auseinanderzusetzen. Hierfür werden Sie ein Array benötigen, da Sie nicht im voraus wissen, welches Intervall auf der Y-Achse benötigt wird.