

## Vorbemerkungen

Erfahrungen zeigen, dass die Programmiermöglichkeiten des ClassPad im Unterricht kaum genutzt werden. Dabei bieten aus unserer Sicht viele Situationen die Gelegenheit, die Programmieroberfläche als Alternative zur Arbeit mit dem Statistikmenü zu verwenden. Zudem ist es in einem Leistungskurs angebracht, das Programm und seine Struktur selbst in den Mittelpunkt des Unterrichts zu rücken, damit Schülerinnen und Schüler die Bedeutung von Algorithmen verstehen. Mit Abstrichen ist dies bereits auch in der Sekundarstufe 1 möglich.

Beim Lösen von Gleichungen komplexerer Art kann das Programmieren das systematische Probieren ersetzen (Bsp. 6). Größere Bedeutung hat das Programmieren beim Simulieren von Zufallsprozessen. Zum einen können z. B. Fragestellungen zu Würfelexperimenten bearbeitet und die Ergebnisse der Simulation aus dem Programm heraus graphisch dargestellt werden (Bsp. 1). Zum anderen kann z. B. ein Näherungswert für  $\pi$  ermittelt werden, indem ein Quadrat mit eingelegtem Viertelkreis zufällig mit Punkten belegt wird (Bsp. 2).

Hilfreich kann ein Programm sein, um z. B. bei dem bekannten Geburtstagsproblem vor der Berechnung eine erste Vorstellung über die Wahrscheinlichkeit zu gewinnen, dass mindestens zwei Personen am gleichen Tag Geburtstag haben (Bsp. 3).

Im Bereich der Analysis können verschiedene Wachstumsprozesse durch Programme simuliert und graphisch dargestellt werden (Bsp. 4) oder einfach zu 2 vorgegebenen Punkten die Geradengleichung ermittelt werden (Bsp. 5).

Im Folgenden wird zunächst einmal die Programmieroberfläche des ClassPad erläutert. Daran anschließend werden die Programme zu den anfangs erwähnten Beispielen vorgestellt und kommentiert. Die Programme stehen auch als vcp-Dateien zur Verfügung.

## Die Programmieroberfläche des ClassPad

Die Einführung in die Programmieroberfläche des ClassPad wird an dem Beispiel eines unfairen Tetraeders gemacht. Die Wahrscheinlichkeit der Wurfresultate  $X$  beträgt:

$$P(X=1) = P(X=2) = P(X=3) = \frac{1}{6} \quad \text{und} \quad P(X=4) = \frac{1}{2}.$$

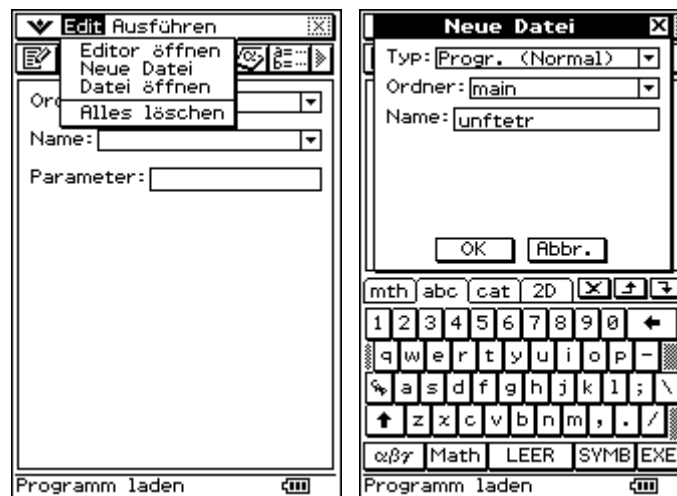
Öffnen Sie die Programmier-Anwendung, indem Sie im unteren Bereich des Hauptmenüs auf



tippen.



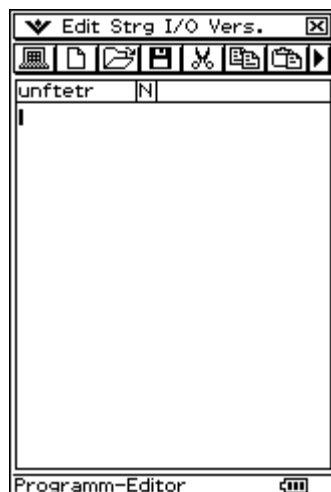
In der Menüleiste oben finden Sie das Edit-Menü. Dort wählen Sie den Menüpunkt *Neue Datei* aus.



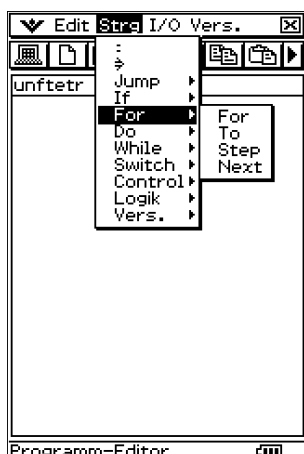
Über die erscheinende Eingabemaske kann eine neue Datei erzeugt werden. In das Fenster *Name* wird der Name des Programms eingegeben, beispielsweise *unftetr*. Die Programmnamen sind auf eine Länge von 8 Zeichen begrenzt.

Für die ersten Versuche sollte weder der Typ noch der Ordner verändert werden.

Durch die Bestätigung des Programmnamens gelangen Sie in die Programmier-Oberfläche.



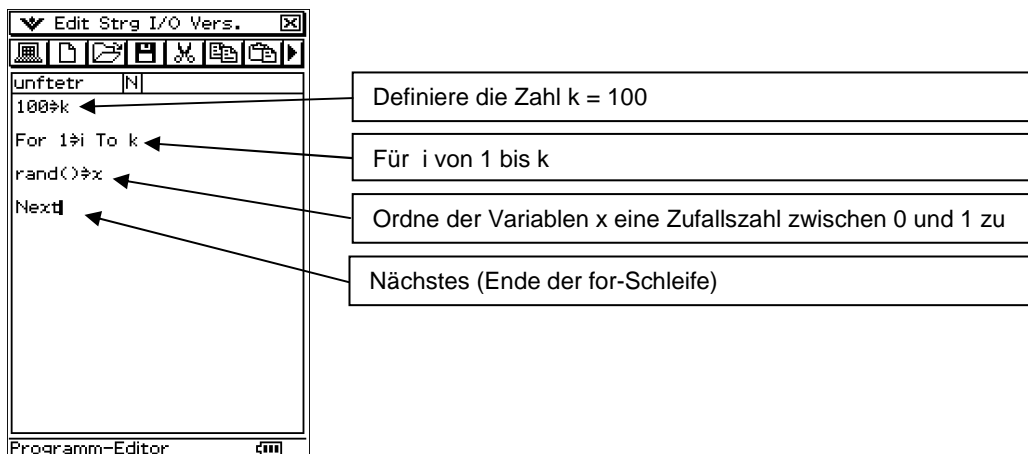
In der Kopfzeile dieser Programmieroberfläche existieren verschiedene Befehlsgruppen.



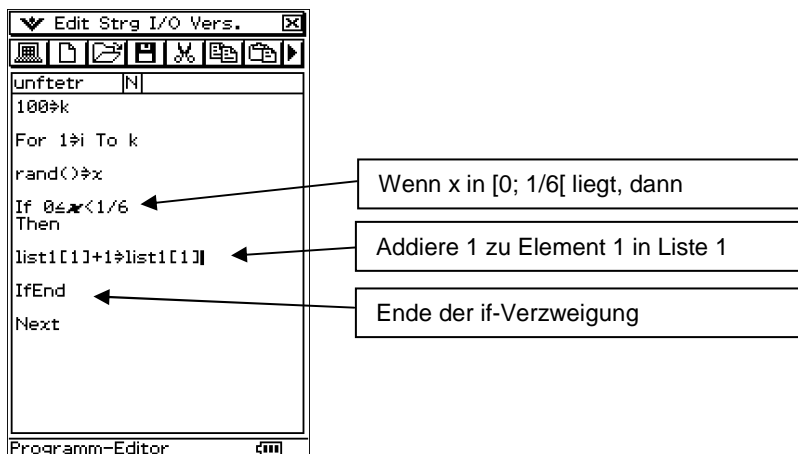
### Steuerung (Strg):

Unter dem Menüpunkt *Strg* finden sich logische Verzweigungen (z.B. *if*) und Schleifen (z.B. *for*), die für das Programmieren besonders wichtig sind. Sie werden im Weiteren noch am Beispiel erläutert.

Für das Tetraeder-Programm *unftetr* wird eine *for*-Schleife genutzt.



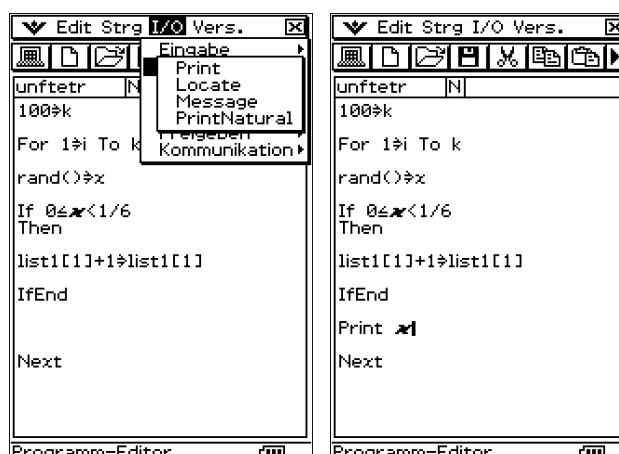
Für den unfairen Wurf benutzen wir eine *if*-Verzweigung für Zufallszahlen in verschiedenen Intervallen.



### Ein- und Ausgabe (Input/Output; I/O)

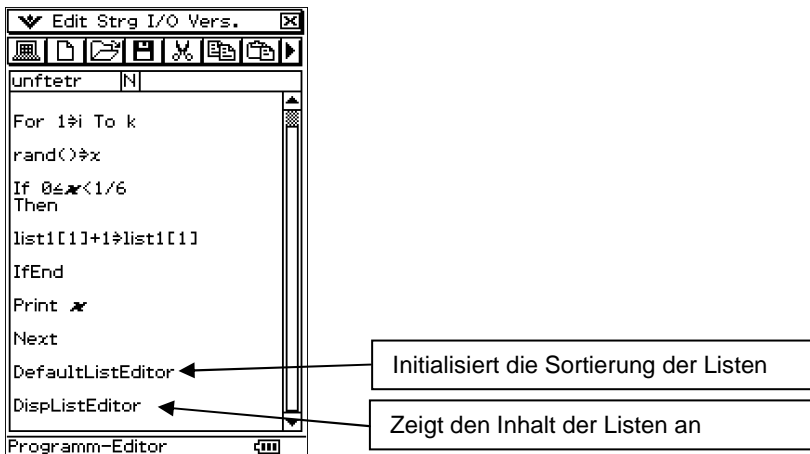
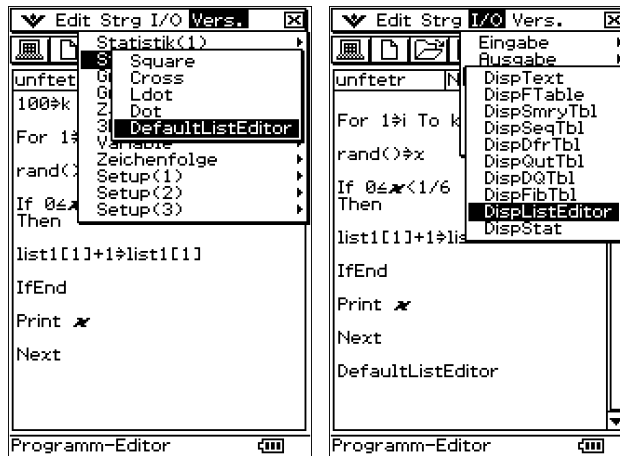
Unter *I/O* finden sich hauptsächlich Befehle zur Ein- und Ausgabe für Texte, Tabellen, Graphen.

Für das Tetraeder-Programm soll der Wert  $x$  ausgegeben werden. Dazu wird der Befehl *Print* genutzt, der sich unter *Ausgabe* verbirgt.



**Programmbefehle (Vers.)**

Abschließend soll eine Liste als Statistik der Würfe angezeigt werden. Wir benötigen dafür zunächst den Befehl *DefaultListEditor*, der sich im Untermenü *Vers* unter *Statistik(2)* befindet. Dieser Befehl wird ergänzt durch *DispListEditor* aus dem Untermenü *I/O* und *Display*.



**Ausführung des Programms**

Das fertige Programm kann nun ablaufen. Hierzu wird mit die Eingabeoberfläche verlassen. Dann kann mit das Programm ausgeführt werden. (Hinweis: Tragen Sie vorher eine 0 in list1[1] ein.)

Das Programm, welches Sie in der vcp-Datei im Anhang finden, heißt *tetraunf* und wurde zur besseren Nutzbarkeit um einige Befehle erweitert. Der Kern des Programmes ist aber identisch mit dem Programm *unftetr*.

```

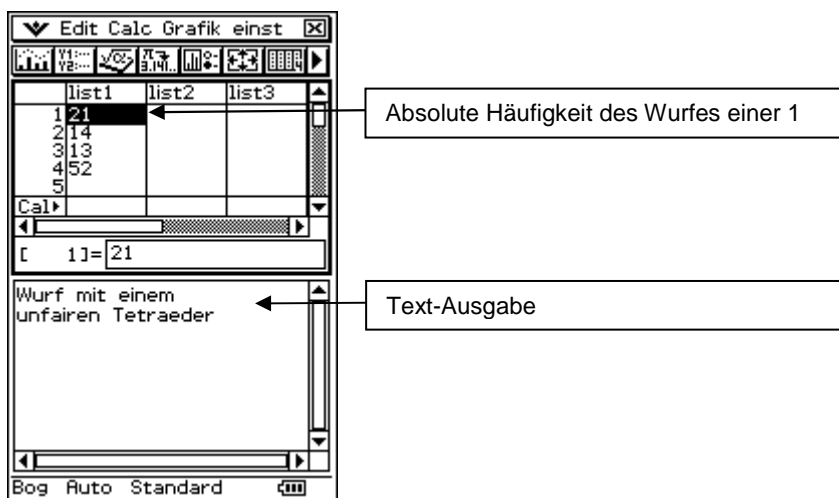
tetraurf [N]
Print "Wurf mit einem"
Print "unfairen Tetraeder"
{0,0,0,0}→list1
100→k
For 1→i To k
rand()→x
If 0≤x<1/6
Then
list1[1]+1→list1[1]
ElseIf 1/6≤x<2/6
Then
list1[2]+1→list1[2]
ElseIf 2/6≤x<3/6
Then
list1[3]+1→list1[3]
Else
list1[4]+1→list1[4]
IfEnd
Next
DefaultListEditor
DispListEditor
    
```

Ausgabe eines erklärenden Textes

list1[1] bis list1[4] wird Null gesetzt

Es werden nicht nur die Einsen gezählt, sondern in list1[2] bis list1[4] auch die Zweien, Dreien und Vieren. Die Werte für x werden nicht ausgegeben, sondern nur gezählt.

Die Ergebnisse des Programms *tetraurf* können folgendermaßen aussehen:



Die Einträge im Ergebnisfenster können im Menü Edit unter *Alle löschen* gelöscht werden.

**Kommentierte Beispiele**

**Bsp. 1** Mit 2 Würfeln wird n-mal gewürfelt. Betrachtet werden die Differenzen der jeweiligen Augenzahlen.  
 Die Verteilung wird ausgedruckt bzw. im Histogramm dargestellt.

Bei der Ausführung wird die Eingabe eines Wertes für n, die Anzahl der Würfelversuche, erwartet.

Den Zählern für die einzelnen Differenzen wird der Wert 0 zugewiesen.

Den Würfeln w und v wird eine Zufallszahl zwischen 1 und 6 zugeordnet.

Die Differenz wird ermittelt und u zugeordnet.

Entsprechend der Höhe der Differenz u wird der zugehörige Zähler um 1 erhöht.

Der Vorgang wird für den nächsten Versuch wiederholt.

Die Anzahlen der jeweiligen Differenzen für die gewählte Wurfanzahl werden angegeben.

Ersetzt man in dem Programm sämtliche Print-Anweisungen durch diese vier Zeilen, wird automatisch ein Histogramm erstellt, wenn als Schrittweite 1 gewählt wird.

```

    Edit Strg I/O Vers.
    minus | N
    Input n
    0→a:0→b:0→c:0→d:0→e:0→f
    For 1→k To n
    rand(1,6)→w
    rand(1,6)→v
    abs(v-w)→u
    If u=0
    Then
    a+1→a
    IfEnd
    If u=1
    Then
    b+1→b
    IfEnd
    If u=2
    Then
    c+1→c
    IfEnd
    If u=3
    Then
    d+1→d
    IfEnd
    If u=4
    Then
    e+1→e
    IfEnd
    If u=5
    Then
    f+1→f
    IfEnd
    Next
    Print "Differenz 0:": Print a
    Print "Differenz 1:": Print b
    Print "Differenz 2:": Print c
    Print "Differenz 3:": Print d
    Print "Differenz 4:": Print e
    Print "Differenz 5:": Print f
    Print " "
    Print "Anzahl der Würfe:": Print n
    Print " "
    Print "-----"
    
```

Programm-Editor

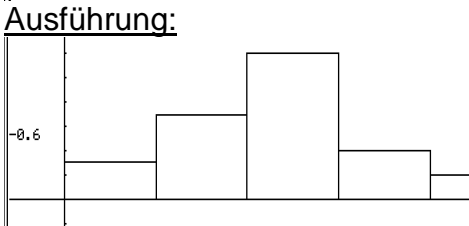
**Ausführung:**

```

    Differenz 0:
    5
    Differenz 1:
    11
    Differenz 2:
    3
    Differenz 3:
    6
    Differenz 4:
    4
    Differenz 5:
    1
    Anzahl der Würfe:
    30
    
```

```

    {0,1,2,3,4,5}→list1
    {a,b,c,d,e,f}→list2
    StatGraph 1, On, Histogram, list1, list2
    DrawStat
    
```



**Bsp. 2 Näherungswert für  $\pi$**

Einschalten der Dezimaldarstellung	<pre> Edit Strg I/O Vers. pi N BetDecimal 0⇨B For 1⇨n To 1000   rand()⇨x   rand()⇨y   √(x^2+y^2)⇨A   If A≤1   Then   B+1⇨B   IfEnd Next Print "Kreisfläche:" Print approx((4×B)/1000)                 </pre>
Zähler für die Punkte im Viertelkreis wird auf Null gesetzt.	
Es werden 1000 Punkte gesetzt.	
Den beiden Koordinaten eines Punktes wird eine Zufallszahl zwischen 0 und 1 zugeordnet.	
Wenn der Abstand zum Nullpunkt höchstens Eins ist, wird der Zähler für die Punkte im Viertelkreis um 1 erhöht.	
Der Näherungswert für $\pi$ wird berechnet und gedruckt.	

**Bsp. 3 Mit welcher Wahrscheinlichkeit haben in einer Gruppe von N Personen mindestens 2 Personen am gleichen Tag Geburtstag?**

Bei der Ausführung wird nach der Gruppengröße gefragt.	<pre> Edit Strg I/O Vers. Geburt N   Input N 'Anzahl in einer Gruppe 0⇨g For 1⇨k To 10   0⇨h   randlist(N,1,365)⇨list1   MultiSortA list1   For 1⇨i To N-1   If list1[i]=list1[i+1]   Then   h+1⇨h   IfEnd Next If h≥1   Then   g+1⇨g   IfEnd Next Print "mind.2 gleich; Anzahl:" Print g Print "Wahrscheinlichkeit für mind. 2 gleich" Print "bei": Print N: Print "Personen" Print approx(g/10)                 </pre>
Kommentar im Programm (ohne Funktion)	
Es wird 10mal eine Gruppe der Größe N betrachtet.	
In Liste 1 werden N Zufallszahlen zwischen 1 und 365 erzeugt und sortiert.	
In der Liste 1 wird jede Zahl mit dem Nachfolger verglichen. Bei Gleichheit wird der Zähler h um 1 erhöht („2 Leute haben am selben Tag Geburtstag“).	
Wenn h mindestens Eins ist, ist diese Gruppe eine von 10 Gruppen, in denen mindestens zwei Personen am gleichen Tag Geburtstag haben.	
Die Wahrscheinlichkeit wird als relative Häufigkeit aus 10 Simulationen bei fester Gruppengröße N berechnet und ausgedruckt.	

**Bsp. 4 Simulation linearen Wachstums durch Würfeln einschließlich graphischer Darstellung**

<p>Es wird 100mal gewürfelt.</p> <p>Liste 2 wird mit den Zahlen 1 bis 20 gefüllt. Der Wert für das jeweilige n wird in Liste 3 notiert.</p> <p>Es werden n Zufallszahlen zwischen 1 und 6 in Liste 1 notiert.</p> <p>k zählt die Anzahl der Sechsen bei n Versuchen.</p> <p>Die Werte für k werden passend in Liste 4 notiert.</p> <p>n wird linear um 60 erhöht; die Versuchsreihe wird wiederholt.</p> <p>Die Anzahl der Sechsen in Liste 4 wird abhängig von der Versuchsnummer in Liste 2 graphisch dargestellt.</p> <p>Durch Verändern des Anfangswertes 100 und der Zunahme 60 lassen sich die Streuungen variieren.</p>	<div style="border: 1px solid black; padding: 5px;"> <p>▼ Edit Strg I/O Vers.</p> <p>Wachstum   N</p> <pre> 100⇒n For 1⇒j To 20 j⇒list2[j] n⇒list3[j] 0⇒k randList(n,1,6)⇒list1 For 1⇒i To n If list1[i]=6 Then k+1⇒k IfEnd Next k⇒list4[j] n+60⇒n Next StatGraph 1, On, Scatter, list2, list4, 1, Square DrawStat                     </pre> </div> <p><b>Ausführung:</b></p>
--	--

**Bsp. 5 Durch zwei beliebige Punkte P und Q verläuft eine Gerade. Es soll die Gleichung einer linearen Funktion bestimmt werden, deren Graph diese Gerade darstellt. (Aufgabe aus Weigert, Peter: ClassPad – Unterrichtsideen für die Sekundarstufe I, Troisdorf 2009, S.52)**

<p>Abfrage der 4 Koordinaten bei der Programmausführung</p> <p>Berechnung der Steigung m</p> <p>Berechnung des y-Achsenabschnitts b</p> <p>Ausgabe der Werte für m und b</p>	<div style="border: 1px solid black; padding: 5px;"> <p>▼ Edit Strg I/O Vers.</p> <p>Gerade   N</p> <pre> Input xP Input yP Input xQ Input yQ (yP-yQ)/(xP-xQ)⇒m yP-m*xP⇒b Print "y=mx+b mit m=" Print m Print "und b=" Print b                     </pre> </div>
--	--



**Bsp. 6** Gesucht ist die Lösung der Ungleichung  $\binom{x}{9} \cdot 9! \leq 2176782336$ .

Bei der Ausführung wird die Eingabe eines Wertes für x erwartet, zweckmäßig wählt man 9 oder eine etwas größere Zahl.

Solange die Ungleichung erfüllt ist, wird der Wert von x um 1 erhöht.

Wenn die Ungleichung zum ersten Mal nicht erfüllt ist, wird der vorige x-Wert ausgedruckt.

```
▼ Edit Strg I/O Vers.
Gewinnsp | N
Input x
While nCr(x,9)*9! ≤ 2176782336
  x+1 → x
WhileEnd
Print "x=": Print x-1
```