Lektion 4: Schleifen | **Übung 2: While-Schleife**

In dieser Übung wirst du von der vielseitigsten Schleife lernen: **While…EndWhile**

Lernziele:

- Eine einfache **While**-Schleife schreiben
- Mit einer **While**-Schleife die Dateneingabe prüfen

Das mächtige While…

Die **While…EndWhile**-Schleife läuft so lange wie die <*Bedingung*> *wahr* bleibt. Das sieht folgendermaßen aus:

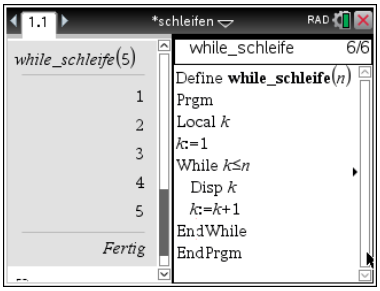
<*Setzen des Anfangswerts der Bedingung* >

While <*Bedingung*>

<*Schleifenkörper*>

EndWhile

- Vor dem Start der Schleife müssen eine oder mehrere Variable so definiert werden, dass die **While**-Anweisung die Bedingung beim ersten Mal auswerten kann. Damit erhält die Variable den Wahrheitswert *true* oder *false*. Bei einer Anfangsbedingung = *false*, wird die Schleife einfach übersprungen, anderenfalls wird in die Schleife eingetreten.
- Die <*Bedingung*> ist ein logischer Ausdruck wie z.B. **x>0**.
- Der <*Schleifenkörper*> ist eine Sammlung von Anweisungen, die weitere Schleifen und **If**-Konstrukte beinhalten kann. Er wird so lange durchgeführt, so lange die <*Bedingung*> *true* (wahr) bleibt.
- Das reservierte Wort **EndWhile** zeigt das Ende des <*Schleifenkörpers*> an. An dieser Stelle verzweigt das Programm zurück an den Beginn der Schleife und überprüft die <*Bedingung*>. Ist diese nicht (mehr) erfüllt, dann wird die Schleife verlassen, sonst wird sie ein weiteres Mal durchlaufen.



k:=1 am Beginn des Programms setzt die Anfangsbedingung auf einen bekannten Wert. Ohne diese Initialisierung könnte die Variable **k** einen gespeicherten Wert annehmen und damit einen unbekannten Wert ins Programm bringen.

Innerhalb des <*Schleifenkörpers*> muss eine Anweisung vorkommen, die sich auf die <*Bedingung*> bezieht, so dass die Schleife dann auch einmal beendet wird und das Programm mit den folgenden Anweisungen weitergeführt wird. Üblicherweise findet sich eine derartige Anweisung am Ende der Schleife. Hier wird durch **k:=k+1** sichergestellt, dass **k** einmal größer wird als **n**.

Dieser **While**-Schleife entspricht die folgende **For**-Schleife:

```
For k, 1, n
  Disp k
EndFor
```

Hinweis: VORSICHT! Endlosschleifen sind gefährlich! Wenn sich ein Programm in einer Endlosschleife gefangen hat, dann drücke und halte die on-Taste bis das Programm anhält (abbricht).

Sei bei der Computer Software besonders vorsichtig, da sich da die on-Taste anders als am Handheld verhält.

Hinweis: Es ist wichtig darauf hinzuweisen, dass eine **While**-Schleife unter Umständen überhaupt nicht durchlaufen werden muss. In der nächsten Übung werden wir die **Loop**-Schleife behandeln, die zumindest einmal durchlaufen wird. Das ist ein kleiner aber wichtiger Unterschied.

Drei Komponenten machen eine erfolgreiche **While**-Schleife aus: ***Inialisierung, Überprüfung und Änderung***:

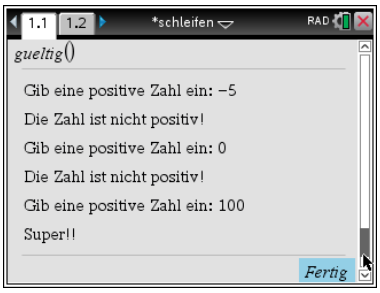
- ***Initialisiere*** eine Variable,
- ***Überprüfe*** die Bedingung bezüglich dieser Variablen,
- ***Ändere*** die Variable so, dass die Bedingung möglicherweise nicht erfüllt ist und somit die Schleife verlassen wird.

Überprüfen, ob eine Eingabe gültig ist mit While…EndWhile

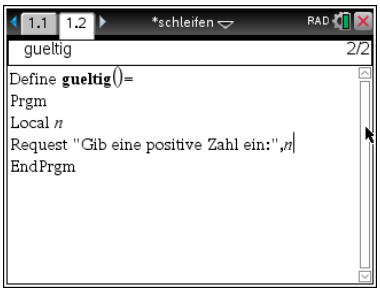
Wir werden einen Programmteil entwerfen, der sicher stellen soll, dass ein allfälliger Anwender eine positive Zahl eingibt. Es soll dem Anwender anzeigen, wenn die Eingabe falsch ist und ihn zu einer neuerlichen Eingabe auffordern.

Die Ausgabe für diesen Teil ist rechts gezeigt. Einige falsche Eingaben (weil negativ) wurden gemacht, um die Auswirkung zu demonstrieren.

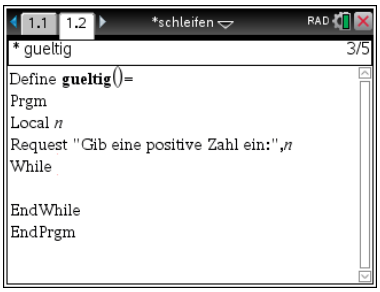
Versuche, diesen Teil zu entwerfen, ohne auf die nächste Seite zu schauen!



1. Wir beginnen mit einem neuen Programm namens **gueltig**.
2. Definiere eine lokale Variable **n** und verwende **Request** zur Eingabeaufforderung durch den Anwender. Beachte, dass der Anwender um eine *positive Zahl* gefragt wird.

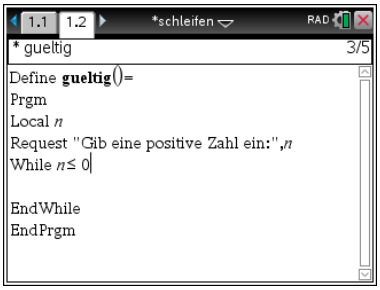


3. Füge die **While**-Anweisung vom Steuerungsmenu ein. Sowohl **While** als auch **EndWhile** werden ins Programm eingefügt und die Schreibmarke erscheint hinter dem Wort **While**.

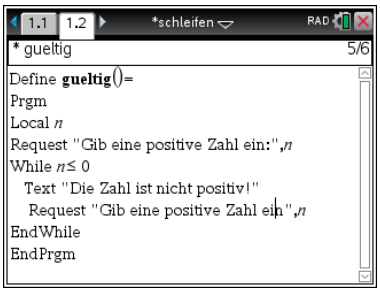


4. Lege die Anfangsbedingung fest mit **n≤0**.

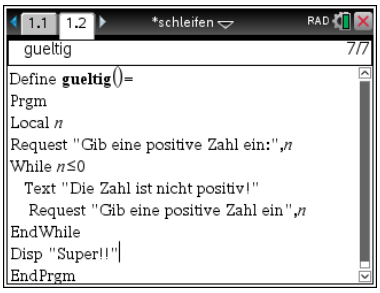
   - Hole den ≤ Operator über ctrl =.



5. Vervollständige nun den Schleifenkörper, indem du eine Fehlermeldung mit der **Text**-Anweisung entwirfst. Mit einer weiteren **Request**-Anweisung fordere den Anwender zu einer neuerlichen – richtigen – Eingabe für *n* auf.



Das rechts gezeigte Programm **gueltig( )** erzeugt den oben abgebildeten Dialog. Beachte die BEIDEN **Request**-Anweisungen.



- Die erste dient dazu, die Bedingung (*n*≤0) zu initialisieren. Wird hier eine positive Zahl eingegeben, dann wird die Schleife nicht ausgeführt.
- Wenn aber mit 0 oder einer negativen Zahl geantwortet wird, dann erzeugt die Schleife die Fehlermeldung und verlangt nach einer anderen Zahl.

Der Schleifenkörper wird so lange ausgeführt, wie eine negative Zahl eingegeben wird. Dann wird die **Disp**-Anweisung mit der „Belohnung“ ausgeführt.

Hinweis: Es ist sehr angenehm, dass man im Editor kopieren und einfügen kann!
Du kannst ‘>=’ am Computer eingeben und das ctrl B konvertiert die beiden Zeichen in eines.