Lektion 4: Schleifen	Übung 2: Die While…End-Schleife
In dieser zweiten Übung von Lektion 4 wird die **While…End**-Schleife untersucht. Sie wird mit der **For…** - Schleife verglichen, und es wird sich zeigen, dass sie leistungsstärker und vielseitiger ist.	**Lernziele:** • Verstehen der **While…End** - Schleife. • Vergleich mit der **For…End** - Schleife. • Programmierübung: gültige Eingaben.

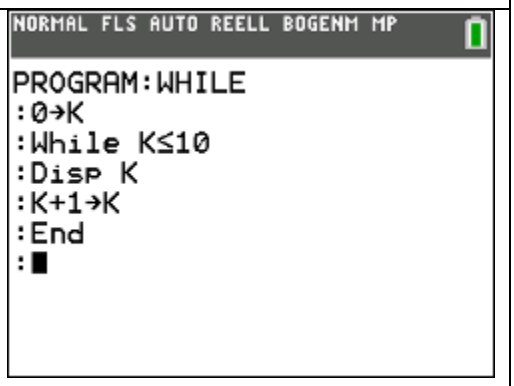
Die While… End - Schleife

Die Befehle innerhalb der **While…End** – Schleife werden so lange wiederholt wie die *Bedingung* wahr ist.

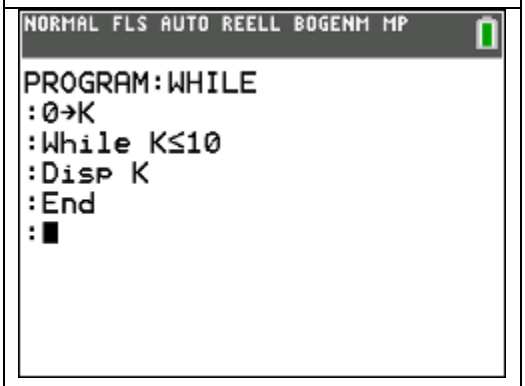
Aufbau:

While <Bedingung>
<Befehlsblock>
End

Hinweise:
Die Bedingung *ist ein logischer Ausdruck wie z.B.* **X>0.**
Der Befehlsblock besteht aus einer Gruppe von Anweisungen, die durchaus ***If*** *– Befehle und andere Schleifen enthalten dürfen. Er wird ausgeführt, solange die* Bedingung *wahr ist.*
Das Schlüsselwort **End** *kennzeichnet das Ende des Befehlsblocks. Das Programm springt zurück zum* **While** *und testet erneut, ob die* Bedingung *wahr ist.*



Den in der *Bedingung* verwendeten Variablen müssen vor der **While** – Schleife Werte zugewiesen werden, so dass man genau sagen kann, ob die *Bedingung* wahr oder falsch ist. Ist die *Bedingung* falsch, wird der Befehlsblock vollständig übersprungen, andernfalls wird er ausgeführt. Die Anweisung **0→K** am Anfang des Programmes sorgt dafür, dass die Bedingung zunächst wahr ist. Ohne diesen Wert wüsste man nicht was passiert, denn in **K** kann irgendein Wert abgespeichert sein.



Irgendwo im Befehlsblock sollte ein Befehl stehen, der eine Auswirkung auf die Bedingung hat, so dass sie beendet wird, damit die nachfolgenden Befehle ausgeführt werden können. Üblicherweise ordnet man diesen Befehl am Ende des Befehlsblocks ein. **K+1→ K** stellt sicher, dass irgendwann **K** größer ist als 10. Fehlt ein solcher Befehl, wird die Schleife fortgeführt, so dass das Programm kein Ende findet. Das ist bei dem zweiten Programm rechts der Fall.

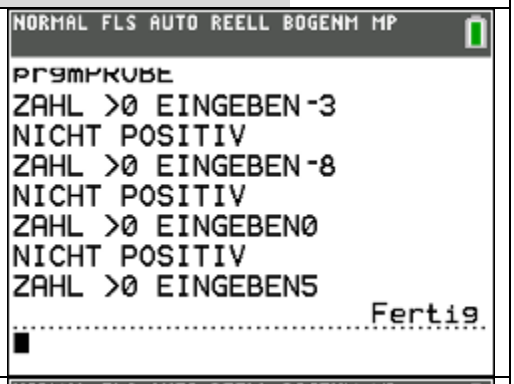
Hinweis: Bei der **While** – Schleife steht die *Bedingung* ganz am Anfang, so dass es durchaus möglich ist, dass die Anweisungen des Befehlsblockes überhaupt nicht ausgeführt werden. Im Gegensatz dazu wird bei der in der nächsten Übung behandelten **Repeat** – Schleife der Befehlsblock stets erst einmal durchlaufen, bevor die Bedingung getestet wird – ein feiner, aber deutlicher Unterschied.

Die **While** – Schleife besteht also aus den drei Komponenten: ***Initialisieren, testen und verändern***:

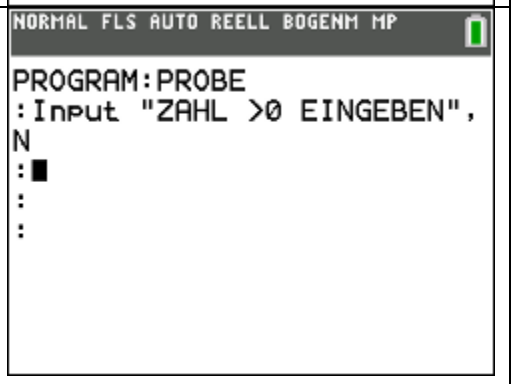
- ***Initialisieren*** einer Variablen.
- ***Testen*** einer auf dieser Variablen basierenden Bedingung.
- ***Verändern*** der Variablen so dass die Bedingung falsch wird und dadurch die Schleife endet.

Verwendung von While…End bei der Überprüfung einer Eingabe

Eingegebene Werte werden vor ihrer weiteren Verwendung oft geprüft, ob sie aus dem richtigen Bereich stammen. Im Beispiel soll ein kleines Programm geschrieben werden, das überprüft, ob eine eingegebene Zahl positiv ist. Es soll angezeigt werden, wenn die Eingabe ungültig ist, und es soll zu einer erneuten Eingabe aufgefordert werden. Das Bild rechts zeigt ein paar Eingaben, um die Wirkungsweise des Programmes vorzuführen.

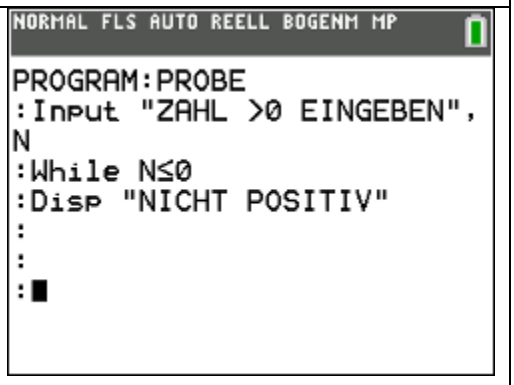


Mittels **Input** wird eine Zahl **N** eingelesen. Damit erhält man den ***Initialwert*** für die Schleifenbedingung.



Nun schließt sich die **While** – Schleife an, in der überprüft wird, ob **N** negativ ist. In diesem Fall wird mit **Disp** eine Fehlermeldung ausgegeben. Hier also wird N ***getestet***.

Hinweis: Der Befehlsblock wird nur ausgeführt, wenn N negativ ist, andernfalls wird er übersprungen.



Mit einem weiteren **Input** wird schließlich ein neuer Wert für N eingefordert und damit endet der Befehlsblock. Hier wird N also ***verändert***.

Als Teil eines größeren Programmes würde diese Befehlsfolge sicherstellen, dass nur bei Eingabe einer positiven Zahl das Programm weiter ausgeführt wird, so dass die eingegebene Zahl weiter verarbeitet werden kann.