

# Grundlagen der Programmierung

Dr. Tom Kamphans

I. Vorlesung

04.04.2018

# Organisatorisches

- Vorlesung:  
Mittwochs 14:00 – 15:30, Raum F 201
- Übung:  
Mittwochs 15:45 – 19:00, Raum F 225
- Übung: alle zwei Wochen
- Folien:  
<http://home.htw-berlin.de/~kamphans>
- Email:  
[tom.kamphans@htw-berlin.de](mailto:tom.kamphans@htw-berlin.de)

# Organisatorisches

- Bestehen des Moduls:
  - Erfolgreiche Teilnahme an den Übungen
    - Anwesenheit
    - Mitarbeit / Aufgaben erfolgreich absolviert

# KLAUSUR

- 1. Zeitraum:  
Vorschlag: 25.07.2018
- 2. Zeitraum: 24.09. bis 08.10.2018

# Fragen

- Wer hat noch NIE...
  - ... einen Computer bedient?
  - ... Email benutzt?
  - ... im Web „gesurft“?
  - ... eine Textverarbeitung benutzt?
  - ... mit Windows 10 gearbeitet?
  - ... mit MacOS gearbeitet?
  - ... mit Linux gearbeitet?

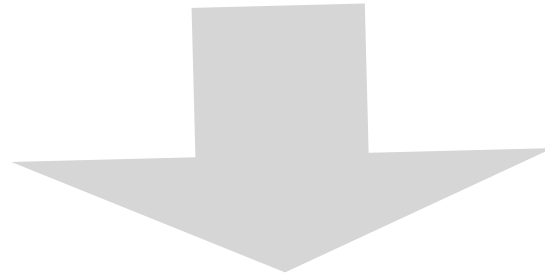
# Fragen

- Wer
  - hat schon einmal programmiert?
  - ... in C?
  - ... in Matlab?
  - ... in C++ oder Java?
  - weiß, was ein Algorithmus ist?
  - kennt das P/NP Problem?

# Was heißt Programmieren?

# Programmieren

Problem



Algorithmus



Implementierung

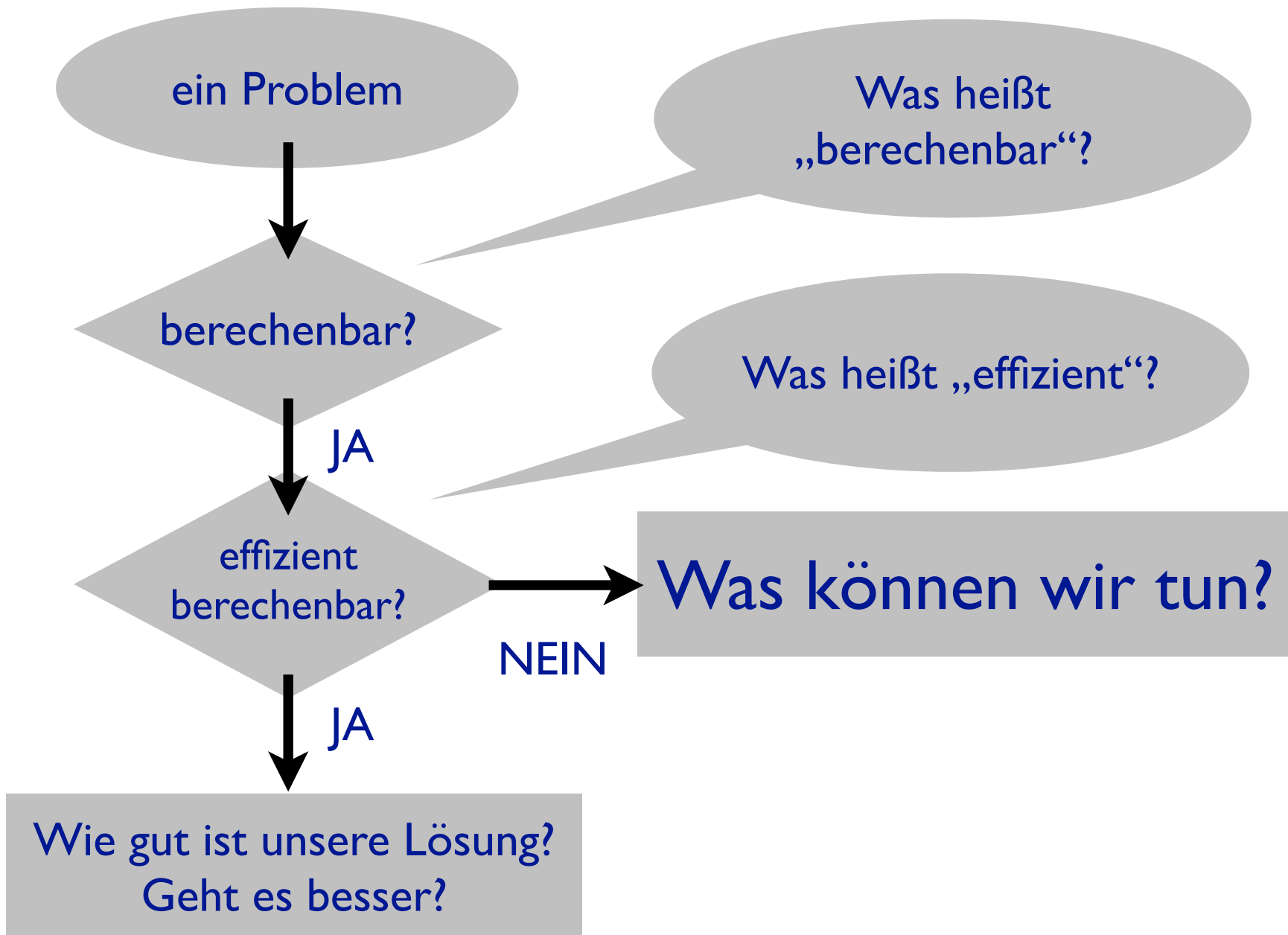
Programm

# Algorithmus

- Ein Algorithmus ist eine aus endlich vielen Schritten bestehende eindeutige Handlungsvorschrift zur Lösung eines Problems oder einer Klasse von Problemen. (Wikipedia)
- Beispiel:
  - nimm Bockwurst aus der Dose
  - koche Wasser
  - lasse Bockwurst 5 Minuten im Wasser ziehen

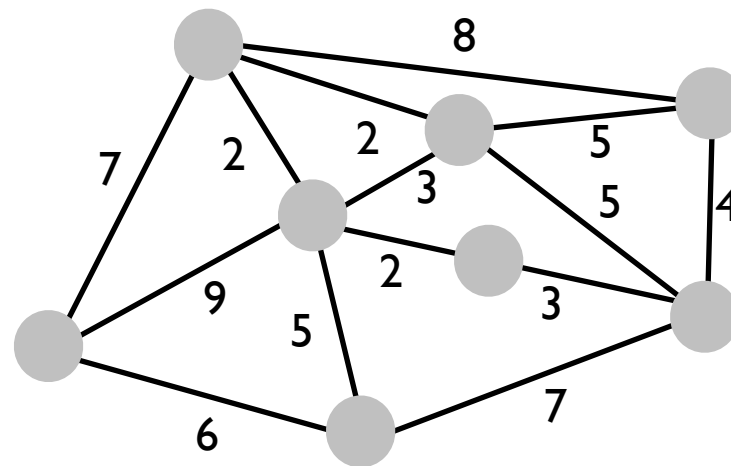


# Vom Problem zum Algorithmus



# Effizienz

- Gegeben ein Graph  
(z.B.: Straßennetz mit Orten und Entfernungen)
- Finde kürzeste Rundtour, die alle Orte besucht



# Effizienz

- Lösung:
  - berechne alle möglichen Touren
  - wähle die billigste aus
- kann sehr lange dauern
  - nicht effizient!
- nicht effizient berechenbar?
- man weiß nicht, ob eine effiziente Lösung existiert!
- Vermutung: existiert nicht

# Wie beschreibt man Algorithmen?

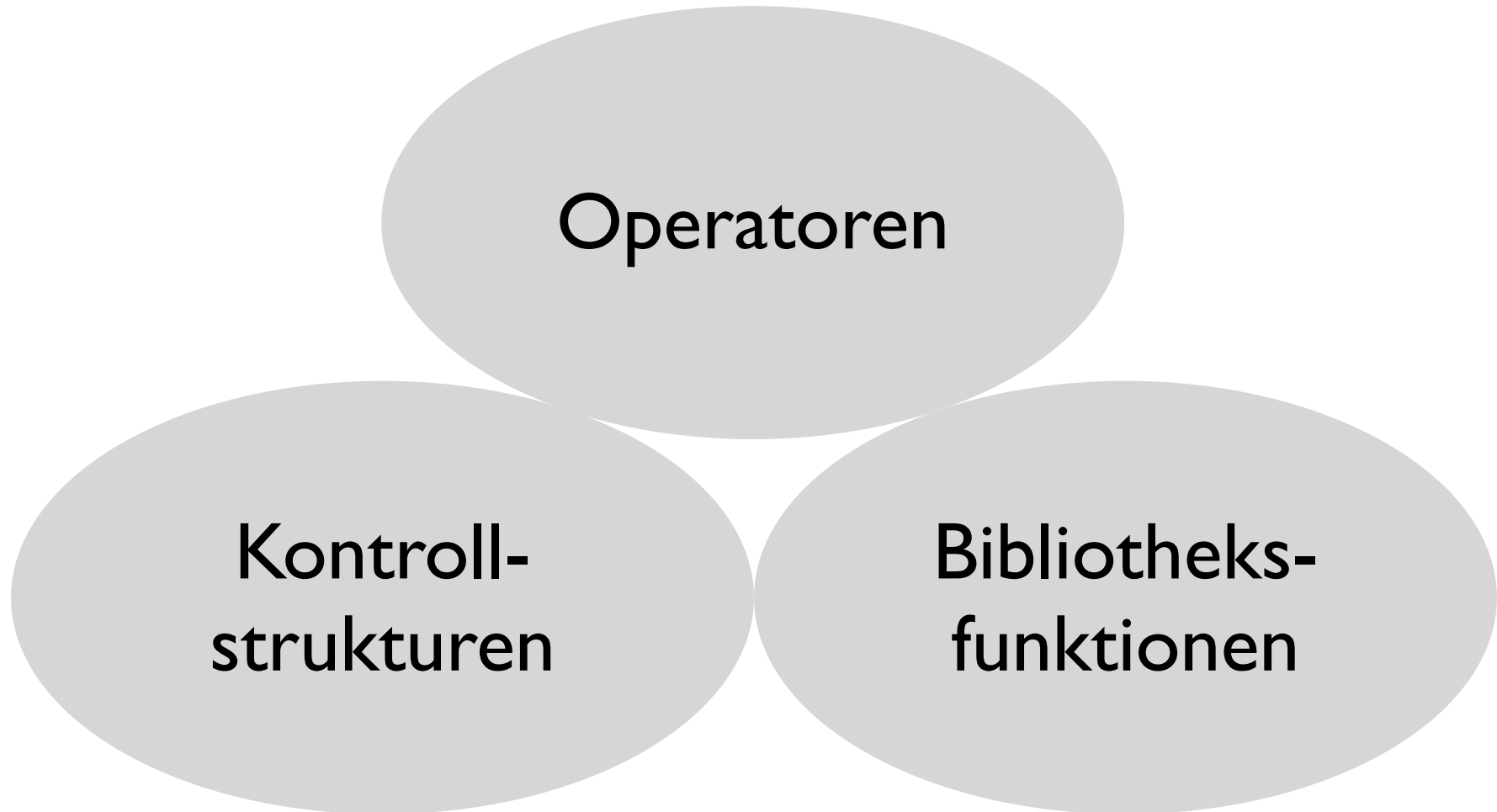
- Programmcode:

```
max=0;  
for(i=0; i<N; i++)  
    if ( a[i] > max ) max=a[i];
```

- Pseudocode:  
*durchlaufe Array a, bestimme das maximale Element*
- auch möglich: Flußdiagramme
- Einzige Regel:  
es muss klar werden, was gemacht werden soll
- Technik für Algorithmenentwurf:  
Schrittweise Verfeinerung

# Vom Algorithmus zum Programm

- Elemente von Programmiersprachen



# Beispiel: Turtlegraphik

- Turtlegraphik: bewegt Zeiger ("Turtle") über den Bildschirm, zeichnet Spur des Zeigers
- Befehle:
  - Bewege Zeiger (MOVE *schrittweite*)
  - Drehe Zeiger (TURN *winkel*)
- Problem: Zeichne Quadrat mit Kantenlänge 100

Erstmal

*Snap* (<https://snap.berkeley.edu/>) oder  
*Scratch* (<http://scratch.mit.edu>)

# Quadrat mit Turtlegraphik

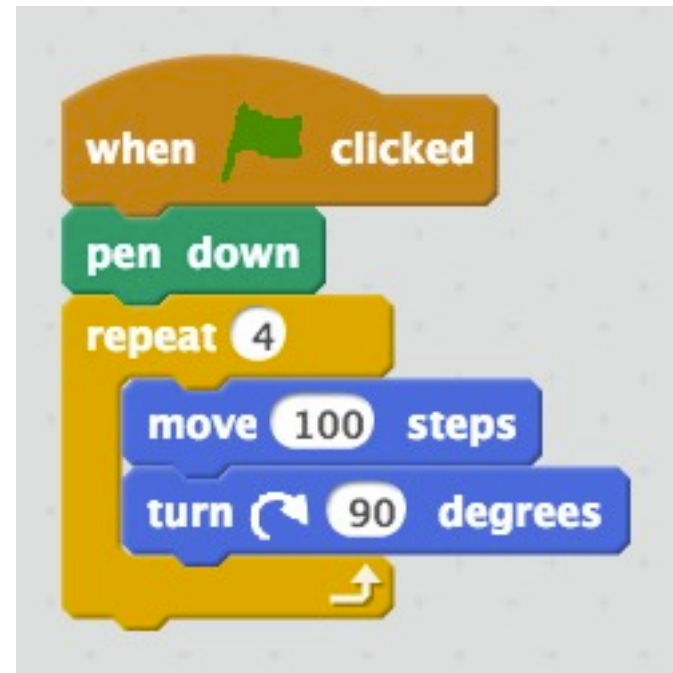
MOVE	100
TURN	90
MOVE	100
TURN	90
MOVE	100
TURN	90
MOVE	100
TURN	90

Befehl      Parameter



# Geht es einfacher?

Wiederhole 4 mal:  
MOVE 100  
TURN 90





# Kontrollstruktur!



"Klammer"  
für Befehle

auch selbst  
Befehl!

# Neue Befehle

Snap:

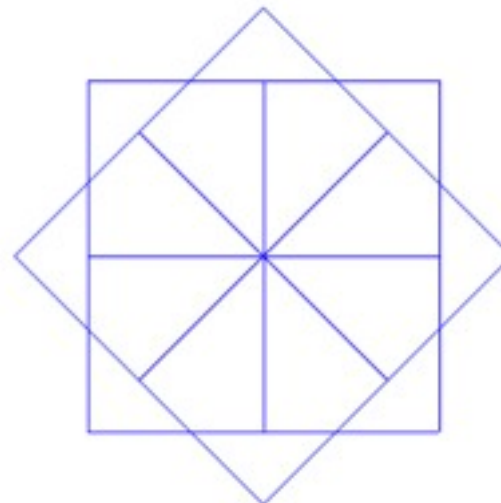
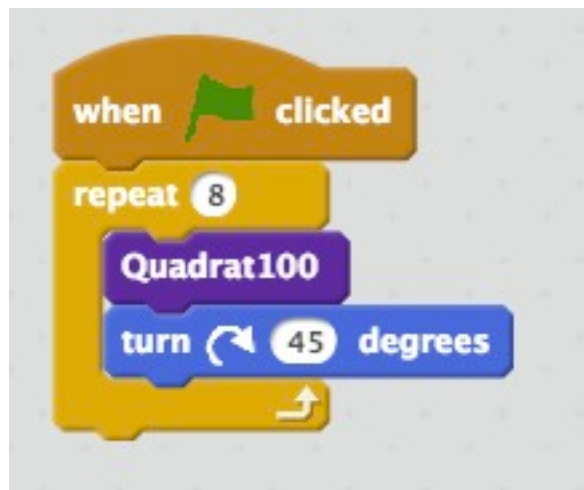


Scratch:

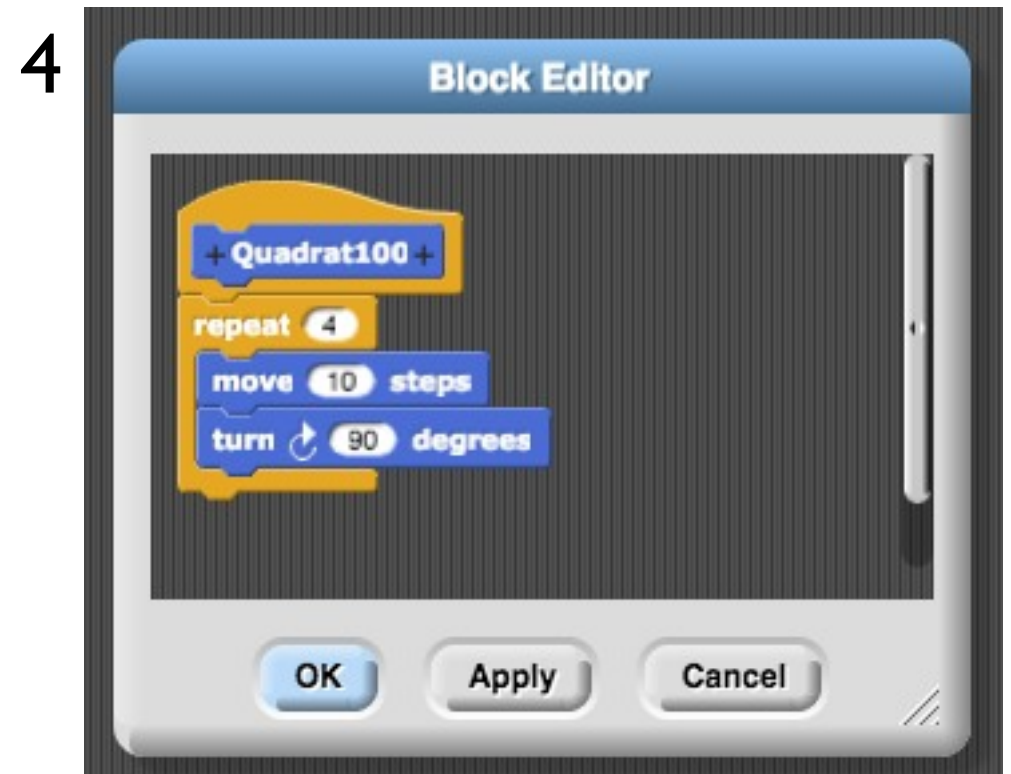


definiert neuen Befehl

Anwendung:

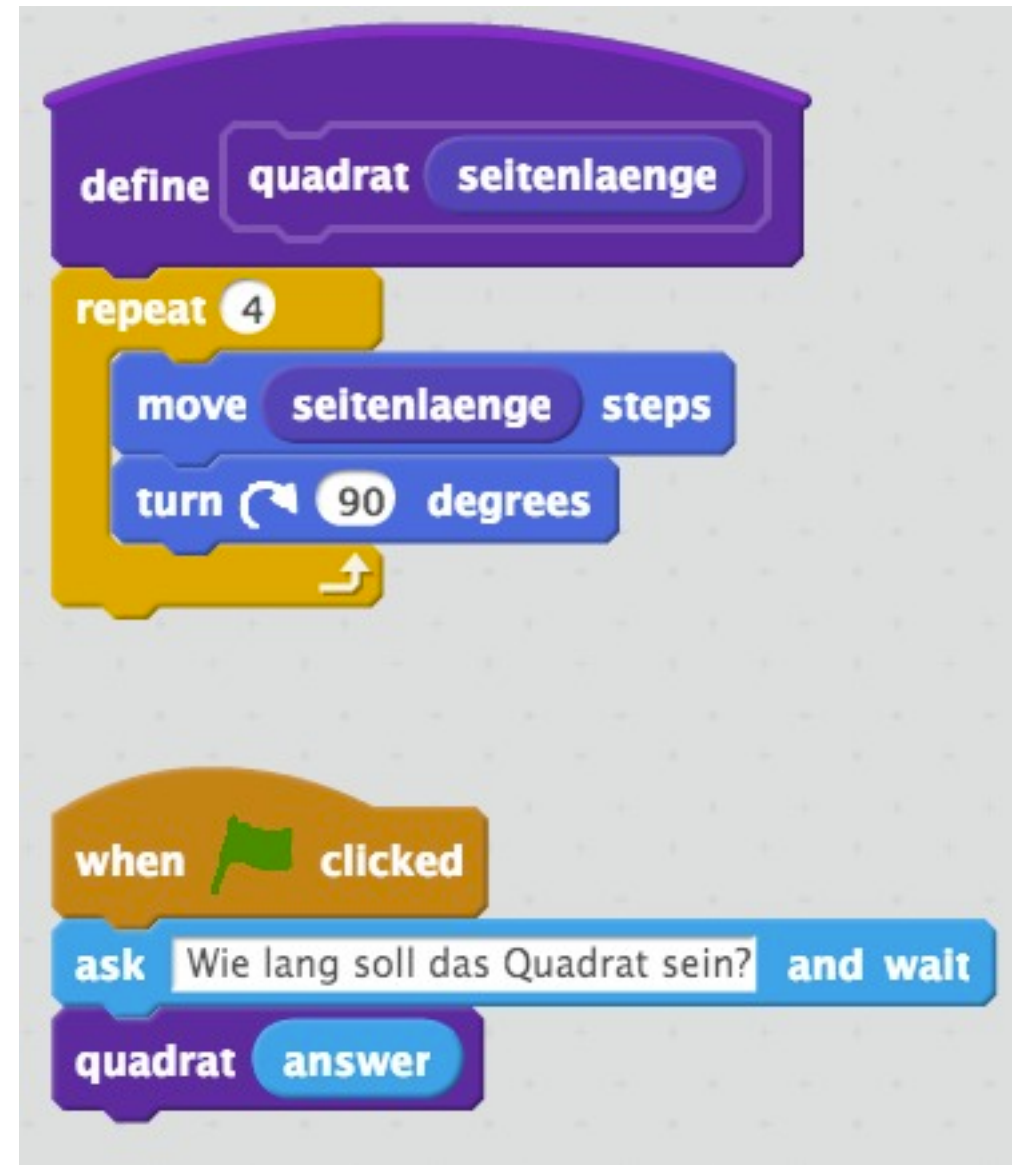


# Neue Befehle (in Snap)

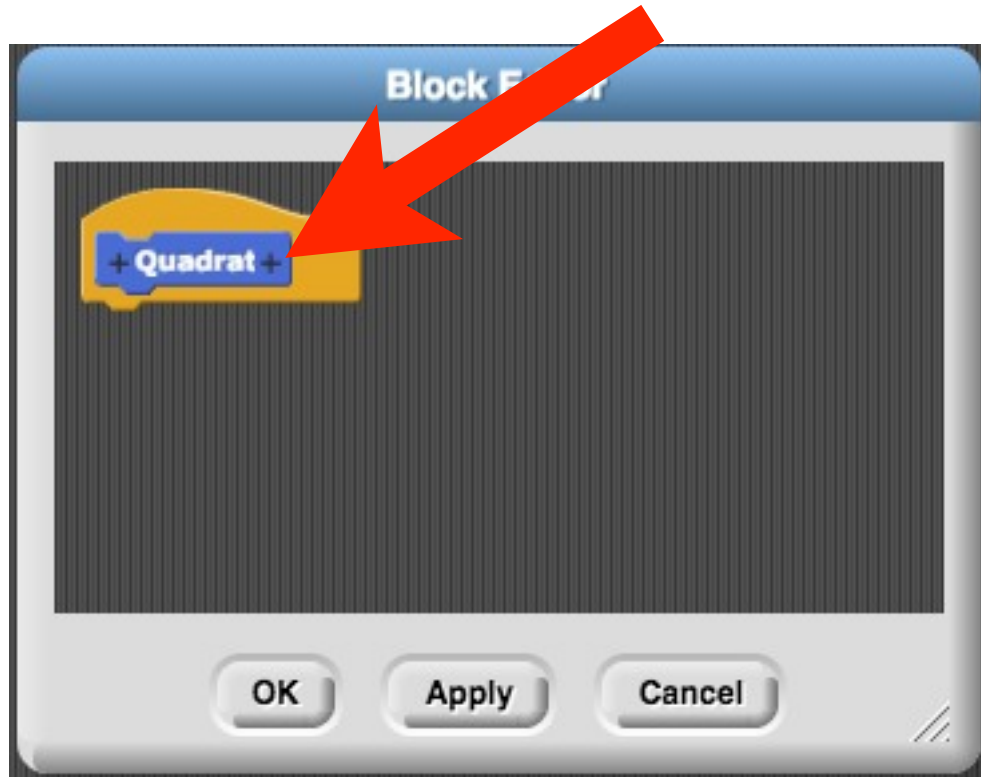


# Neue Befehle

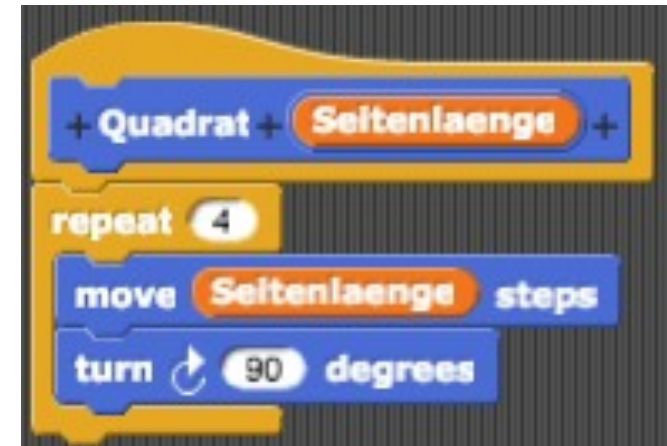
- Ziel: Befehl zum Zeichnen eines Quadrates definieren, unabhängig von der Seitenlänge
- Die Seitenlänge ist ein **Parameter** des neuen Befehls *Quadrat*
- Neuer Befehl kann genau wie geg. Befehl verwendet werden



# Neue Befehle mit Parameter



Hinteres '+' klicken



Ändern:  
rechte Maustaste

