

# Algorithmus

# Definition Algorithmus

- Ein Algorithmus ist eine **eindeutige Handlungsvorschrift** zur Lösung eines Problems oder einer Klasse von Problemen. Algorithmen bestehen aus **endlich vielen**, wohldefinierten **Einzelschritten**.



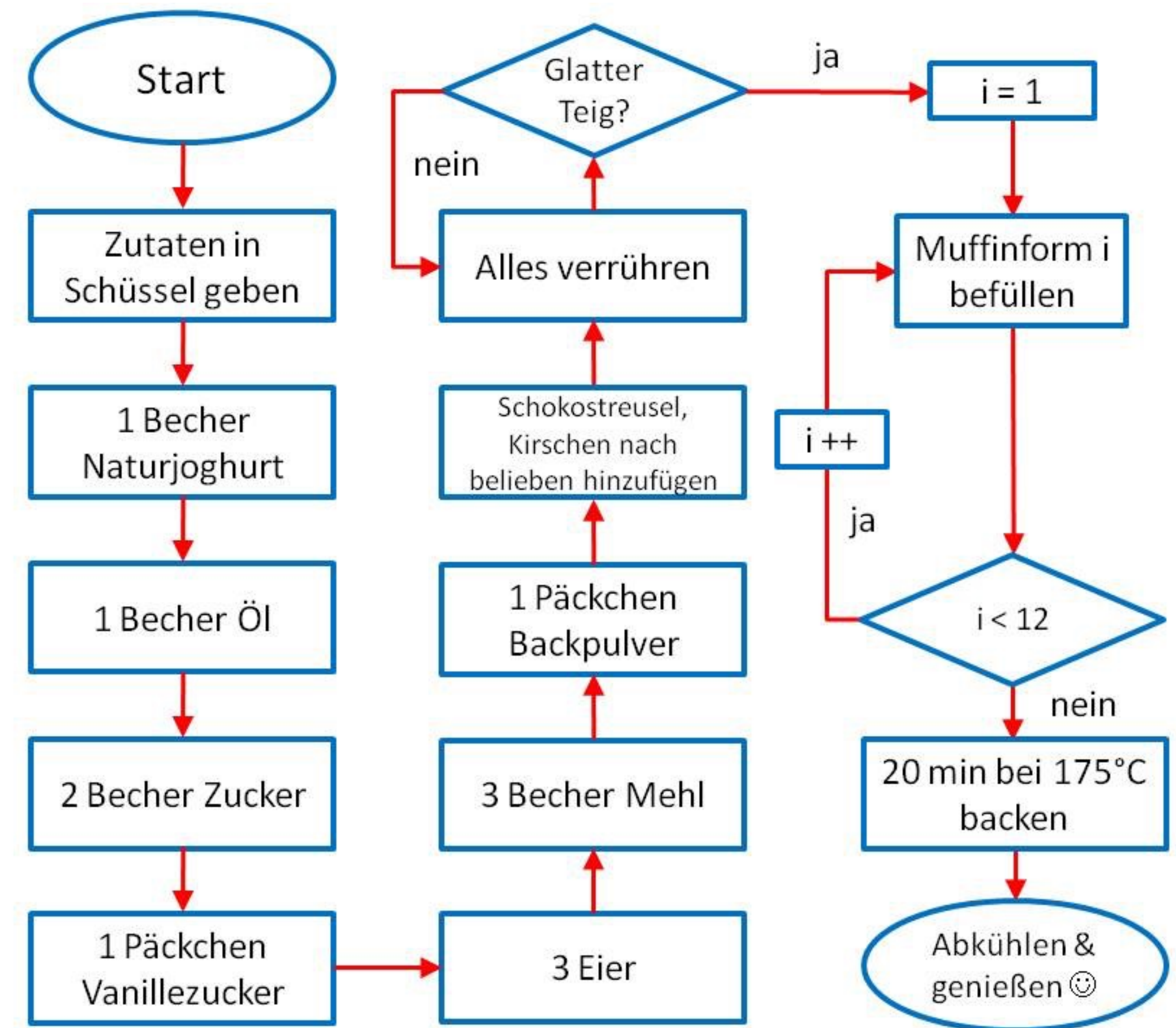
- Der Begriff ist eine Abwandlung des Namens des persischen Rechenmeisters und Astronomen Abu Dscha'far Muhammad ibn Musa **al-Chwārizmī**

# Algorithmus

- Ein Algorithmus ist eine **Handlungsvorschrift**
  - Gibt an, **welche Schritte** zur Erreichung eines Ziels wann durchgeführt werden müssen
  - Gibt an, mit **welchen Objekten** operiert werden muss

# Beispiel: Kochrezept

- Operationen auf / mit den Objekten werden beschrieben
- Algorithmen sprechen eine Sprache, die die ausführende Maschine versteht
- Menschen wissen in der Regel, was Becher und Schüsseln sind und wie man verrührt oder befüllt

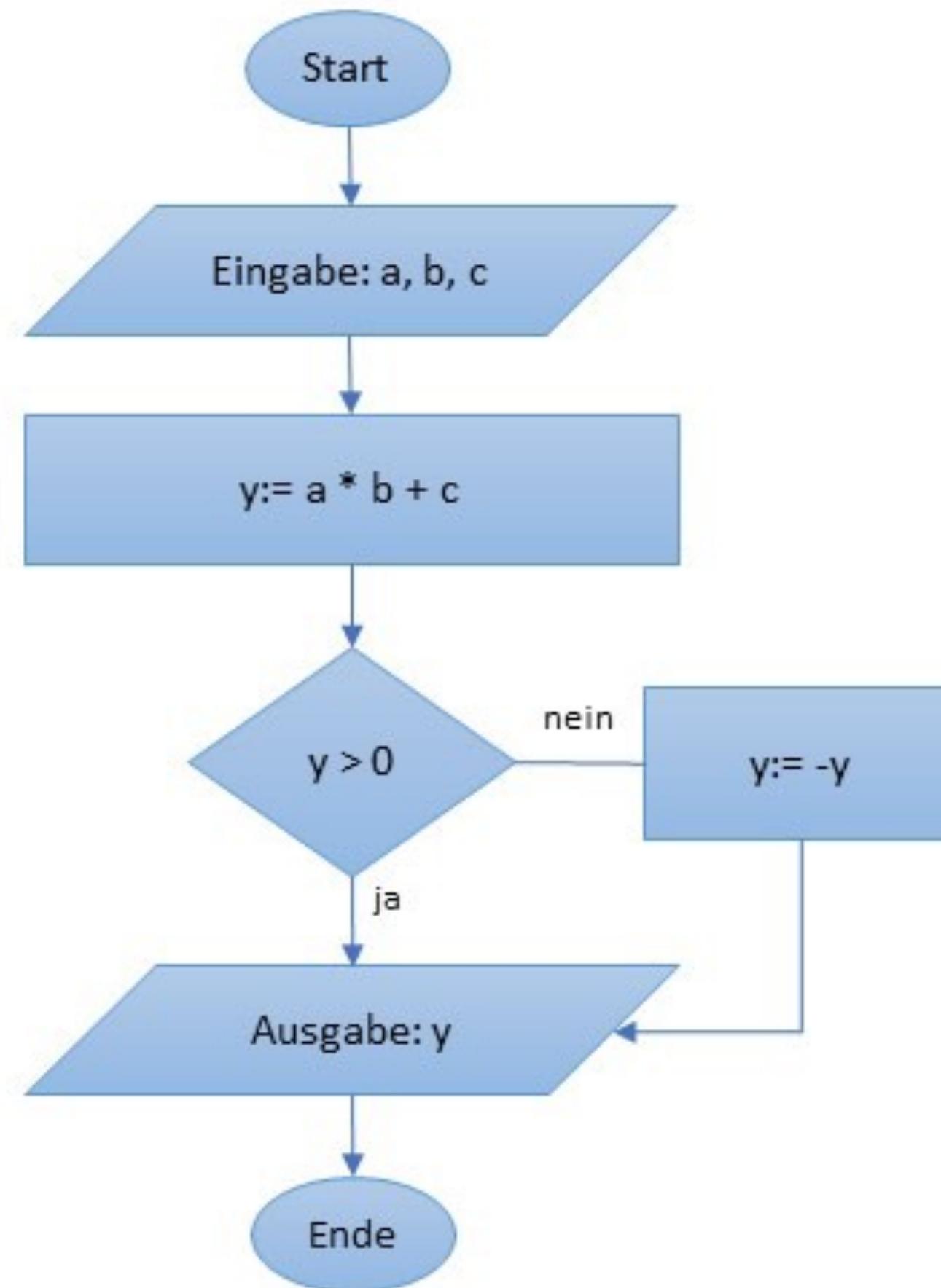




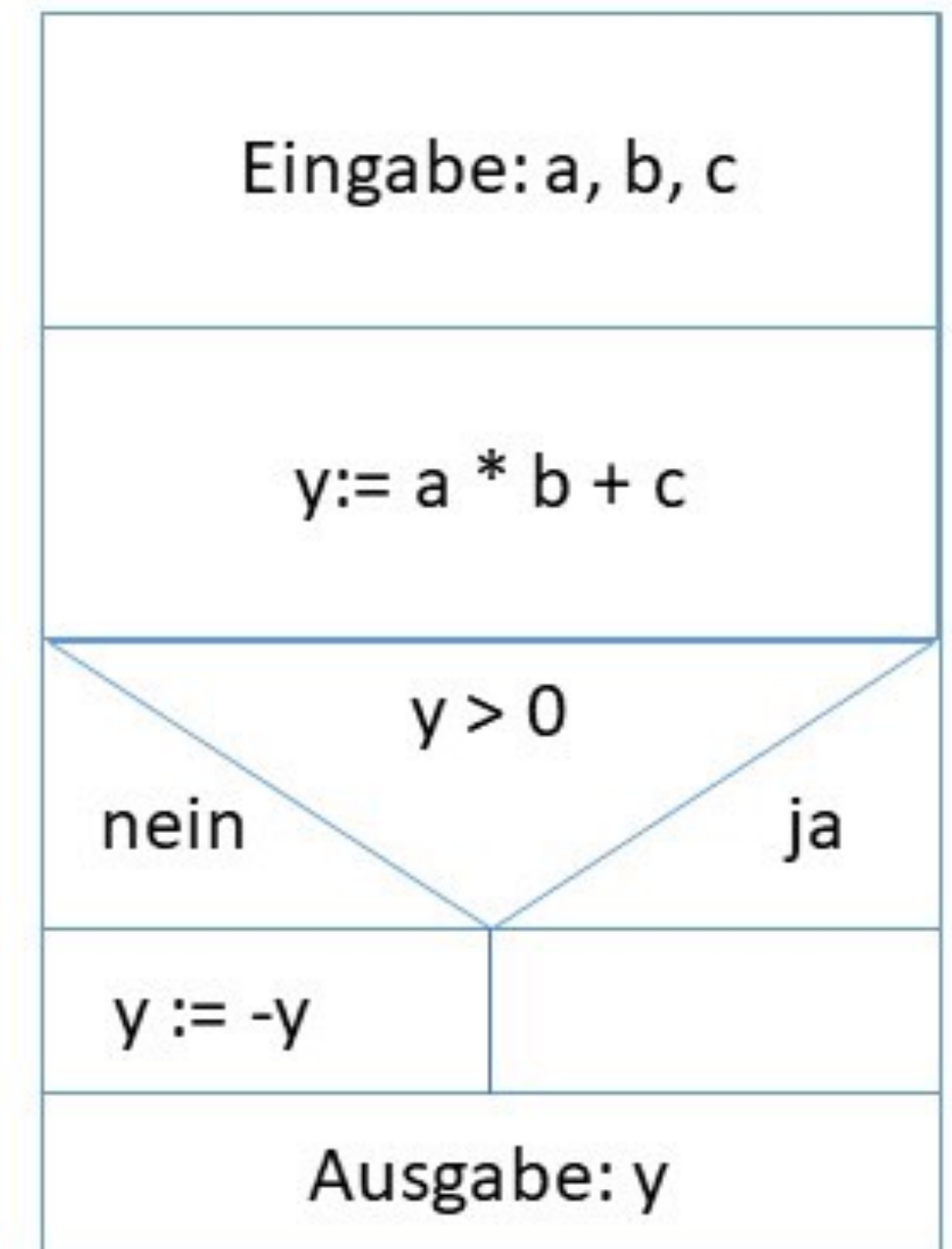
# Beispiel II

- Ablauf immer gleich
- Variation durch Eingaben (a, b, c)
- Algorithmen beschreiben i.d.R. allgemeines Vorgehen

Programmablaufplan



Struktogramm



# Beispiel III

- Beispiel Quadrierung einer Anzahl von Objekten

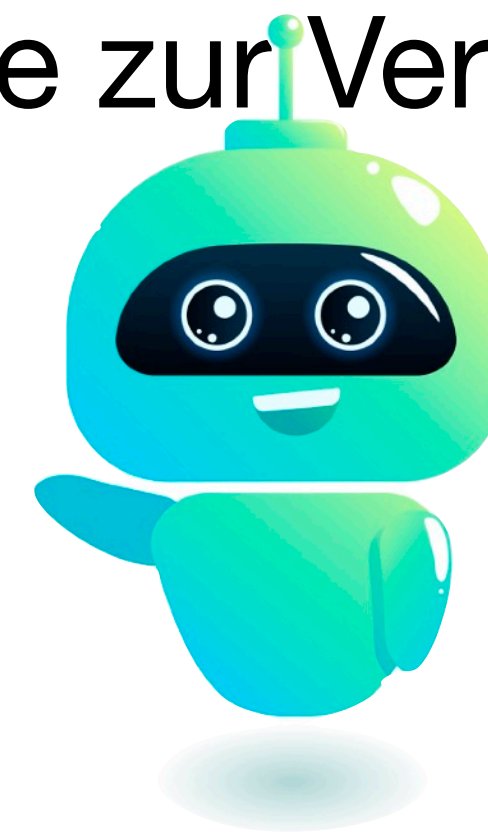


Mathematisch ist *Datenverarbeitung* eine Abbildung

*Datenverarbeitung*:  $N_0 \rightarrow N_0$

**Beispiel:**  $\text{Datenverarbeitung}(x) = x^2$ , mit  $x$  aus den natürlichen Zahlen

- Die ausführende Maschine bestimmt die zur Verfügung stehen Operationen
- Hier: Benutze die Maschine „MUSA“

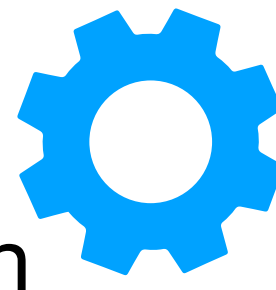


# Beispiel III

- Die Maschine „MUSA“ stellt folgende Operationen zur Verfügung:

- Zählen von Objekten
- Aufnehmen von Objekten
- Weglegen von Objekten
- Lesen und vergleichen von Zahlen
- Schreiben von Zahlen



- Bevorzugte Objekte der Maschine „MUSA“:
- Algorithmus zur Quadrierung einer Anzahl von  muss mit obigen Grundoperationen auskommen

# Beispiel III

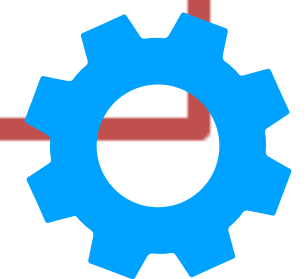
- **Aufgabenstellung:**

- Werden  $x$  Zahnräder übergeben, dann sollen  $x^2$  Zahnräder zurückgegeben werden

- **Problem:**

- Nur die folgenden Operationen stehen zur Verfügung:

- Zählen von Objekten
- Aufnehmen von Objekten
- Weglegen von Objekten



- **Lösung:**

- Mit zählen, aufnehmen und weglegen kann man einfach addieren -> Aufgabenstellung mittels Addition lösen

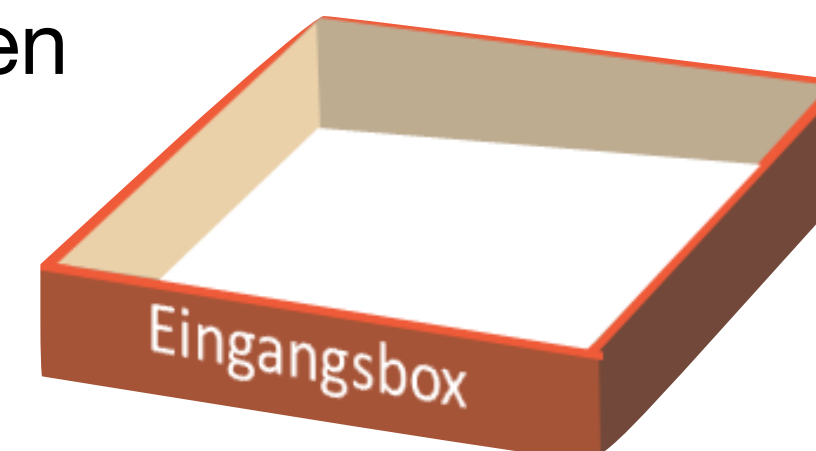
- Algorithmus:

Datenverarbeitung( $x$ ) = wenn ( $x=0$ ) dann 0,  
sonst  $\sum_{i=1, \dots, x} x$



# Beispiel III

- Baue Algorithmus gemäß:  $\text{Datenverarbeitung}(x) = \text{wenn } (x=0) \text{ dann } 0, \text{sonst } \sum_{i=1, \dots, x} x$
- Aufnehmen von Objekten:
  - Woher kommen die Objekte?
- Weglegen von Objekten:
  - Wohin sollen die Objekte gelegt werden?
- Irgendetwas muss Objekte aufnehmen, damit die übergebenen Zahnräder von den Ergebniszahnrädern und etwaigen Zwischenergebnissen nicht durcheinander kommen
- Boxen (Schachteln, Container, ...)

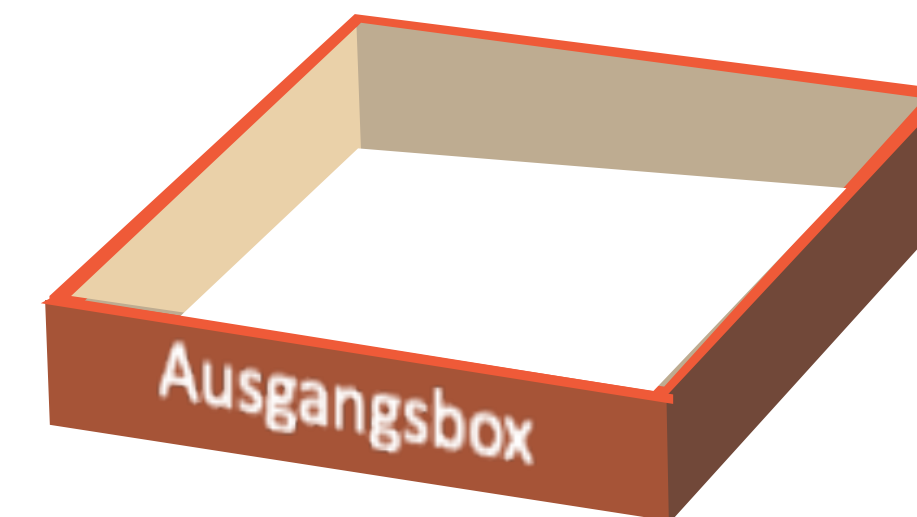
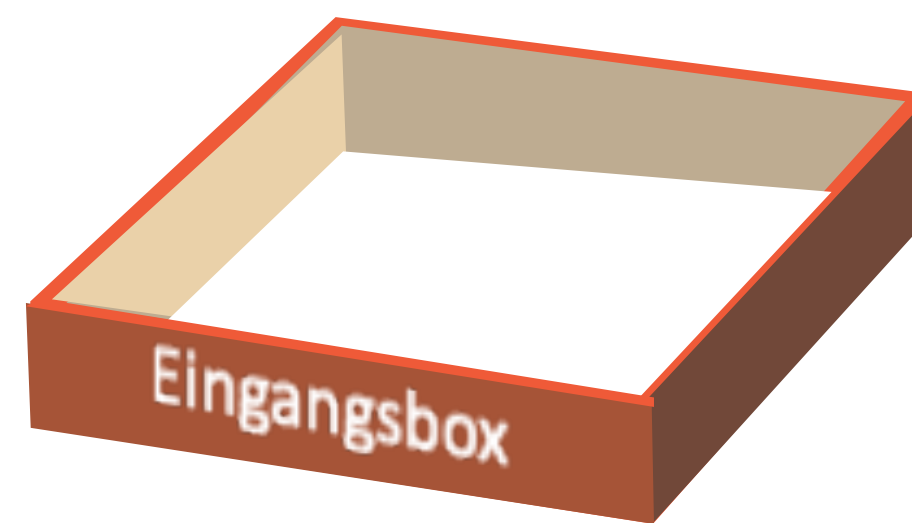


# Beispiel III - Schritt 1

- Baue Algorithmus gemäß:

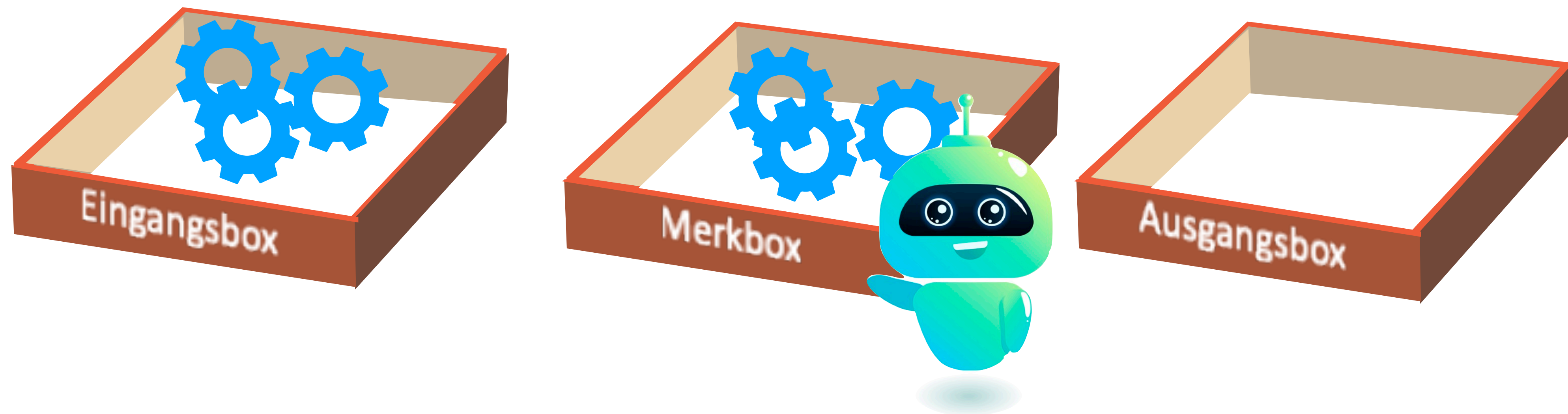
Datenverarbeitung(x) = wenn (x=0) dann 0,  
sonst  $\sum_{i=1, \dots, x} x$

- Daten werden in der Eingangsbox aufgenommen und in der Ausgangsbox abgelegt
- 1. Wenn Eingangsbox leer, dann Ausgangsbox leer. **Ende**



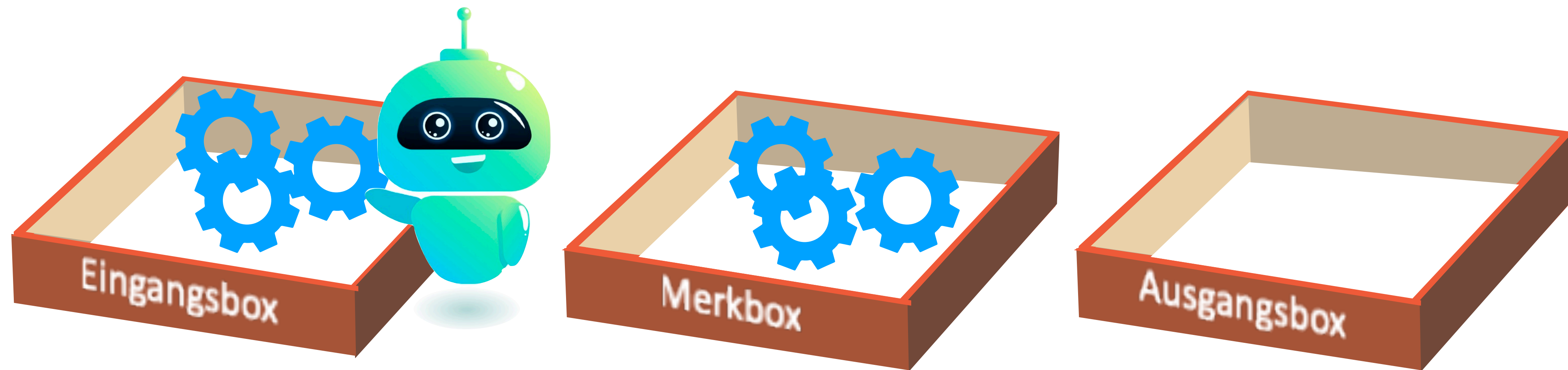
# Beispiel III - Schritt 2

- 2. Wenn Eingangsbox nicht leer, lege so viele Zahnräder in eine Merkbox, wie in der Eingangsbox liegen



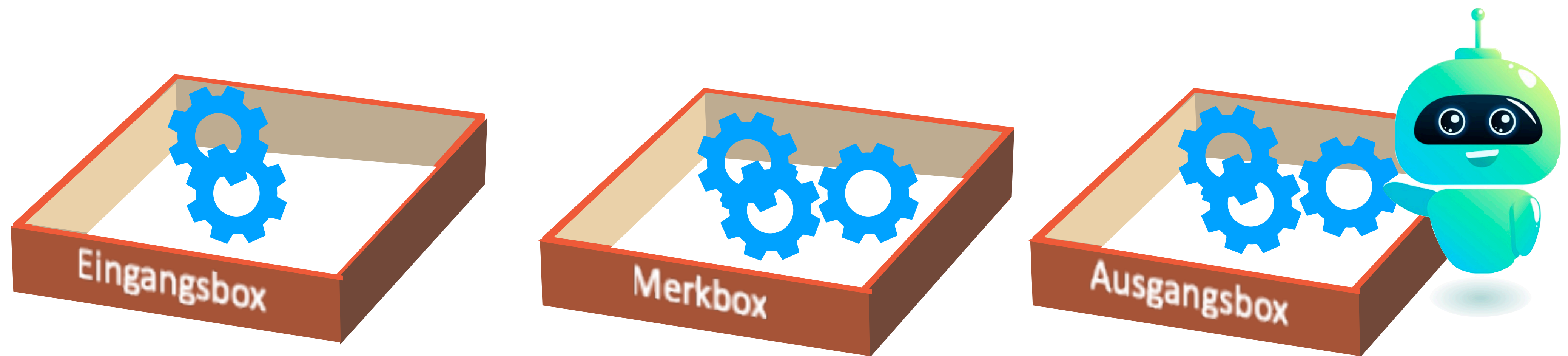
# Beispiel III - Schritt 3

- 3. Solange Eingangsbox nicht leer, lege so viele Zahnräder in die Ausgangsbox, wie in der Merkbox liegen und nehme ein Zahnrad aus der Eingangsbox heraus



# Beispiel III - Schritt 3

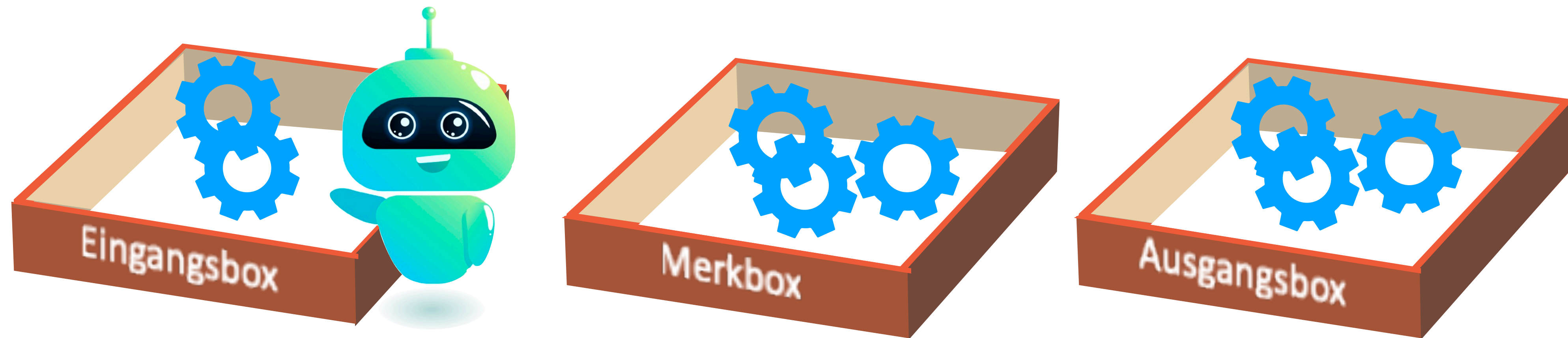
- 3. Solange Eingangsbox nicht leer, lege so viele Zahnräder in die Ausgangsbox, wie in der Merkbox liegen und nehme ein Zahnrad aus der Eingangsbox heraus





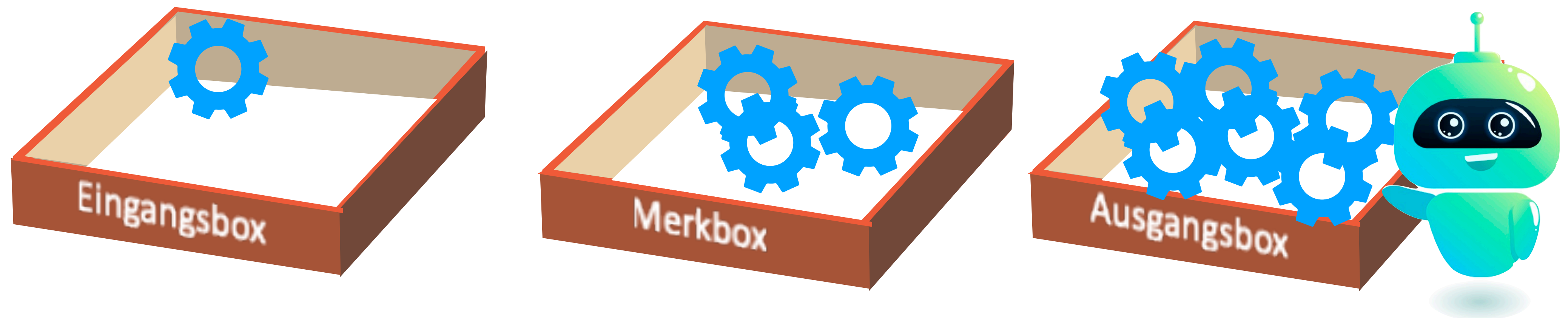
# Beispiel III - Schritt 4

- 3. Solange Eingangsbox nicht leer, lege so viele Zahnräder in die Ausgangsbox, wie in der Merkbox liegen und nehme ein Zahnrad aus der Eingangsbox heraus



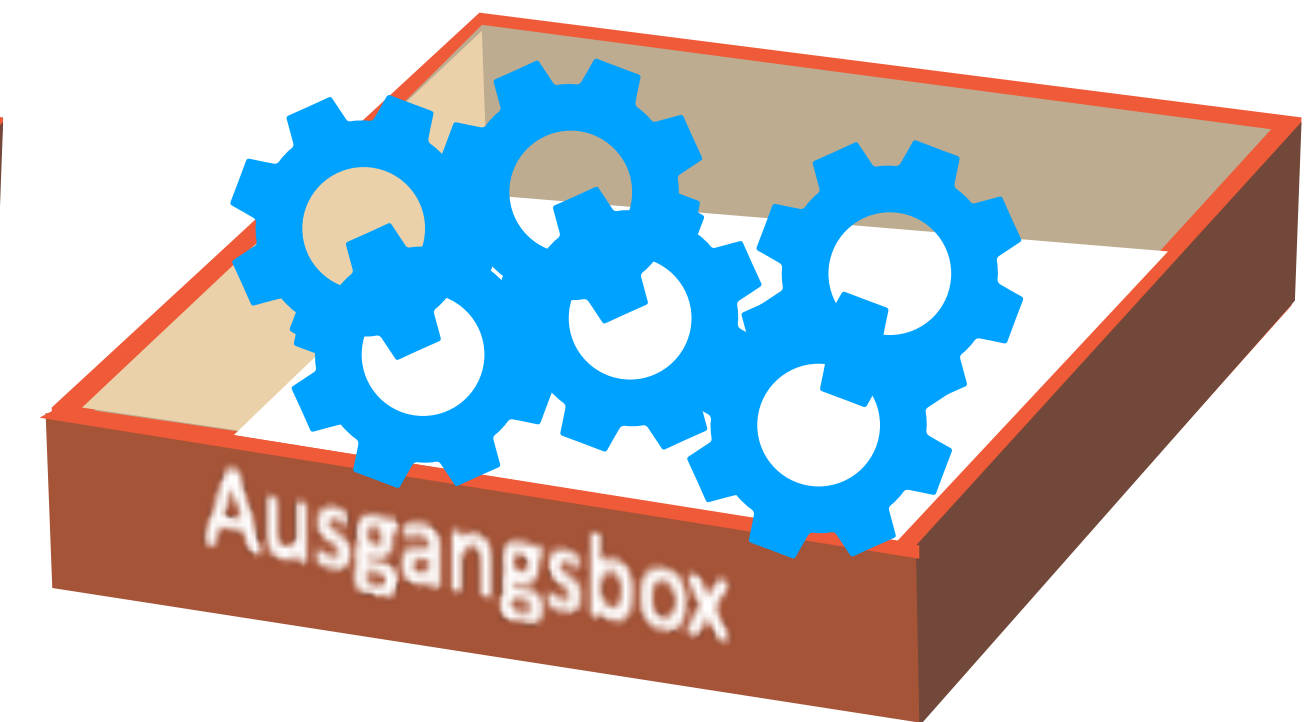
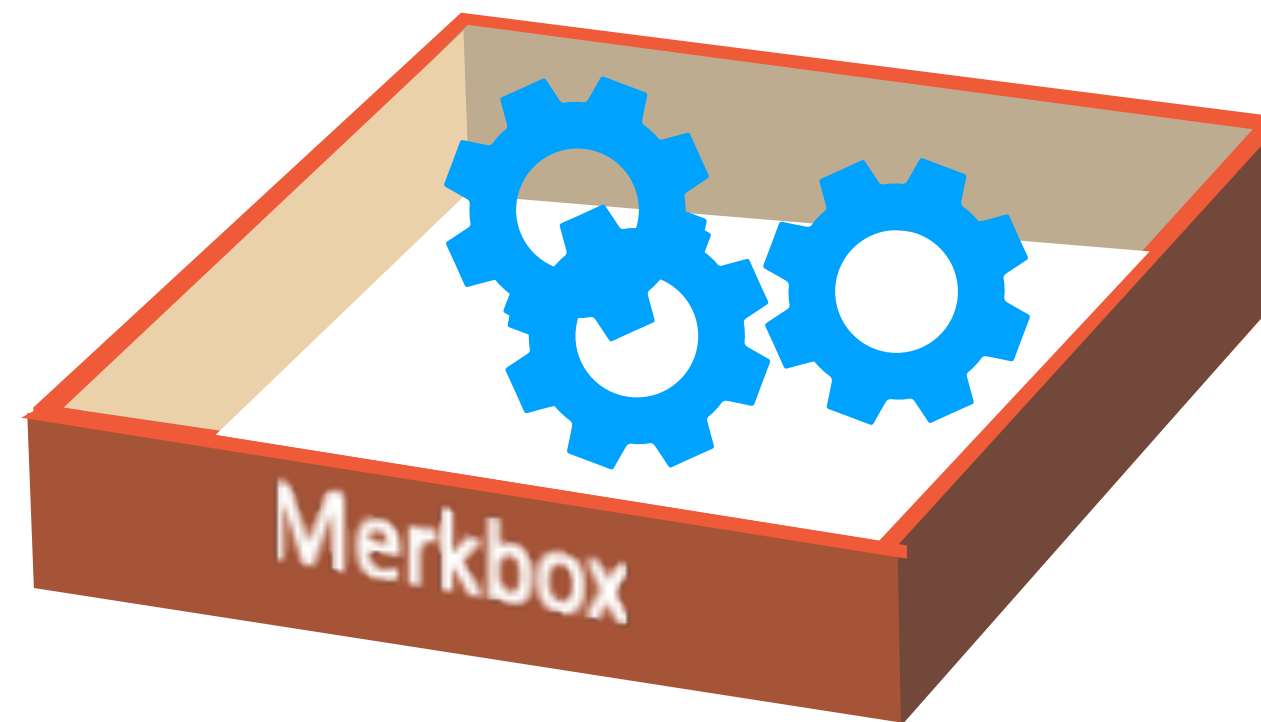
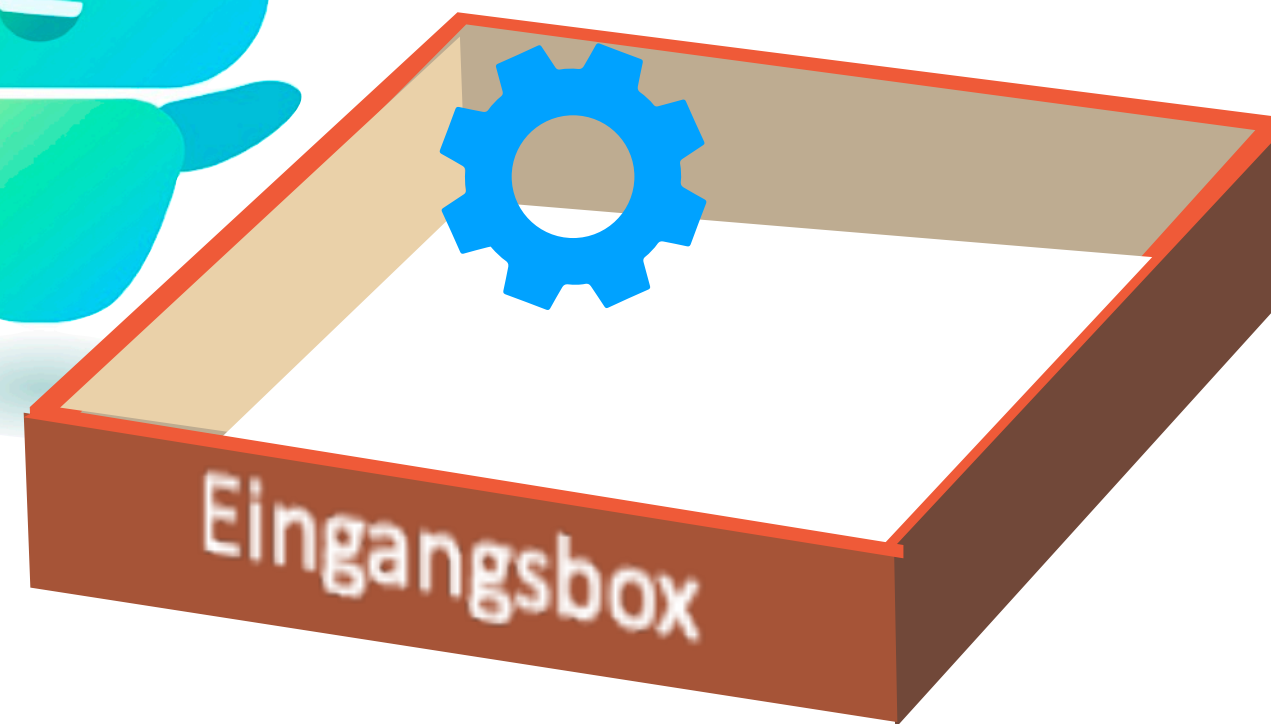
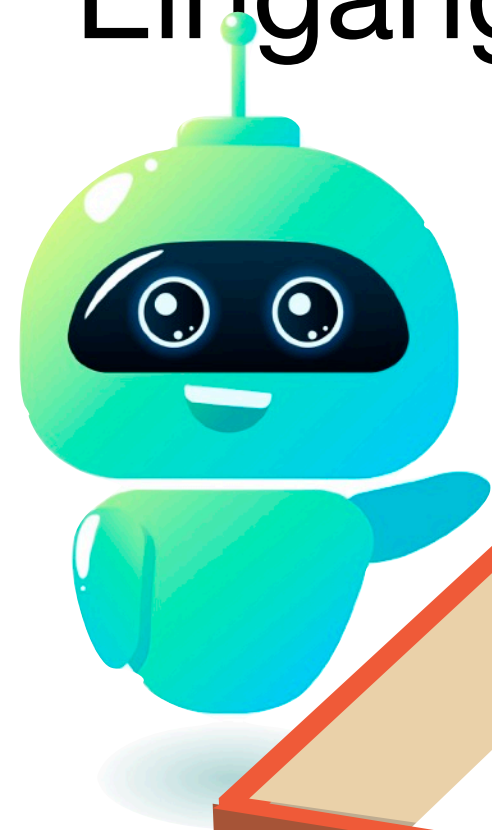
# Beispiel III - Schritt 4

- 3. Solange Eingangsbox nicht leer, lege so viele Zahnräder in die Ausgangsbox, wie in der Merkbox liegen und nehme ein Zahnrad aus der Eingangsbox heraus



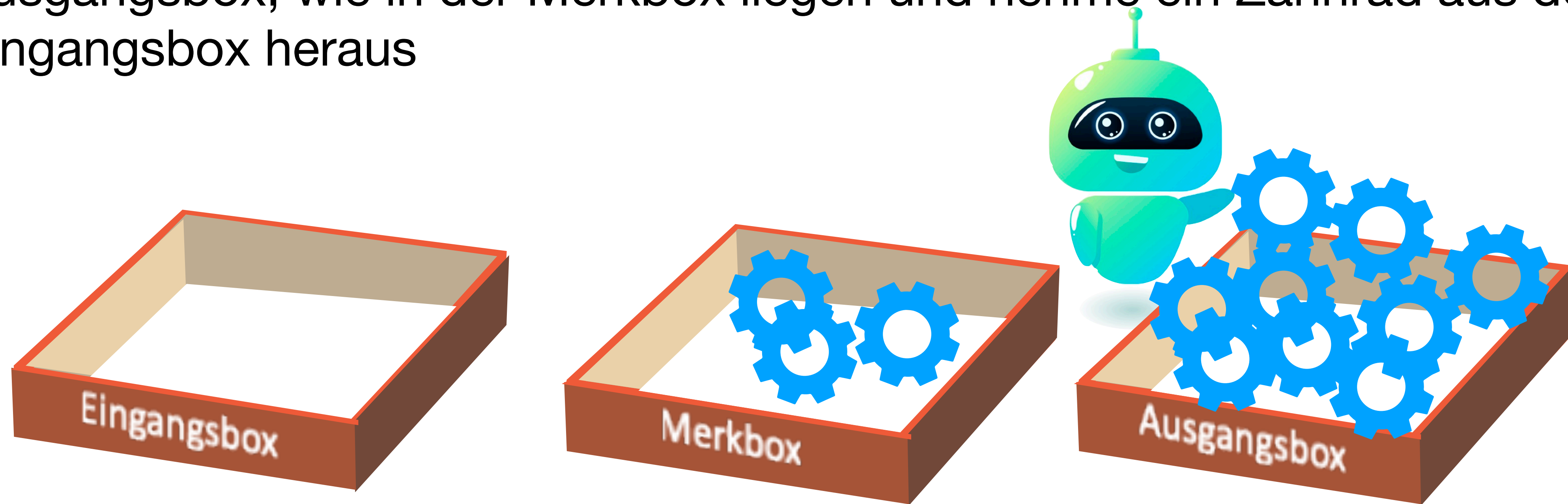
# Beispiel III - Schritt 5

- 3. Solange Eingangsbox nicht leer, lege so viele Zahnräder in die Ausgangsbox, wie in der Merkbox liegen und nehme ein Zahnrad aus der Eingangsbox heraus



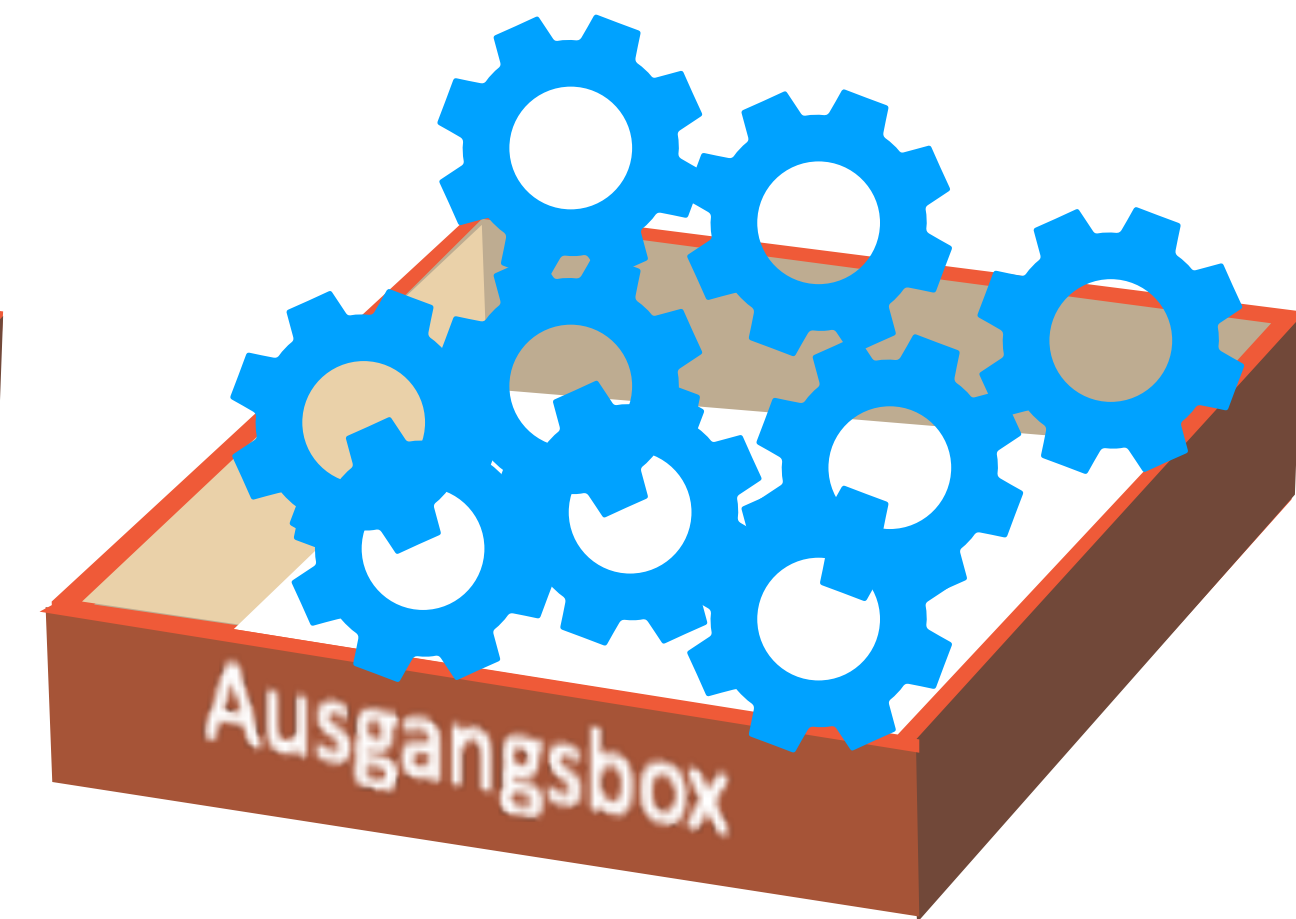
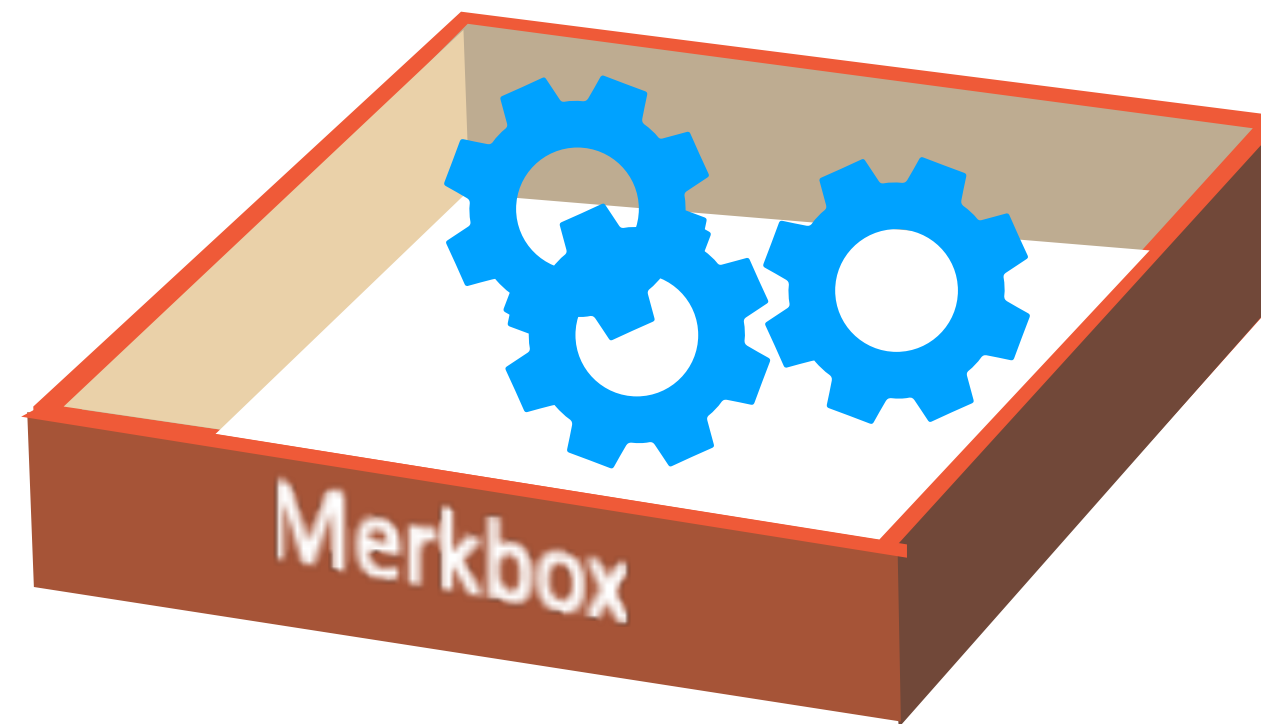
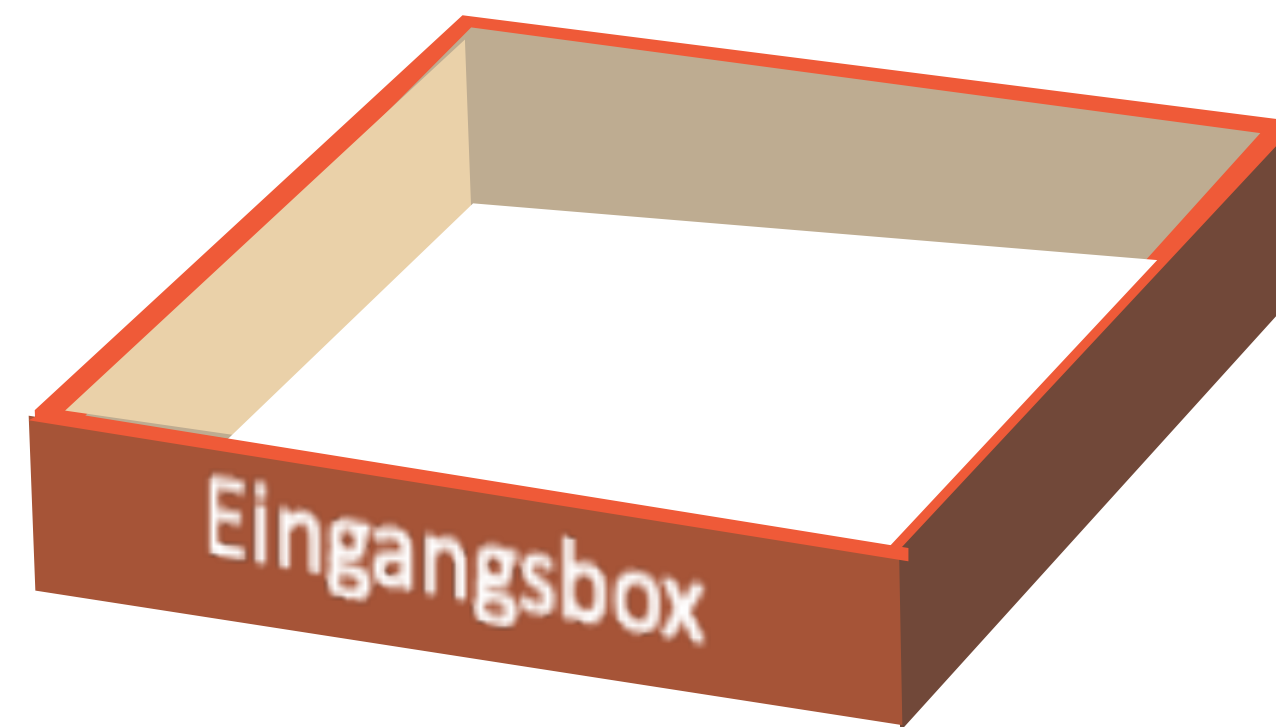
# Beispiel III - Schritt 5

- 3. Solange Eingangsbox nicht leer, lege so viele Zahnräder in die Ausgangsbox, wie in der Merkbox liegen und nehme ein Zahnrad aus der Eingangsbox heraus



# Beispiel III - Schritt 6

- 4. Wenn Eingangsbox leer, dann **Ende**.





# Beispiel III

- 1. Wenn Eingangsbox leer, dann Ende.
- 2. Wenn Eingangsbox nicht leer, lege so viele Zahnräder in eine Merkbox, wie in der Eingangsbox liegen
- 3. Solange Eingangsbox nicht leer, lege so viele Zahnräder in die Ausgangsbox, wie in der Merkbox liegen und nehme ein Zahnrad aus der Eingangsbox heraus
- 4. Wenn Eingangsbox leer, dann Ende.

# Was ist Programmieren?

- Programmieren bedeutet die Umwandlung des Algorithmus in ein Format, das der ausführende Apparat versteht
- Im Fall MUSA übernimmt der Roboter automatisch die Programmierung des Körpers: Das Greifen und Zählen und Weglegen von Gegenständen über die Motorik von Armen und Händen in Kombination mit visuellen (und anderen) Reizen ist bereits angelegt
- Diese Grundfunktionen lösen nicht das Problem des Quadrierens!
- Der Algorithmus löst das Problem des Quadrierens durch richtige Koordination und Aneinanderreihung der Grundfunktionen!

# Was ist Programmieren?

- Programmieren bedeutet die Umwandlung des Algorithmus in ein Format, das der ausführende Apparat versteht
- Im Fall des Computers fehlt dieser Roboter
- Der Mensch muss selbst ran und die Umsetzung in Computer-verständliche Ausdrücke umformen
- Dies ist ein kreativer Akt!
- Was ist also zu tun?

# Was ist Programmieren?

- Unter Programmieren versteht man die Umsetzung einer definierten Verarbeitung von Daten auf einer Maschine



- Mathematisch ist Datenverarbeitung eine Abbildung:

Datenverarbeitung:  $\text{DOM}(\text{Daten}) \rightarrow \text{DOM}(\text{Daten}')$

- Ein **Algorithmus** bestimmt das „**WIE**“ einer Datenverarbeitung und dient der Lösung des Problems
- **Programmieren** nennt man die Tätigkeit zur Umsetzung des Algorithmus in eine für den Computer verständliche Form

# Was ist Programmieren?

- Algorithmen sind abstrakte Handlungsanweisungen (oftmals mit mathematischer Notationen) und sollen dazu dienen, ein Problem auf dem Computer zu lösen
- Ersteller von Algorithmen sollten zumindest wissen, **welche Grundfunktionen von Computern benutzt werden können**
- Später: Strategien bei der Algorithmus-Entwicklung für ein Problem, fließender Übergang zum Programmieren



# Beispiel III

- 1. Wenn Eingangsbox leer, dann Ende.
- 2. Wenn Eingangsbox nicht leer, lege so viele Zahnräder in eine Merkbox, wie in der Eingangsbox liegen
- 3. Solange Eingangsbox nicht leer, lege so viele Zahnräder in die Ausgangsbox, wie in der Merkbox liegen und nehme ein Zahnrad aus der Eingangsbox heraus
- 4. Wenn Eingangsbox leer, dann Ende.

```
int quadrieren(int eingangsbox)
{
    int ausgangsbox ;
    int merkbox = eingangsbox;
    if (eingangsbox==0)
        ausgangsbox = 0;
    else
    {
        ausgangsbox = 0;
        while (eingangsbox!=0)
        {
            eingangsbox = eingangsbox - 1;
            ausgangsbox = ausgangsbox + merkbox;
        }
    }
    return ausgangsbox;
}
```

# Turingmaschine

- Eine Berechnungsvorschrift zur Lösung eines Problems heißt genau dann Algorithmus, wenn eine zu dieser Berechnungsvorschrift äquivalente Turingmaschine existiert, die für jede Eingabe, die eine Lösung besitzt, stoppt.
- Aus dieser Definition sind folgende Eigenschaften eines Algorithmus ableitbar:
  - Das Verfahren muss in einem endlichen Text **eindeutig beschreibbar sein** (Finitheit).
  - Jeder Schritt des Verfahrens muss tatsächlich **ausführbar sein** (Ausführbarkeit).
  - Das Verfahren darf zu jedem Zeitpunkt nur **endlich viel Speicherplatz** benötigen (Dynamische Finitheit).
  - Das Verfahren darf nur **endlich viele Schritte** benötigen (Terminierung, siehe auch Zeitkomplexität).
- Darüber hinaus wird der Begriff Algorithmus in praktischen Bereichen oft auf die folgenden Eigenschaften eingeschränkt:
  - Der Algorithmus muss bei **denselben Voraussetzungen das gleiche Ergebnis** liefern (Determiniertheit).
  - Die **nächste anzuwendende Regel** im Verfahren ist zu jedem Zeitpunkt **eindeutig definiert** (Determinismus).