Zufallsexperimente mit dem TI-92

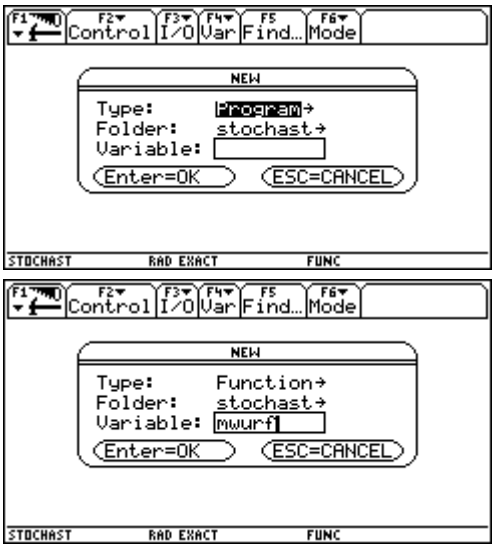
In der Einführungsphase der Stochastik werden in der Regel Zufallsexperimente wie Münzwurf oder Würfeln durchgeführt. Wie der TI-92 dabei eingesetzt werden kann, zeigen die folgenden beiden Beispiele.

1. Münzwurf als Funktion

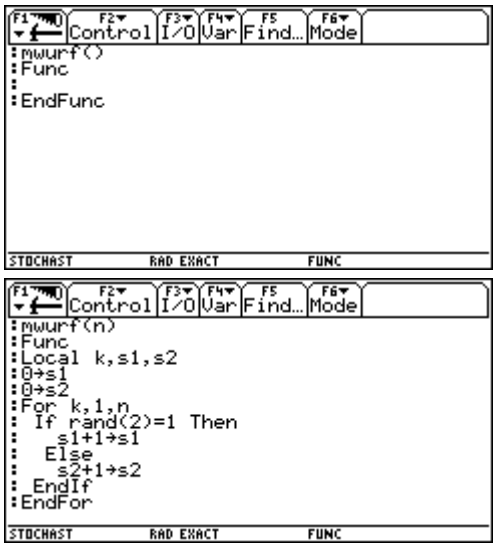
Es soll eine Funktion mwurf(n) erstellt werden, die für n Würfe einer Münze die Anzahlen für Kopf und Zahl als Liste (Vektor) als Ergebnis liefert.

Zuerst legen wir uns ein Verzeichnis *stochast* an. Mit 2 ° kommt man in den Datei-Manager (vergleichbar mit dem Explorer unter Windows). Dort wählt man *f* (Manage)-5:Create Folder. In dem erscheinenden Fenster gibt man *stochast* ein und bestätigt mit ¸ . Anschließend rufen wir mit der Taste 3 das Mode-Fenster auf. Mit dem Cursor-Rad gehen wir auf die Zeile Current Folder. Mit dem Cursor-Rad nach rechts (B) machen wir das Pulldown-Menü auf und wählen *stochast* aus. Anschließend ¸ für auswählen und noch mal ¸ für save. Wir befinden uns nun im Ordner *stochast* und können mit der Programmierung beginnen.

Zuerst öffnen wir mit O - 7:Program Editor – 3:New folgendes Fenster:



Dann mit B das Pulldown-Menü öffnen und 2:Function auswählen (¸), dann bei Variable mwurf eingeben (¸). Es erscheint der Funktions-Editor. Dort geben wir den Programmtext ein.



Der Rahmen der Funktion wird bereits vorgegeben. Der vollständige Programmtext sieht dann so aus:

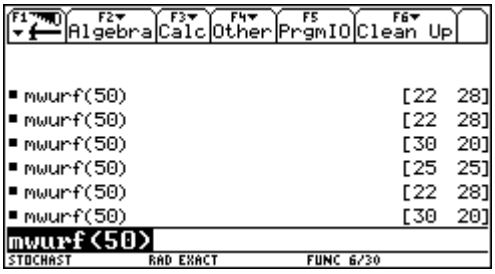
```
mwurf(n)
Func
Local k,s1,s2
0→s1
0→s2
For k,1,n
 If rand(2)=1 Then
  s1+1→s1
 Else
  s2+1→s2
 EndIf
EndFor
[s1,s2]
EndFunc
```

Die einzelnen Programmschritte erklären sich wie folgt:

- die Variablen k, s1 und s2 werden nur innerhalb des Programms benutzt, sie können deshalb als lokal vereinbart werden.
- die Variablen s1 und s2 zählen die Ergebnisse Kopf bzw. Zahl, die durch die Werte 1 und 2 dargestellt werden.
- rand(2) erzeugt eine Zufallszahl aus der Menge {1; 2}.
- den Zuweisungspfeil erhält man mit § .

- [s1,s2] gibt das Ergebnis der Funktion an.

Mehrfaches Aufrufen von mwurf(50) könnte dann so aussehen:



2. Grafische Darstellung eines Würfelexperimentes

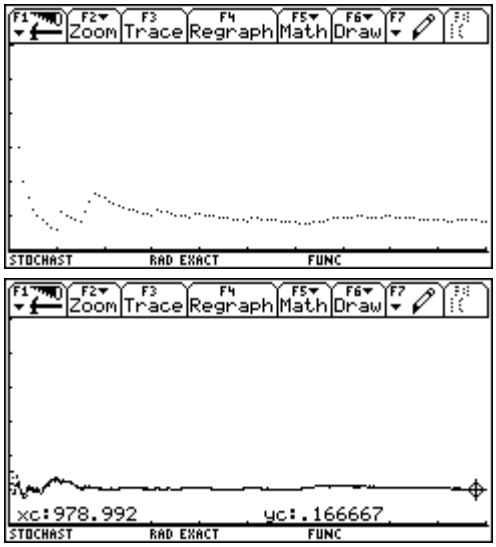
Um das empirische Gesetz der großen Zahlen deutlich zu machen bietet sich eine Simulation an. Das folgende Programm führt ein n-maliges Würfeln durch und stellt nach jedem Wurf die relative Häufigkeit der 6 dar. In den Programm-Editor kommt man mit O - 7:Program Editor–3:New. Diesmal lässt man als Type Program und gibt als Variable dice ein.

```
dice(n)
Prgm
Local k,anzahl,sum6
ClrDraw
0→xmin:n→xmax:n/10→xscl
0→ymin:1→ymax:1/6→yscl
0→anzahl:0→sum6
For k,1,n
  If rand(6)=6 Then
    sum6+1→sum6
  EndIf
  anzahl+1»anzahl
  PtOn anzahl,sum6/anzahl
EndFor
EndPrgm
```

Die neuen Befehle bedeuten

- ClrDraw: löscht das Grafikfenster
- die folgenden beiden Zeilen legen das Koordinatensystem fest
- PtOn: setzt einen Punkt (Pixel) an die angegebenen Koordinaten

Aufrufe von dice(100) bzw. dice(1000) liefern folgende Bilder



Im zweiten Bild wurde nach Beendigung mit dem Cursor-Rad ein Fadenkreuz aufgerufen. Dies bewegt man an Ende des Graphen und kann die Koordinaten ablesen. Das Fadenkreuz lässt sich schneller bewegen, wenn man zusätzlich die 2 –Taste drückt. Die relative Häufigkeit stabilisiert sich um den Wert 1/6.

Es ist auch interessant, den ersten Bereich des ersten Graphen zu interpretieren. Man erkennt sehr schön, dass ím zweiten Wurf eine 6 kam. Die nächste 6 tritt erst im elften Wurf auf.