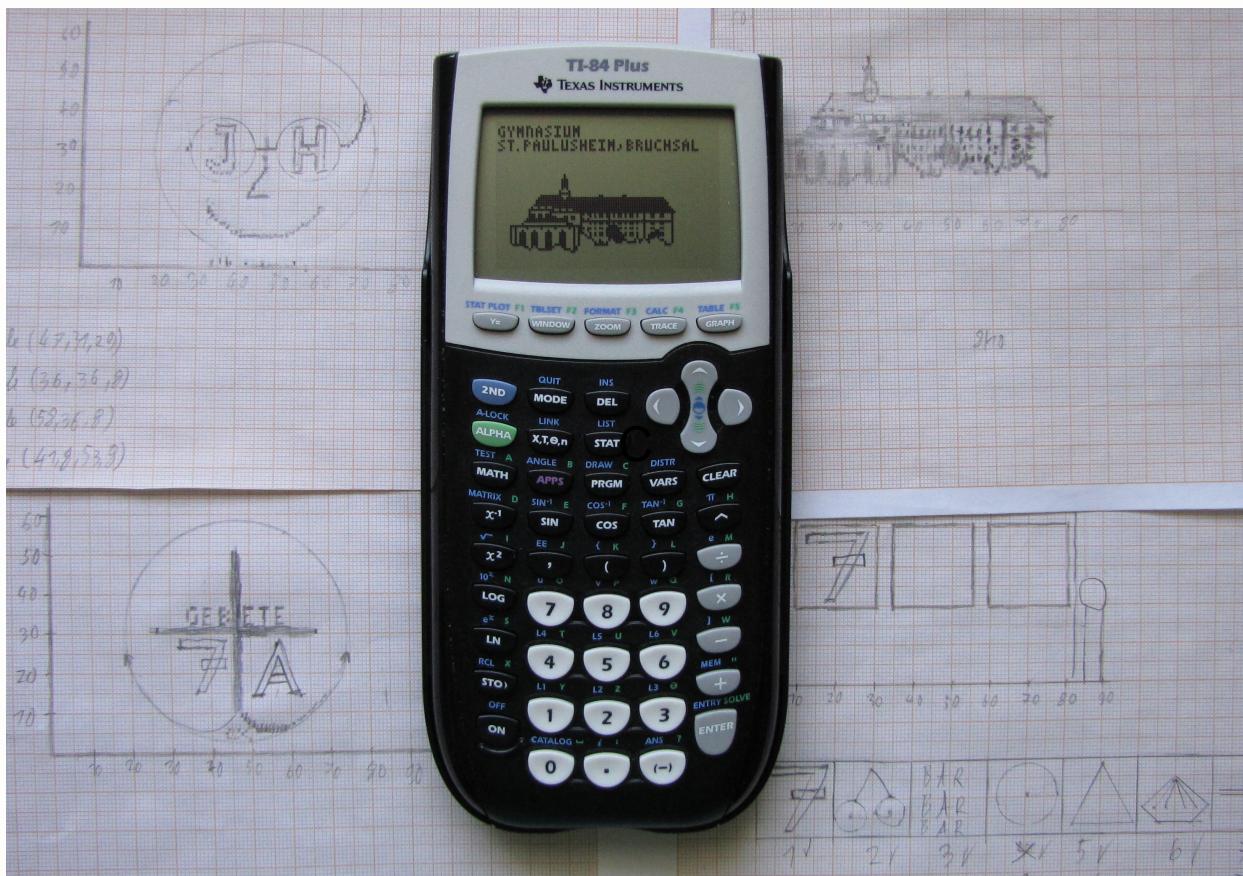


Programmieren auf dem TI-84 Plus

Projekt von Julian Hauck



Inhalt des Projekts

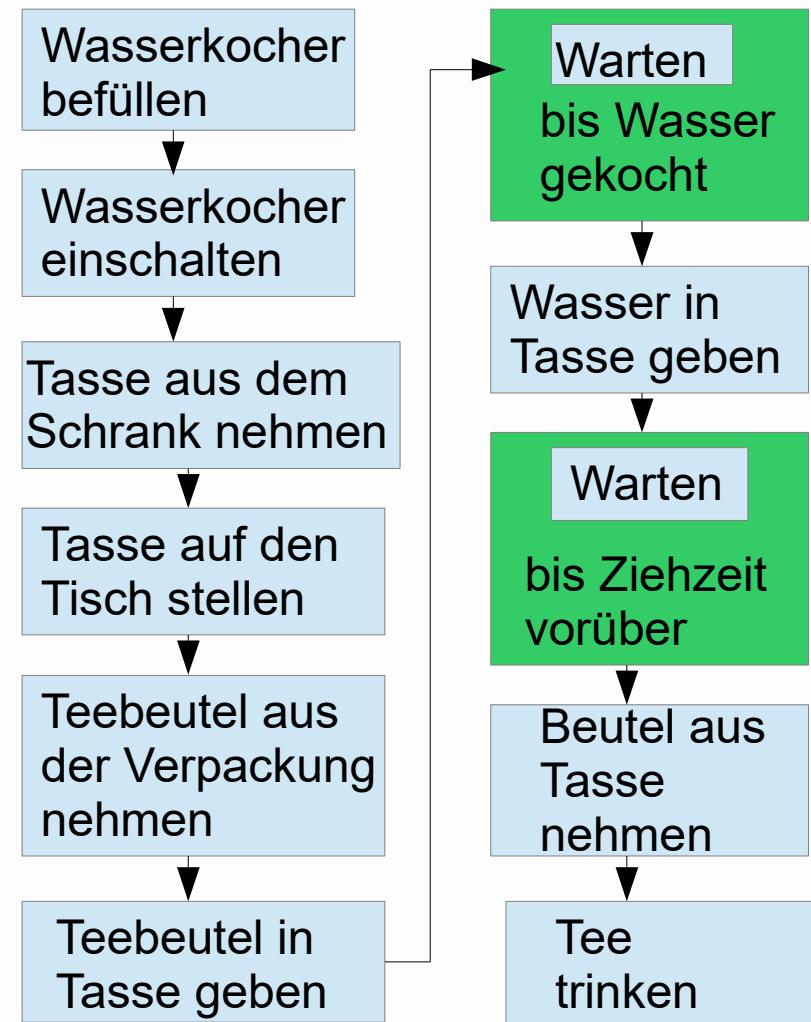
1. Erstellen von Flussdiagrammen
2. Erklärung der wichtigsten Befehle mit Hilfe von Beispielprogrammen
3. Selbstständiges Programmieren

Erstellen von Flussdiagrammen

- Wir stellen die Abläufe in einem Flussdiagramm dar, die dazu nötig sind, eine Tasse Tee zu richten.

Erstellen von Flussdiagrammen

- Wir stellen die Abläufe in einem Flussdiagramm dar, die dazu nötig sind, eine Tasse Tee zu richten.

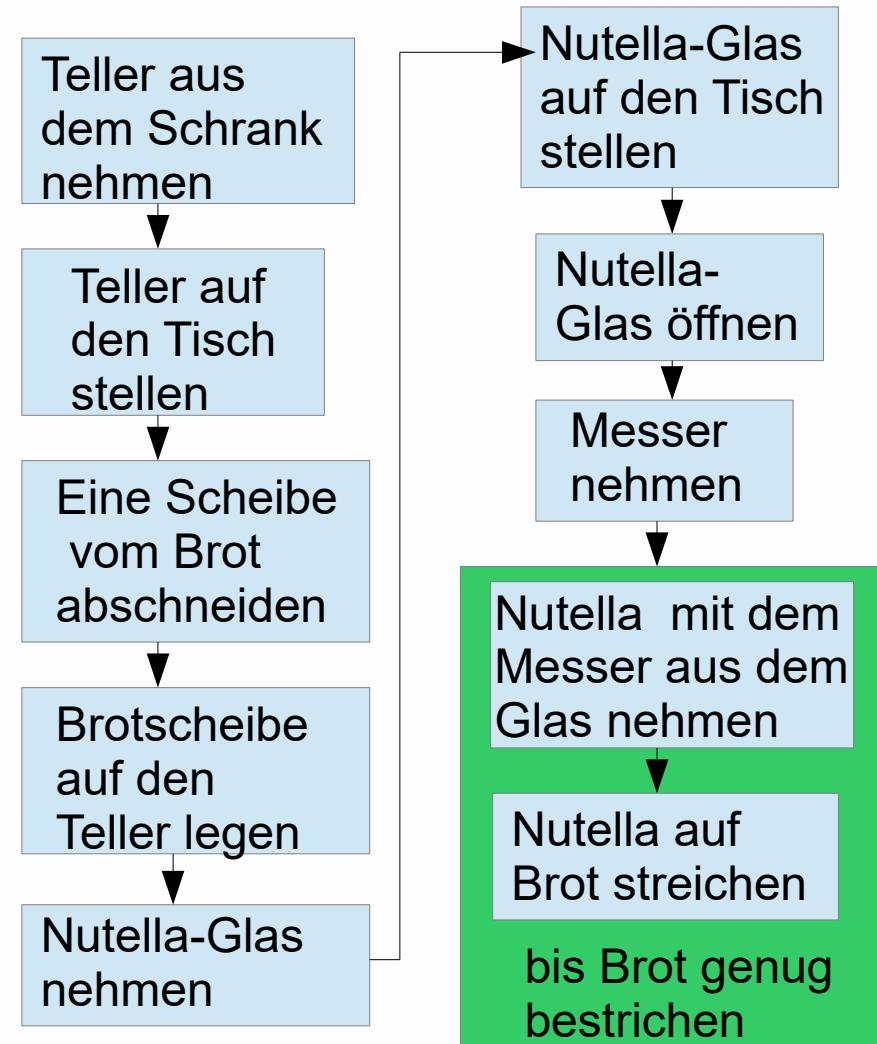


Erstellen von Flussdiagrammen

- Wir stellen die Abläufe in einem Flussdiagramm dar, die dazu nötig sind, ein Nutella-Brot zu richten.

Erstellen von Flussdiagrammen

- Wir stellen die Abläufe in einem Flussdiagramm dar, die dazu nötig sind, ein Nutella-Brot zu richten.



Variablen

- Definition: Variablen sind Speicherplätze, in denen z.B.
 - Zahlenwerte
 - Strings(beliebige Aneinanderreihungen von Zeichen)gespeichert werden.

Benutzen von Variablen



- Wir wollen der Variable A den Wert 10 zuordnen:
 $1 \rightarrow 0 \rightarrow \text{STO} \rightarrow \text{ALPHA} \rightarrow [A] \rightarrow \text{ENTER}$
- Abrufen des Wertes von A:
 $\text{ALPHA} \rightarrow [A] \rightarrow \text{ENTER}$

Benutzen von Variablen



- Wir ordnen der Variable String1 das Wort „GTR“ zu:

2ND→[ALOCK]→["]
→[G]→[T]→[R]→["]→
ALPHA→STO→
VARS→7→

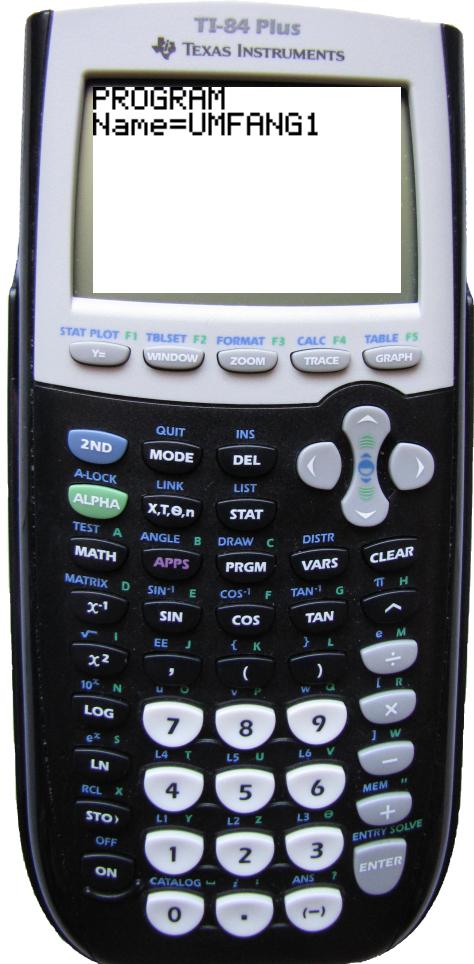
ENTER→ENTER

Erstellen unseres ersten Programms



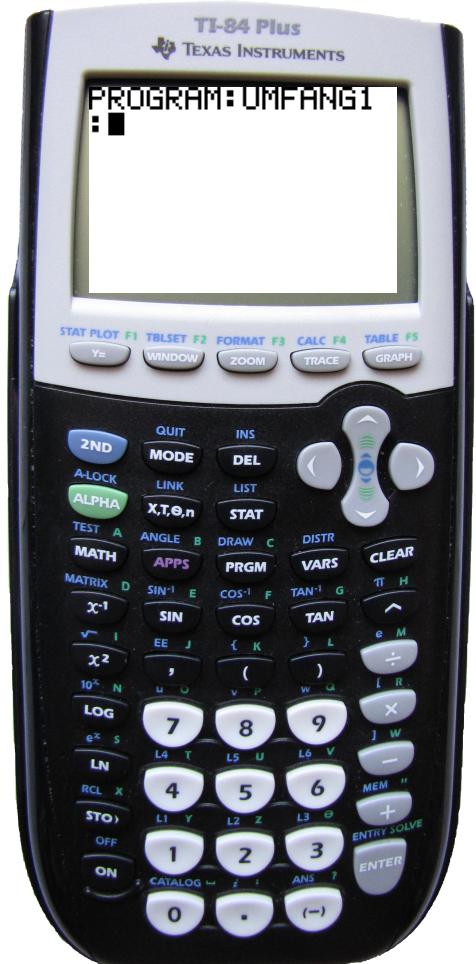
- PRGM → ➤ ➤ ➤

Erstellen unseres ersten Programms



- PRGM → ► → ► →
- ENTER → [U] → [M] →
- [F] → [A] → [N] → [G] →
- ALPHA → 1

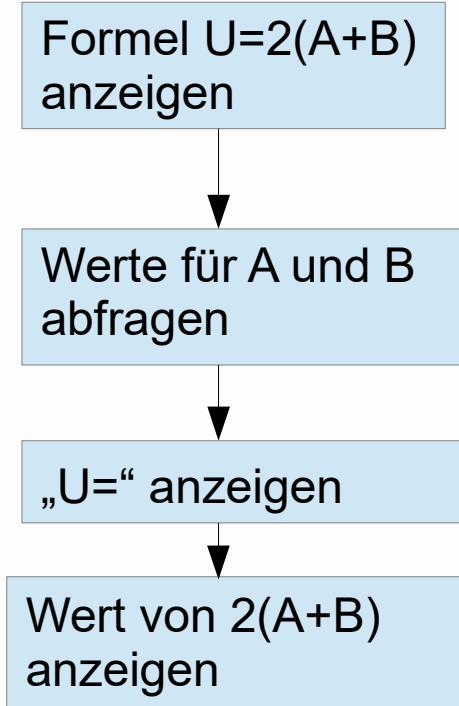
Erstellen unseres ersten Programms



- PRGM → ► → ► →
ENTER → [U] → [M] →
[F] → [A] → [N] → [G] →
ALPHA → 1 → ENTER

Umfang-Berechnung eines Rechtecks

PROGRAM: UMFANG1

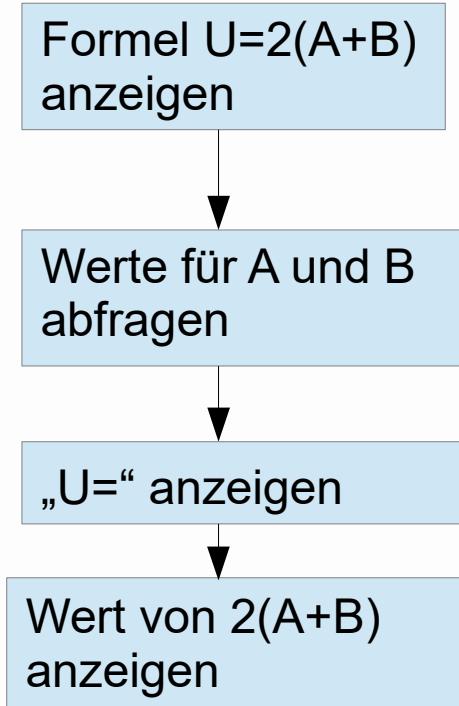


:

Der Disp-Befehl

- PRGM → ► → 3
CTL EXEC
1: Input
2: Prompt
3: Disp
4: DispGraph
5: DispTable
6: Output
7: getKey
- Disp Ausdruck/ *String*
- zeigt den **Wert** des dahinterstehenden Ausdrucks oder einen String an
- Um einen Ausdruck von einem String zu unterscheiden, schreibt man einen String in **Anführungszeichen**.

Umfang-Berechnung eines Rechtecks



PROGRAM: UMFANG1
:Disp „U=2(A+B)“

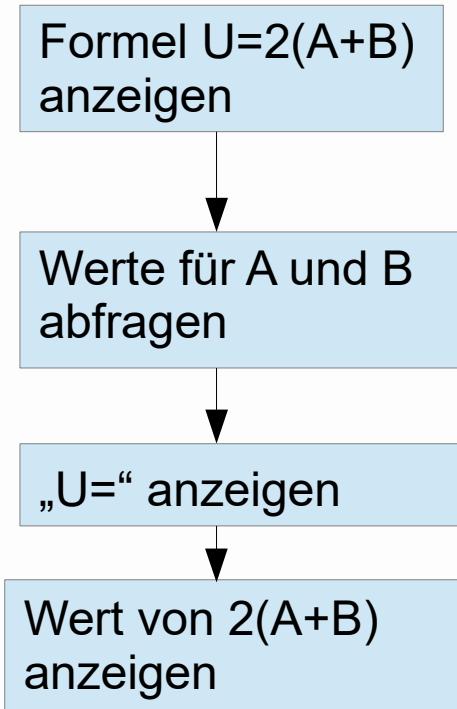
Der Prompt-Befehl

- PRGM → ► → 2

```
CTL EXEC
1:Input
2:Prompt
3:Disp
4:DispGraph
5:DispTable
6:Output
7:GetKey
```

- Prompt *Variable(n)* [*durch ein , abgetrennt*]
- fragt die Werte der dahinterstehenden *Variable(n)* ab

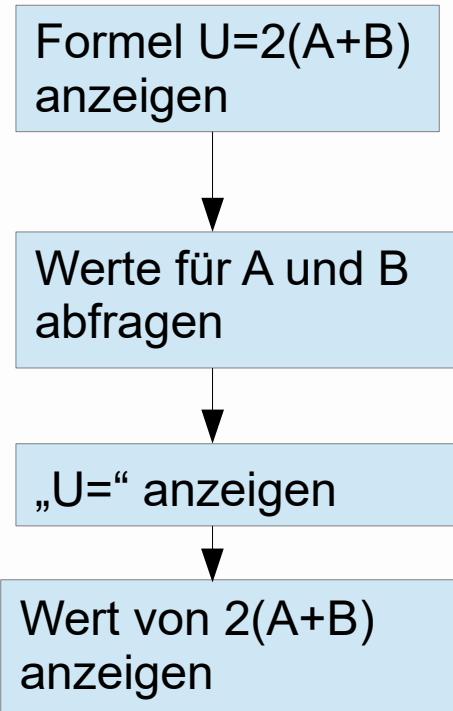
Umfang-Berechnung eines Rechtecks



PROGRAM: UMFANG1

:Disp „U=2(A+B)“
:Prompt A,B

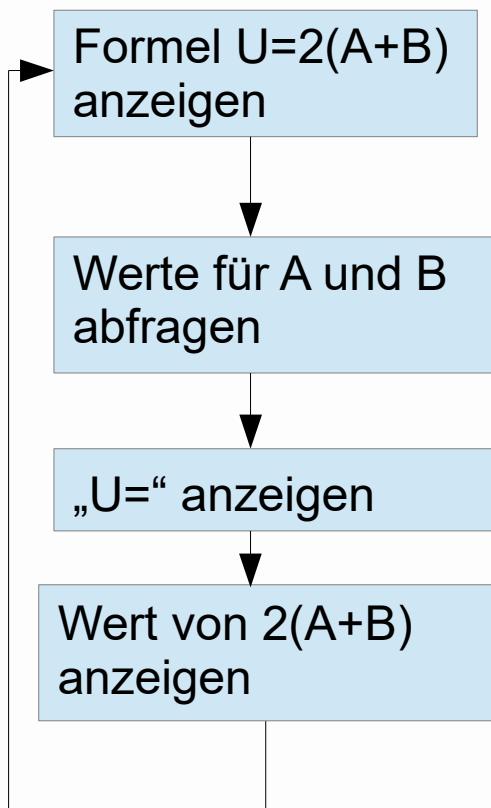
Umfang-Berechnung eines Rechtecks



PROGRAM: UMFANG1

```
:Disp „U=2(A+B)“  
:Prompt A,B  
:Disp „U=“  
:Disp 2(A+B)
```

Umfang-Berechnung eines Rechtecks



PROGRAM: UMFANG1

:Disp „U=2(A+B)“

:Prompt A,B

:Disp „U=“

:Disp $2(A+B)$

Der Lbl-Befehl

- PRGM→9

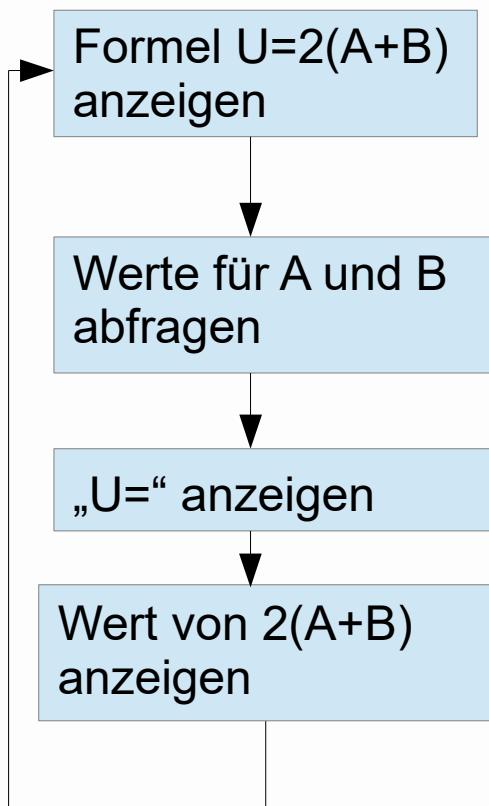
```
I/O EXEC  
?End  
0:Pause  
3:Lbl  
0:Goto  
A:IS><  
B:DS<<  
C↓Menu<
```

- Lbl *Name der Marke*
- definiert eine Marke

Der Goto-Befehl

- $PRGM \rightarrow 0$
 I/O EXEC
4:For
5:While
6:Repeat
7:End
8:Pause
9:Lbl
0:Goto
- Goto *Name der Marke*
- lässt das Programm an der *Marke* weiterlaufen

Umfang-Berechnung eines Rechtecks



PROGRAM: UMFANG1

:Lbl 1

:Disp „U=2(A+B)“

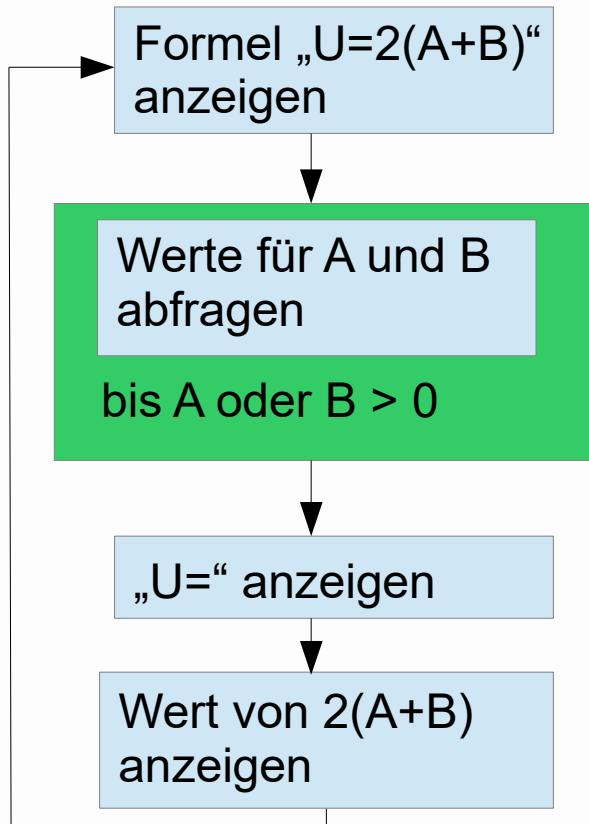
:Prompt A,B

:Disp „U=“

:Disp $2(A+B)$

:Goto 1

Umfang-Berechnung eines Rechtecks



PROGRAM: UMFANG1

```
:Lbl 1  
:Disp „U=2(A+B)“  
:Prompt A,B  
:Disp „U=“  
:Disp 2(A+B)  
:Goto 1
```

Der Repeat-Befehl

- PRGM → 6
- :Repeat *Bedingung*
 - : *Befehlsgruppe*
 - : End
- überprüft nach jedem Ausführen der Befehlsgruppe (also an der Stelle des End-Befehls), ob die Bedingung wahr ist.
Die Befehlsgruppe wird solange wiederholt, bis die Bedingung erfüllt ist.

Elemente zum Definieren von Bedingungen

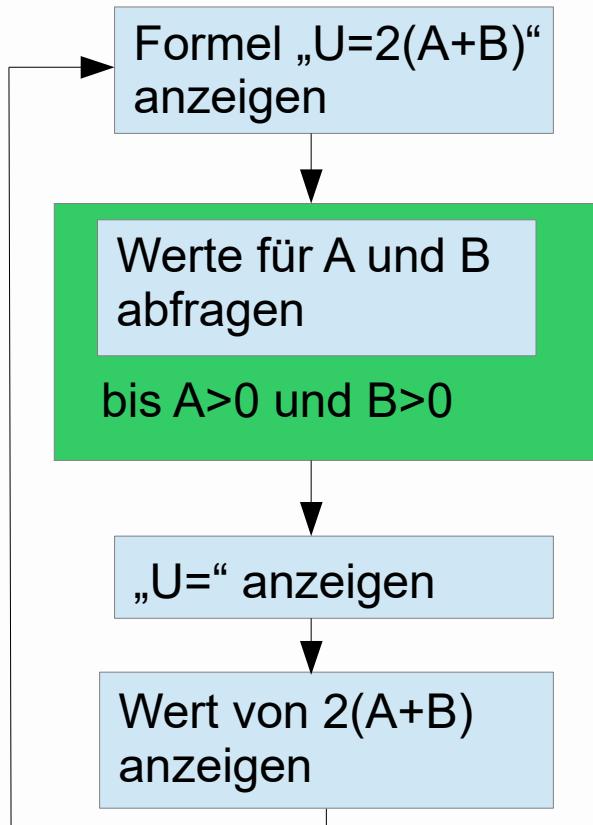
- 2ND→TEST

```
TEST LOGIC  
1:=  
2:≠  
3:>  
4:>  
5:<  
6:<
```

- 2ND→TEST→►

```
TEST LOGIC  
1:and  
2:or  
3:xor  
4:not<
```

Umfang-Berechnung eines Rechtecks



PROGRAM: UMFANG1

:Lbl 1

:Disp „U=2(A+B)“

:Repeat A>0 and B>0

:Prompt A,B

:End

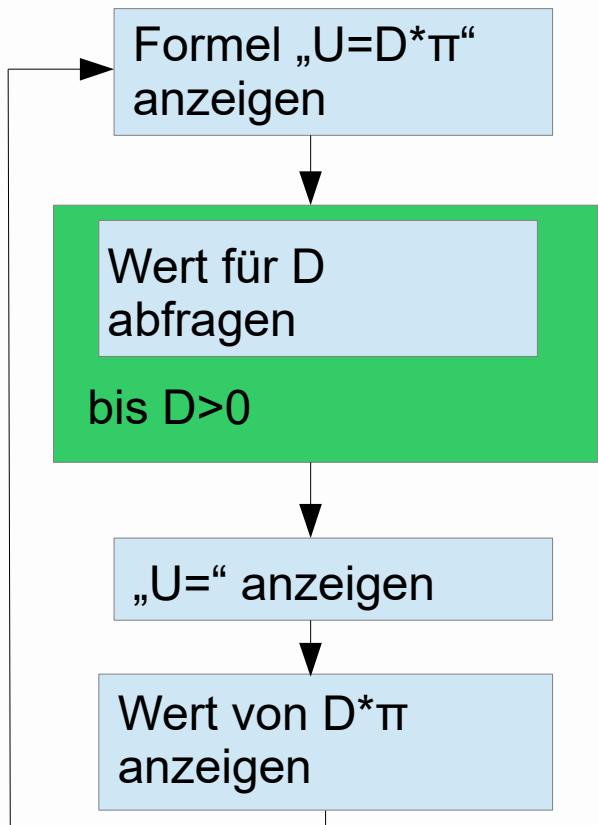
:Disp „U=“

:Disp 2(A+B)

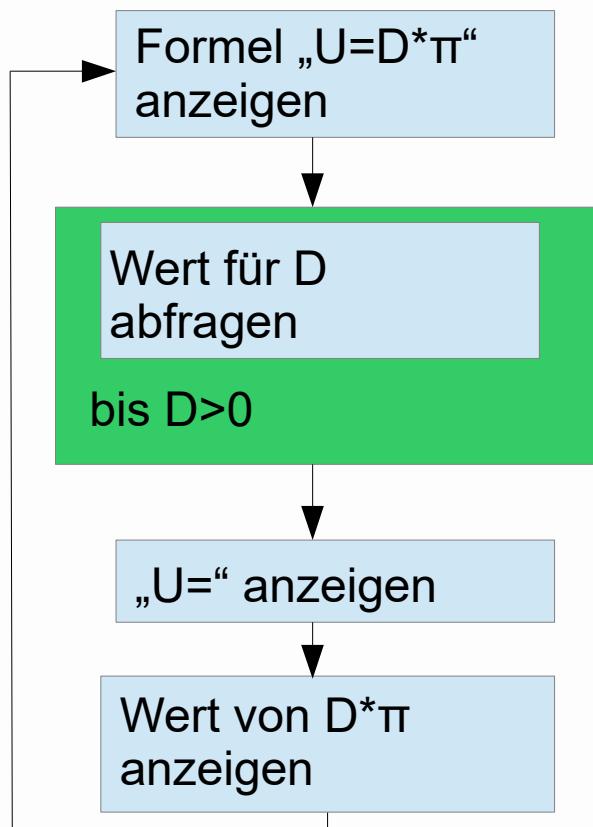
:Goto 1

Umfang-Berechnung eines Kreises

PROGRAM: UMFANG2



Umfang-Berechnung eines Kreises



PROGRAM: UMFANG2

```
:Lbl 1
:Disp „U=D*π“
:Repeat D>0
:Prompt D
:End
:Disp „U=“
:Disp 2(A+B)
:Goto 1
```

Umfang-Berechnung von Kreisen und Rechtecken



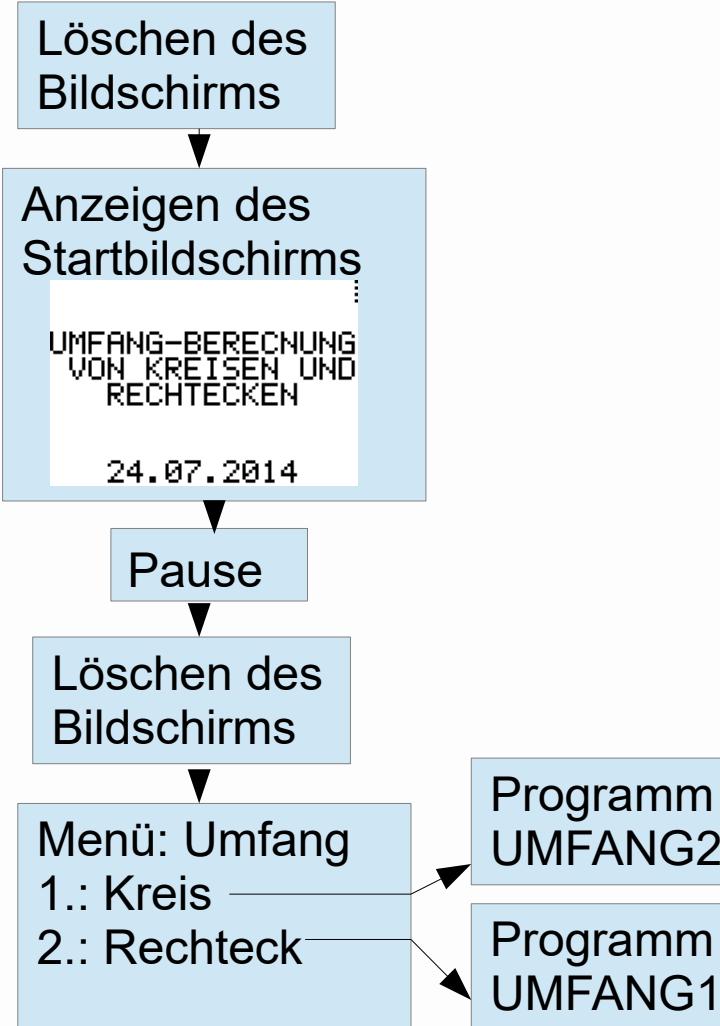
PROGRAM: UMFANG

:

Der ClrHome-Befehl

- PRGM→►8 CTL EXEC
3:Disp
4:DispGraph
5:DispTable
6:Output
7:getKey
8:ClrHome
9:ClrTable
- löscht den Hauptbildschirm

Umfang-Berechnung von Kreisen und Rechtecken



PROGRAM: UMFANG
:ClrHome

Der Output-Befehl

- PRGM → ► 6

```
CTL EXEC
3:Disp
4:DispGraph
5:DispTable
6:Output<
7:getKey
8:ClrHome
9:ClrTable
```

 - Output(*Zeile, Spalte, String/Ausdruck/Wert*)
 - Zeigt einen Text/ Wert an, der in der angegebenen Zeile(1-8) und Spalte(1-16) beginnt

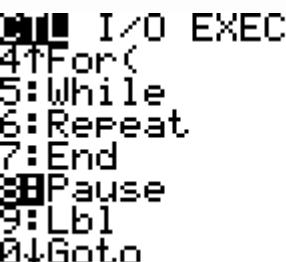
Umfang-Berechnung von Kreisen und Rechtecken



PROGRAM: UMFANG

```
:ClrHome  
:Output(3,1,"UMFANG-  
BERECHNUNG")  
:Output(4,2,"VON KREISEN  
UND")  
:Output(5,4,"RECHTECHK  
EN")  
:Output(8,4,"24.07.2014")
```

Der Pause-Befehl

- PRGM→8

 - 1:I/O EXEC
 - 2:For(
 - 3:While
 - 4:Repeat
 - 5:End
 - 6:Pause
 - 7:Lbl
 - 8:Goto
- hält das Programm so lange an, bis ENTER gedrückt wird

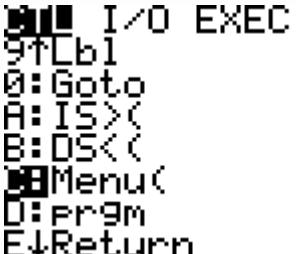
Umfang-Berechnung von Kreisen und Rechtecken



PROGRAM: UMFANG

```
:ClrHome
:Output(3,1,"UMFANG-
BERECHNUNG")
:Output(4,2,"VON KREISEN
UND")
:Output(5,4,"RECHTECHK
EN")
:Output(8,4,"24.07.2014")
:Pause
:ClrHome
```

Der Menu-Befehl

- PRGM→ALPHA→[C]

- *Menu(„Name des Menüs“, „Option1“, „Marke1“, „Option2“, „Marke2“, ...)*
- öffnet ein Menü, das die angegebenen Optionen anzeigt. Sobald eine Option ausgewählt wurde, geht das Programm an der dazugehörigen Marke weiter.

Umfang-Berechnung von Kreisen und Rechtecken



PROGRAM: UMFANG

```
:ClrHome
:Output(3,1,"UMFANG-
BERECHNUNG")
:Output(4,2,"VON KREISEN UND")
:Output(5,4,"RECHTECHK
EN")
:Output(8,4,"24.07.2014")
:Pause
:ClrHome
:Menu(„UMFANG“, „KREIS“, U1,
„RECHTECK“, U2)
:Lbl U1
```

Benutzen von Unterprogrammen

- 1. Möglichkeit:

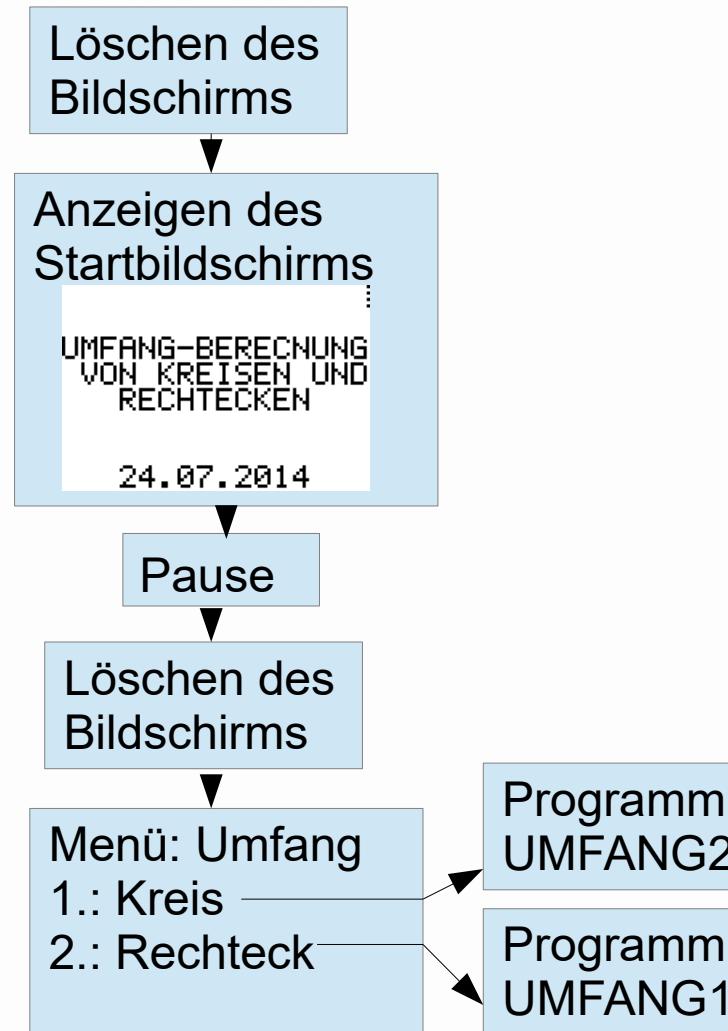
- PRGM → ◀
- Programm auswählen

```
CTL I/O EXEC
1: A
2: A1
3: A2
4: *AAA
5: AAAA
6: AAAAAAAA
7: ABC
```

- 2. Möglichkeit:

- PRGM → ALPHA → [D]
- *prgm Programmname*

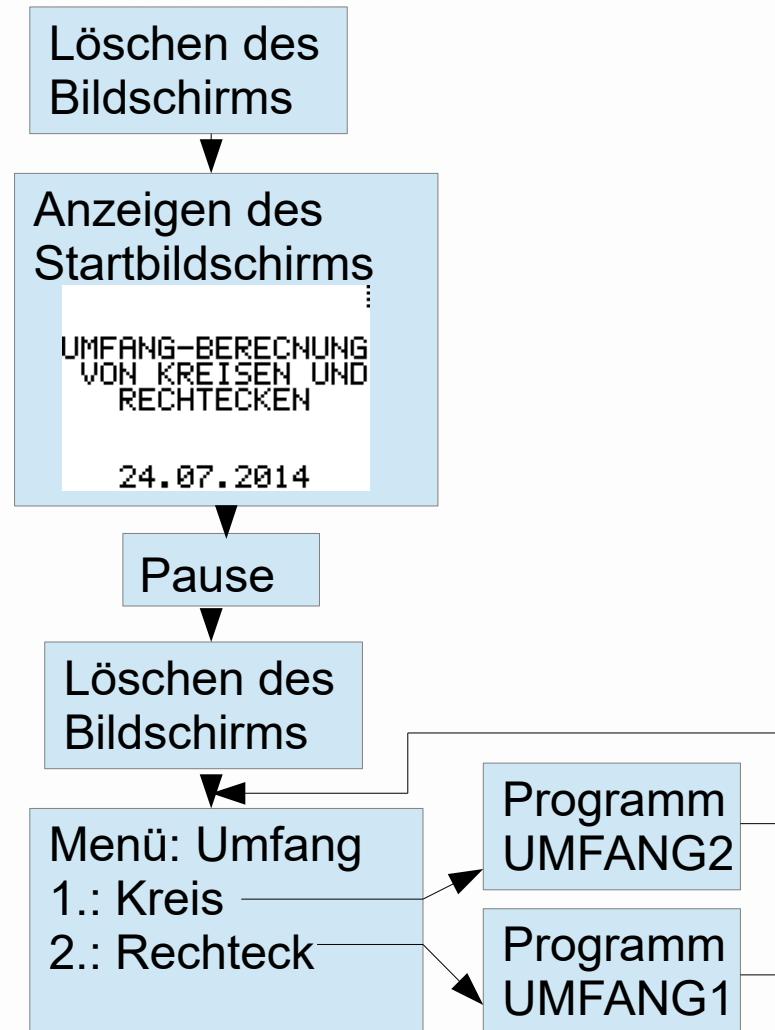
Umfang-Berechnung von Kreisen und Rechtecken



PROGRAM: UMFANG

```
:ClrHome  
:Output(3,1,"UMFANG-  
BERECHNUNG")  
:Output(4,2,"VON KREISEN UND")  
:Output(5,4,"RECHTECHK  
EN")  
:Output(8,4,"24.07.2014")  
:Pause  
:ClrHome  
:Menu(„UMFANG“, „KREIS“, U1,  
„RECHTECK“, U2)  
:Lbl U1  
:prgm UMFANG2  
:Lbl U2  
:prgm UMFANG1
```

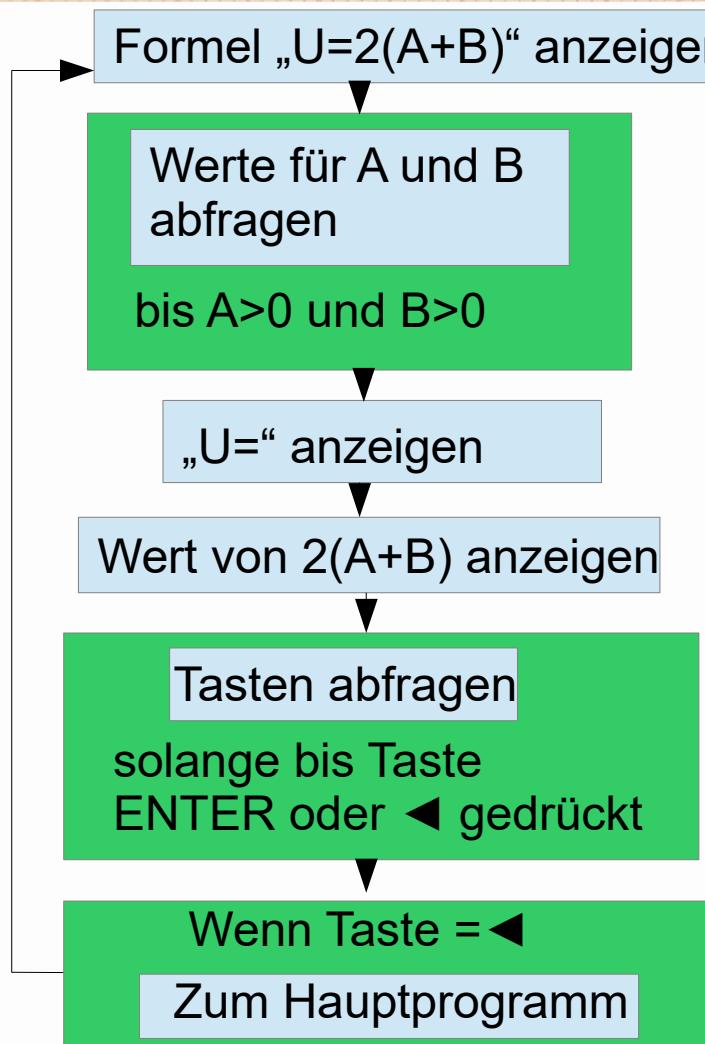
Umfang-Berechnung von Kreisen und Rechtecken



PROGRAM: UMFANG

```
:ClrHome
:Output(3,1,"UMFANG-BERECHNUNG")
:Output(4,2,"VON KREISEN UND")
:Output(5,4,"RECHTECHK
EN")
:Output(8,4,"24.07.2014")
:Pause
:Lbl M
:ClrHome
:Menu(„UMFANG“, „KREIS“, U1,
„RECHTECK“, U2)
:Lbl U1
:prgm UMFANG2
:Goto M
:Lbl U2
:prgm UMFANG1
:Goto M
```

Umfang-Berechnung eines Rechtecks



PROGRAM: UMFANG1

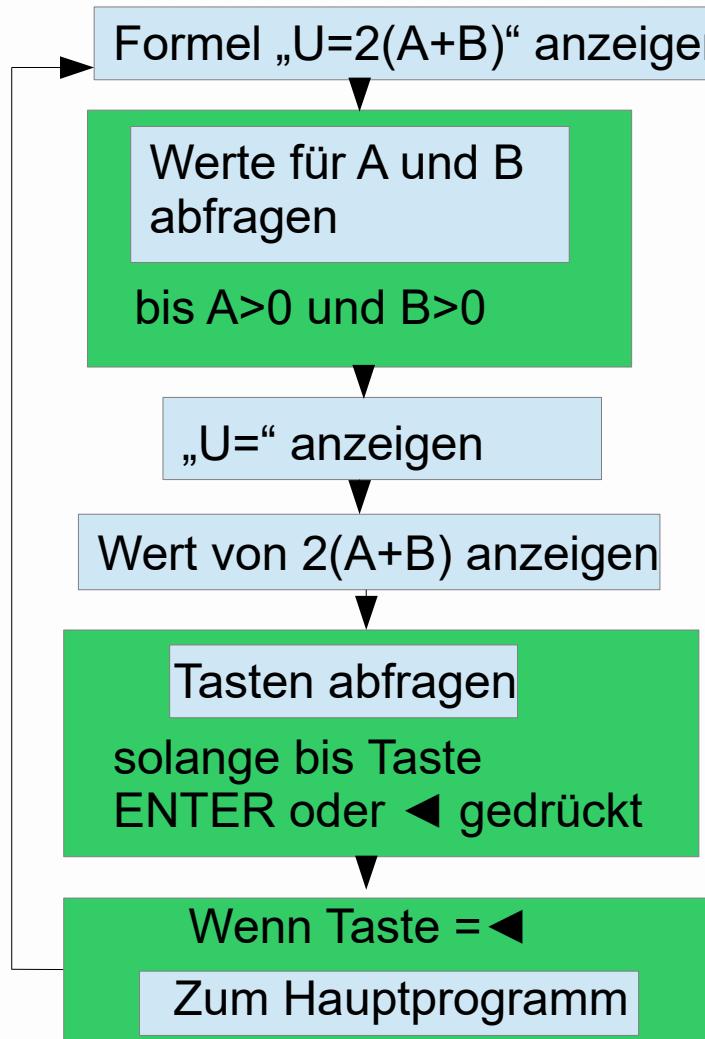
```
:Lbl 1  
:Disp „U=2(A+B)“  
:Repeat A>0 and B>0  
:Prompt A,B  
:End  
:Disp „U=“  
:Disp 2(A+B)  
:Goto 1
```

Der getKey-Befehl

- PRGM → ► 7
 - 1 CTL
 - 2 PROMPT
 - 3: DISP
 - 4: DISPGRAPH
 - 5: DISPTABLE
 - 6: OUTPUT
 - 7: GETKEY
 - 8: CLRHOME
- liefert den Wert der momentan gedrückten Taste. Wird keine Taste gedrückt, ist dieser 0.

11	12	13	14	15
21	22	23	24	25
31	32	33		34
41	42	43	44	45
51	52	53	54	55
61	62	63	64	65
71	72	73	74	75
81	82	83	84	85
91	92	93	94	95
	102	103	104	105

Umfang-Berechnung eines Rechtecks



PROGRAM: UMFANG1

```
:Lbl 1
:Disp „U=2(A+B)“
:Repeat A>0 and B>0
:Prompt A,B
:End
:Disp „U=“
:Disp 2(A+B)
:Repeat K=105 or K=24
:getKey →K
:End
```

Der If-Befehl

- PRGM→1  I/O EXEC
1:If
2:Then
3:Else
4:For(
5:While
6:Repeat
7↓End
- *:If Bedingung*
:Befehl, wenn Bedingung wahr
:Befehl
- lässt den **nächsten** Befehl nur ausführen, wenn die Bedingung wahr ist.

Der Return-Befehl

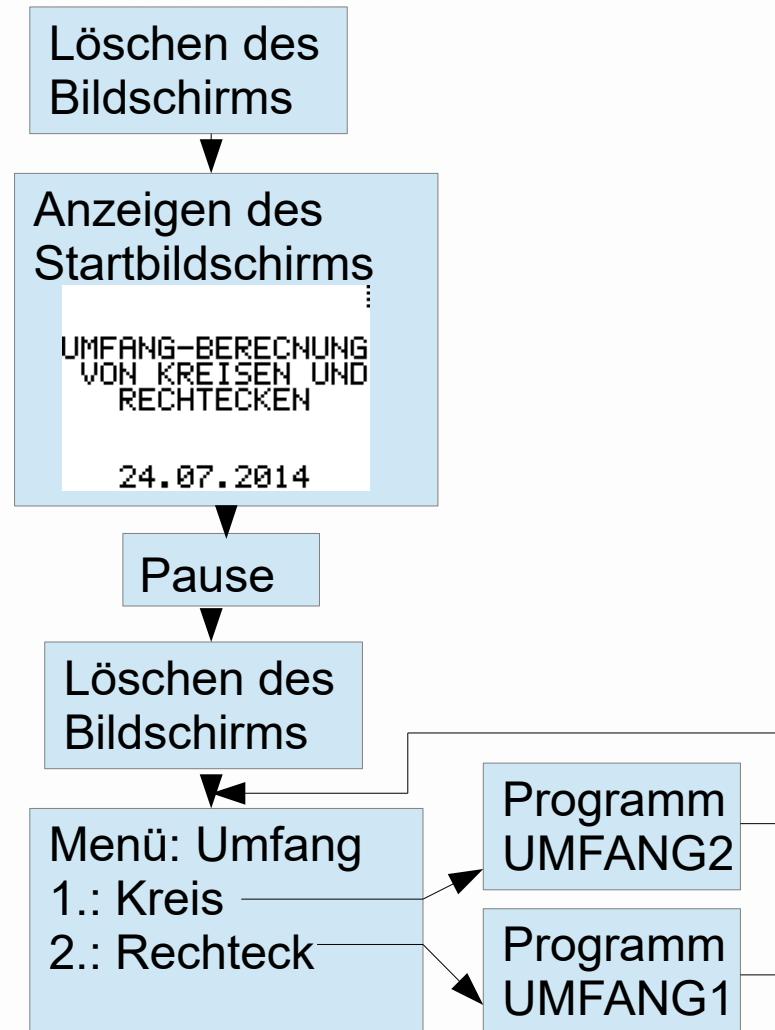
- PRGM→ALPHA→[E]

```
I/O EXEC  
UfPrgm  
Return  
F:Stop  
G:DelVar  
H:GraphStyle<  
I:OpenLib<  
J:ExecLib
```

- Beendet das Unterprogramm.

Das Programm läuft nun an dem Befehl im Hauptprogramm weiter, der nach dem Aufrufen des Unterprogramms steht.

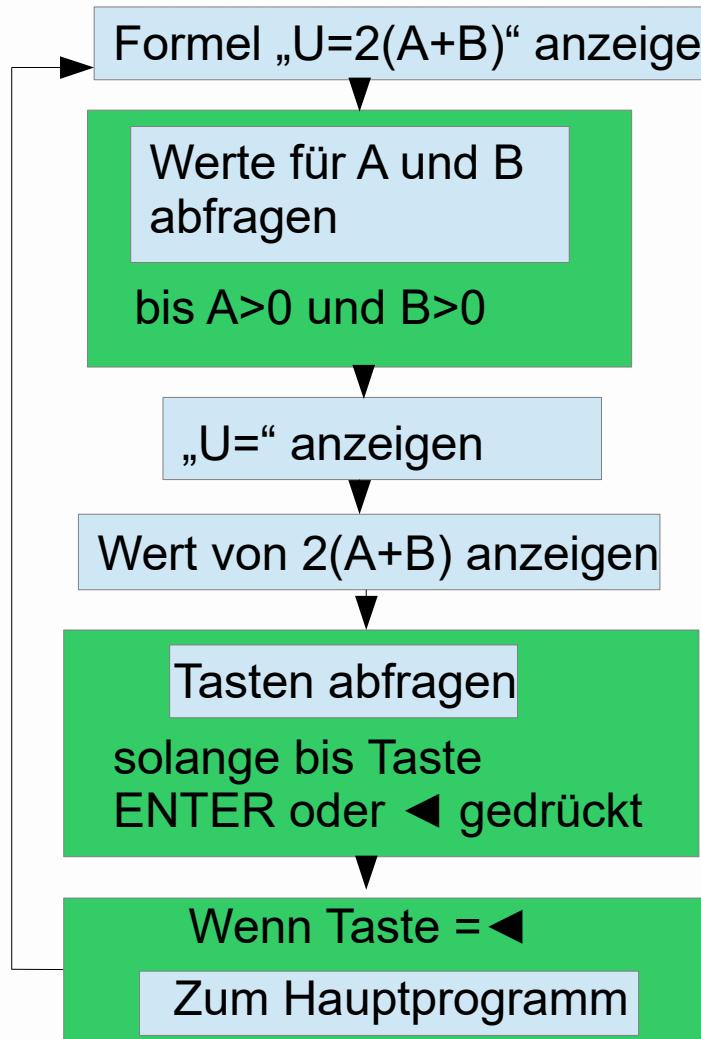
Umfang-Berechnung von Kreisen und Rechtecken



PROGRAM: UMFANG

```
:ClrHome
:Output(3,1,"UMFANG-BERECHNUNG")
:Output(4,2,"VON KREISEN UND")
:Output(5,4,"RECHTECHK
EN")
:Output(8,4,"24.07.2014")
:Pause
:Lbl M
:ClrHome
:Menu(„UMFANG“, „KREIS“, U1,
„RECHTECK“, U2)
:Lbl U1
:prgm UMFANG2
:Goto M
:Lbl U2
:prgm UMFANG1
:Goto M
```

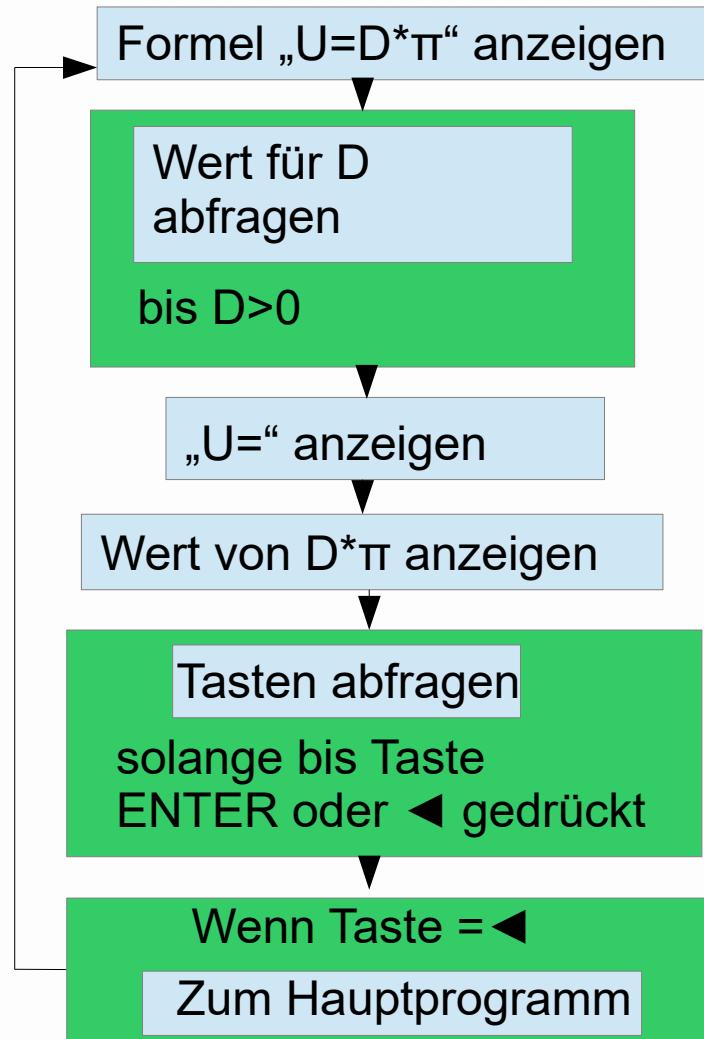
Umfang-Berechnung eines Rechtecks



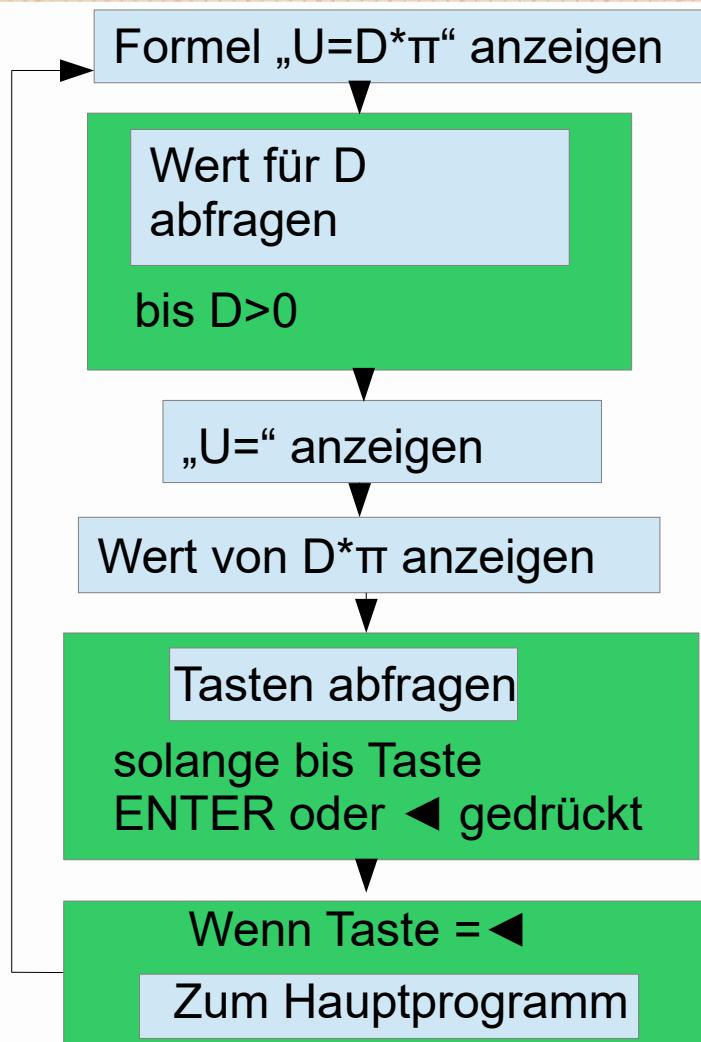
PROGRAM: UMFANG1

```
:Lbl 1
:Disp „U=2(A+B)“
:Repeat A>0 and B>0
:Prompt A,B
:End
:Disp „U=“
:Disp 2(A+B)
:Repeat K=105 or K=24
:getKey →K
:End
:If K=24
:Return
:Goto 1
```

Umfang-Berechnung eines Kreises



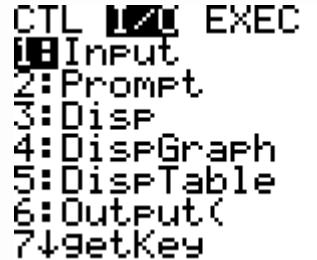
Umfang-Berechnung eines Kreises



PROGRAM: UMFANG1

```
:Lbl 1
:Disp „U=D*π“
:Repeat D>0
:Prompt D
:End
:Disp „U=“
:Disp D*π
:Repeat K=105 or K=24
:getKey →K
:End
:If K=24
:Return
:Goto 1
```

Der Input-Befehl

- PRGM→ENTER 
 - :Input [Text,]Variable
 - zeigt einen bis zu 16 Zeichen langen *Text* an und fragt den Wert einer *Variablen* an.

Der If-Then-End-Befehl

- PRGM → 2 [=Then]

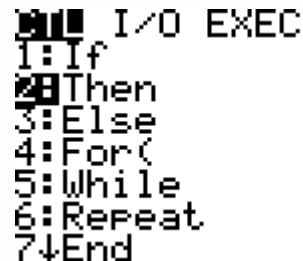
:If *Bedingung*

:Then

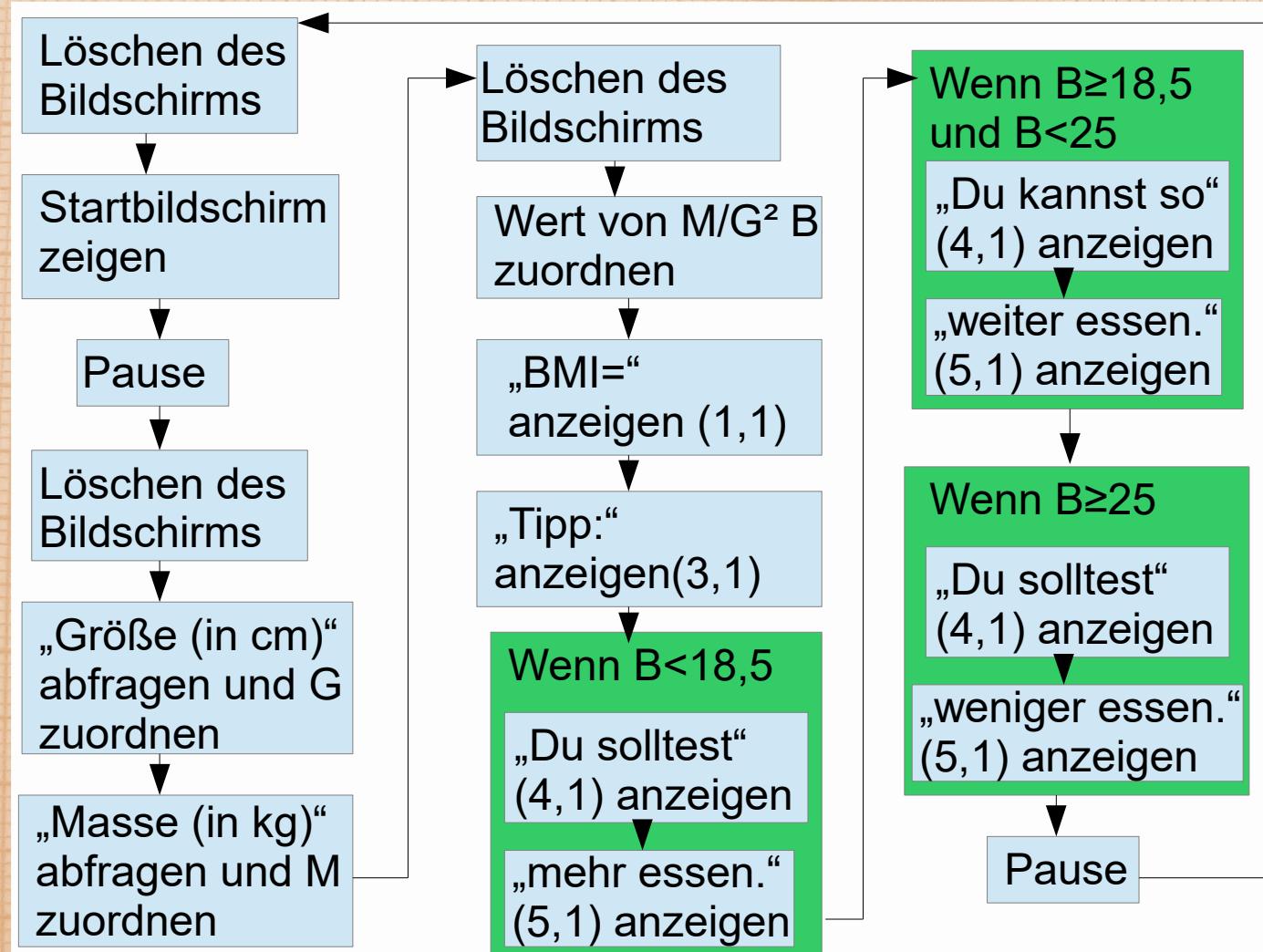
:*Befehlsguppe*

:End

- Führt eine **Befehlsgruppe**, die zwischen Then und End steht, aus, wenn die nach If stehende Bedingung wahr ist.



BMI



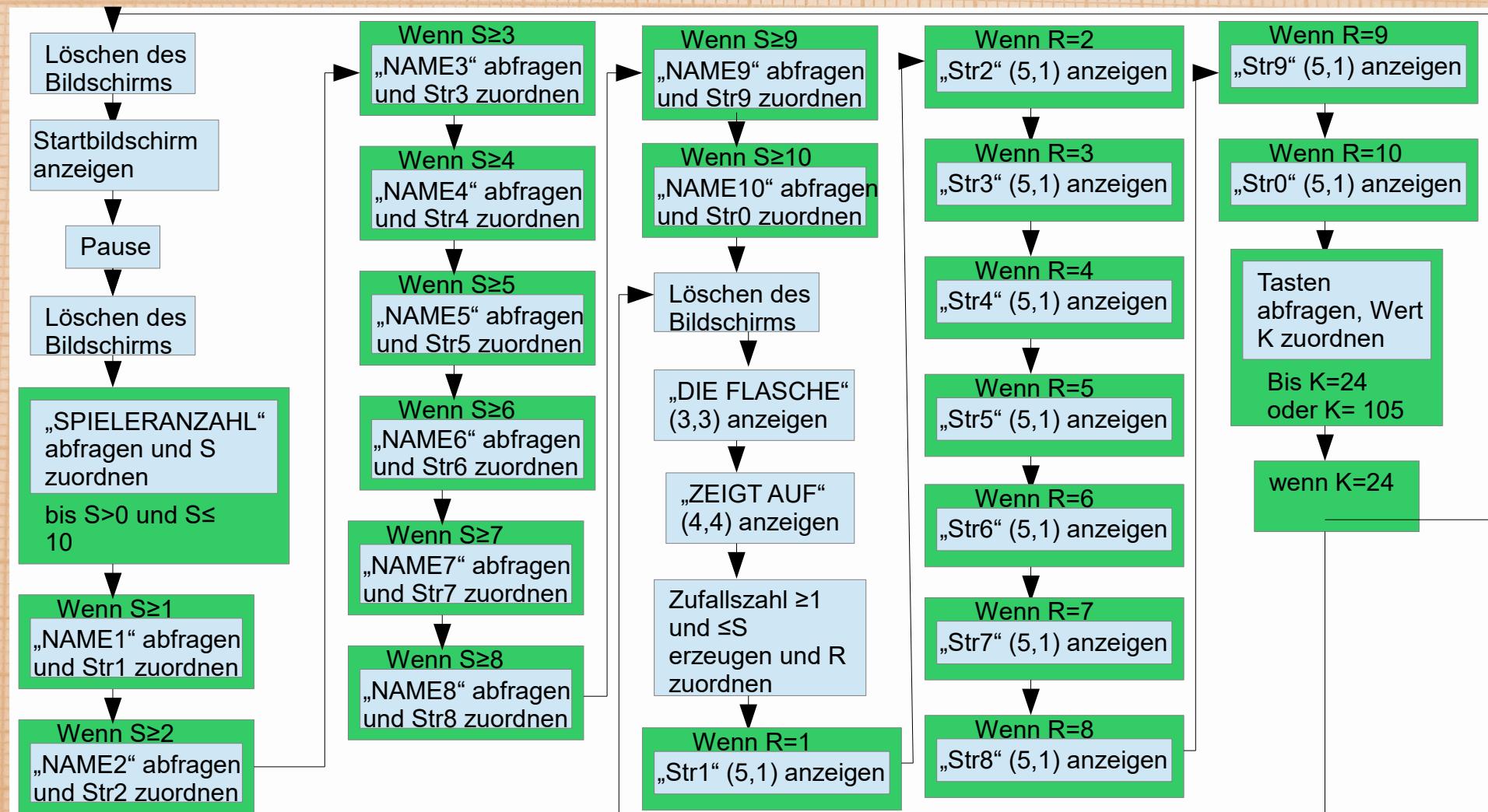
BMI

```
:ClrHome  
:Output(4,2,"BMI-BERECHNUNG")  
:Pause  
:Lbl1  
:ClrHome  
:Input „GROESSE (IN M)“,G  
:Input „MASSE (IN KG)“,M  
:ClrHome  
:M/G2→B  
:Output(1,1,"BMI=")  
:Output(1,5,B)  
:Output(3,1,"TIPP:")  
:If B<18,5  
:Then  
    :Output(4,1,"DU SOLLTEST")  
    :Output(5,1,"MEHR ESSEN.")  
    :End  
    :If B≥18,5 and B<25  
    :Then  
        :Output(4,1,"DU KANNST SO")  
        :Output(5,1,"WEITER ESSEN")  
        :End  
        :If B≥25  
        :Output(4,1,"DU SOLLTEST")  
        :Output(5,1,"WENIGER ESSEN.")  
        :End  
    :Pause  
    :Goto 1
```

Die randInt-Funktion

- MATH → ↺ → 5
- `randInt(untere Grenze,obere Grenze)`
- erzeugt ganze Zufallswerte \geq untere Grenze und \leq obere Grenze.

Flaschendrehen



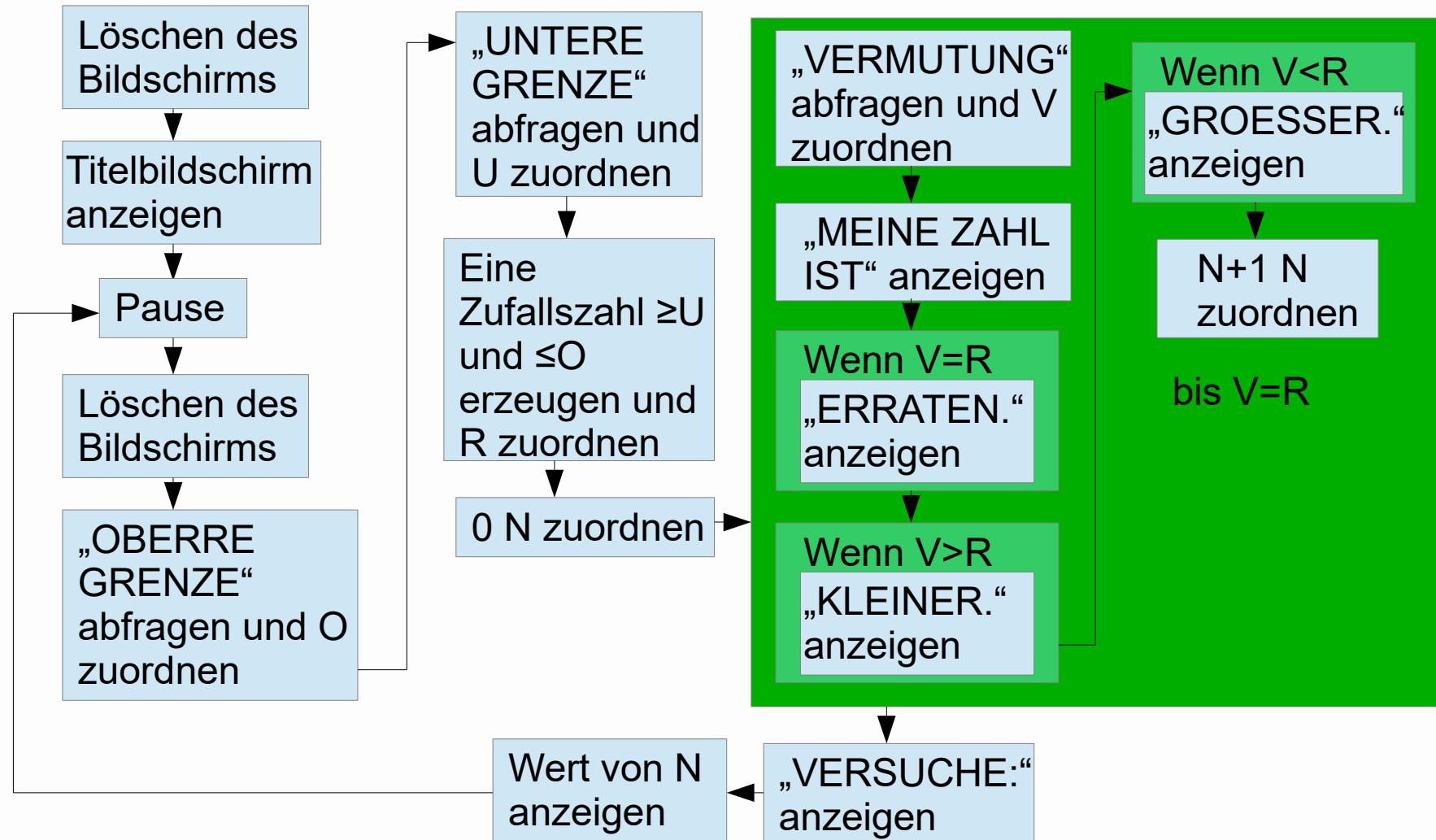
Flaschendrehen

```
:Lbl 1
ClrHome
:Output(4,2,"FLASCHENDRE
HEN")
:Pause
:ClrHome
:Repeat S≥0 und S≤10
:Input „SPIELERANZAHL“,S
:End
:If S≥1
:Input „NAME1“,Str1
:If S≥2
:Input „NAME2“,Str2
:If S≥3
:Input „NAME3“,Str3
:If S≥4
:Input „NAME4“,Str4
:If S≥5
:Input „NAME5“,Str5
:If S≥6
:Input „NAME6“,Str6
:If S≥7
:Input „NAME7“,Str7
:If S≥8
:Input „NAME8“,Str8
:If S≥9
:Input „NAME9“,Str9
:If S≥10
:Input „NAME10“,Str10
:Lbl 2
:ClrHome
:Output(3,3,"DIE FLASCHE")
:Output(4,4,"AUF")
:randInt(1,S)→R
:If R=1
:Output(5,1,Str1)
:If R=2
:Output(5,1,Str2)
:If R=3
:Output(5,1,Str3)
:If R=4
:Output(5,1,Str4)
:If R=5
:Output(5,1,Str5)
:If R=6
:Output(5,1,Str6)
:If R=7
:Output(5,1,Str7)
:If R=8
:Output(5,1,Str8)
:If R=9
:Output(5,1,Str9)
:If R=10
:Output(5,1,Str10)
:Repeat K=24 or K=105
:getKey→K
:End
:If K=24
:Goto 1
:Goto 2
```

Aufgabe: Ratespiel

- Zuerst erscheint ein Titelbildschirm, der nach einer Bestätigung durch ENTER verschwindet. Nun werden Zahlenwerte für die obere und die untere Grenze der zu erratenden Zahl abgefragt. Anschließend gibt der Spieler von ihm vermutete Zahlen ein. Hierbei zeigt der GTR nach jedem Versuch an, ob die richtige Zahl größer oder kleiner als der Versuch ist. Wenn die wahre Zahl erraten wurde, beginnt das Programm erneut. (Zusatz: Das Programm soll auch die Anzahl der Versuche anzeigen.)
- **Tipp:** Erstellt vor dem Programmieren zunächst ein Flussdiagramm.

Ratespiel



Ratespiel

```
:ClrHome  
:Output(4,4,"RATESPIEL")  
:Lbl 1  
:Pause  
:ClrHome  
:Input „OBERE GRENZE“,O  
:Input „UNTERE GRENZE“,U  
:randInt(U,O)→R  
:0→N  
:Repeat V=R  
:Input „VERSUCH“,V  
:Disp „MEINE ZAHL IST“  
:If V=R  
:Disp „ERRATEN.“  
:If V>R  
:Disp „KLEINER.“  
:If V<R  
:Disp „GROESSER.“  
:N+1→N  
:End  
:Disp „VERSUCHE:“  
:Disp N  
:Goto 1
```

Aufgaben

1.) Programmiert das Spiel Vier Gewinnt. Der Bildschirm soll während des Spielens so aussehen:

2.) Programmiert eine Spielesammlung: Das Programm öffnet nach Anzeigen des Startbildschirms ein Menü, in dem man alle bisher programmierten Spiele auswählen kann.

3.) Programmiert die Spiele so um, dass man durch Drücken der MODE-Taste wieder zum Menü der Spielesammlung zurückkehren kann.