Problemlösung

Vorlesungsskript Digitalisierung und Programmierung

Prof. Dr. Nicolas Meseth

Inhaltsverzeichnis

Der wichtigste Grund für die Nutzung von Computern ist das Lösen von Problemen. Warum? Weil Computer zwei Eigenschaften besitzen, die für viele Probleme und deren Lösung vorteilhaft sind:

1. Computer machen keine Fehler. Wenn wir einem Computer einen Lösungsweg beibringen, wendet er ihn fehlerfrei auf neue Probleme an.
2. Computer sind unglaublich schnell. Ob einfache Schritte, komplexe Berechnungen oder die Verarbeitung großer Datenmengen – Computer lösen Probleme in einem Bruchteil der Zeit, die wir Menschen benötigen würden.

Diese beiden Eigenschaften ermöglichen es uns, mit Computern besonders solche Probleme effizient zu lösen, die wiederkehrend und in großer Zahl auftreten. Wir sprechen dann von **Automatisierung**.

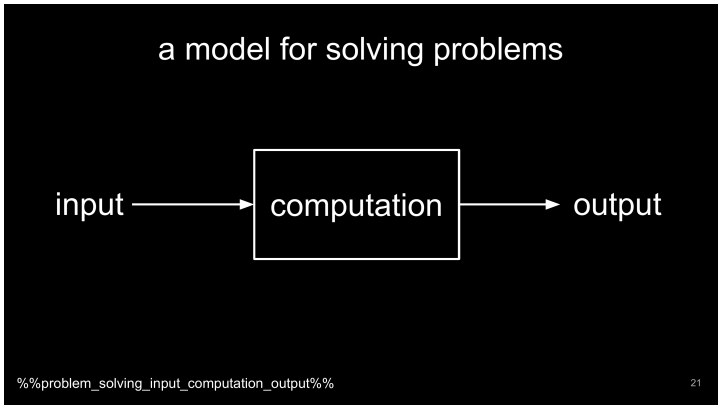
Damit den Begriff des Problems besser verstehen und im Kontext von Computern eingrenzen können, führen wir zunächst ein einfaches Modell ein.

Das Eingabe-Verarbeitung-Ausgabe-Modell

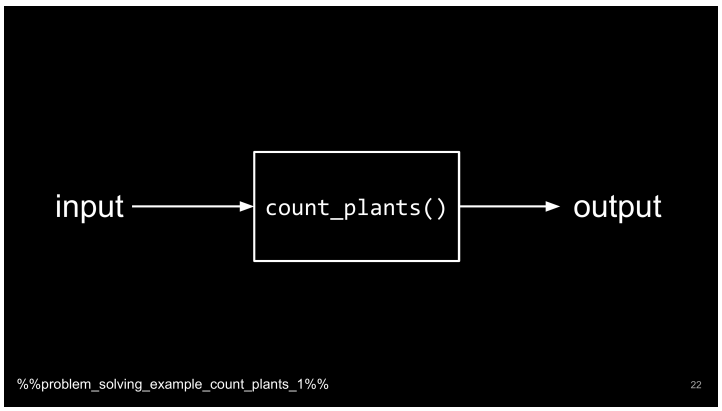
Das Eingabe-Verarbeitung-Ausgabe-Modell (EVA-Modell) ist ein grundlegendes Konzept der Informatik, das die Arbeitsweise von Computern vereinfacht erklärt. Es zeigt, wie Computer Probleme lösen: Sie wandeln Eingabedaten durch einen definierten Verarbeitungsprozess in gewünschte Ausgaben um. Ein Taschenrechner veranschaulicht dies gut: Die Eingabe besteht

aus Zahlen und der gewünschten Operation, die Verarbeitung erfolgt durch mathematische Berechnung, und die Ausgabe ist das Ergebnis.

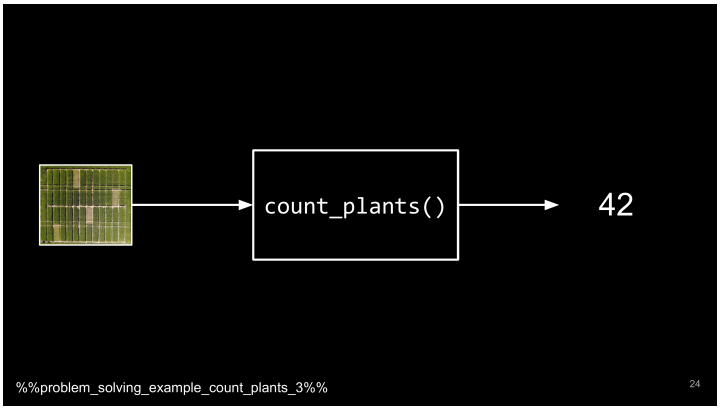
Verallgemeinert beschreibt das Modell ein Problem und dessen Lösung als eine Eingabe (*Input*), die dem Computer übergeben wird. Darauf folgt eine Verarbeitung (*Computation*) auf Basis dieser Eingabe, die wiederum eine Ausgabe (*Output*) erzeugt. Beim Taschenrechner lässt sich das sehr eingängig darstellen, und man kann sich bildlich vorstellen, wie ein Mensch die Eingabe tätigt und das Ergebnis abliest. Es ist aber wichtig zu verstehen, dass Eingabe und Ausgabe sehr unterschiedliche Formen annehmen können und keinesfalls nur über eine Tastatur erfolgen müssen.



Betrachten wir ein weiteres Beispiel: Stell dir vor, du möchtest einen Computer nutzen, um Maispflanzen auf einer Drohnenaufnahme eines Ackers zu zählen. Diese Aufgabe ist für Menschen zwar einfach zu verstehen, wäre aber sehr zeitaufwändig auszuführen. Moderne Algorithmen ermöglichen es Computern, Objekte auf Bildern präzise zu lokalisieren und zu zählen. Nehmen wir an, wir haben für dieses Problem bereits eine Lösung entwickelt und ein Programm namens `count_plants` erstellt. Nun stellt sich die Frage: Wie sehen die Eingabe und die Ausgabe für dieses Problem aus? Was benötigt das Programm von uns, und was liefert es als Ergebnis?

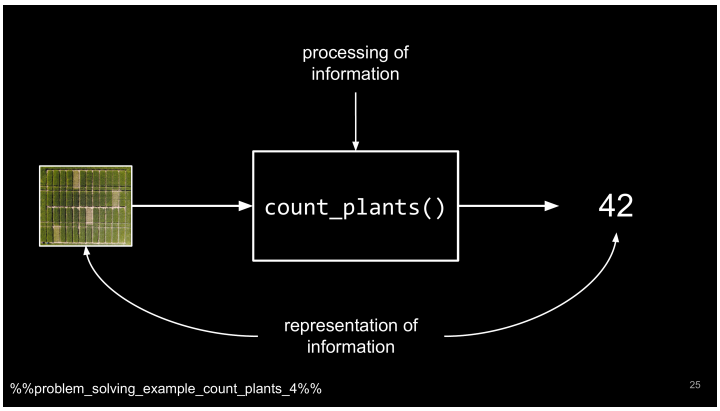


Die erwartete Ausgabe lässt sich einfach beschreiben: Das Ergebnis der Zählung ist eine ganze Zahl. Anders als beim Taschenrechner ist die Eingabe für dieses Problem kein Tastendruck, sondern ein Bild. Dieses Bild muss dem Computer in digitaler Form bereitgestellt werden. Was das genau bedeutet, lernen wir in einem späteren Kapitel. Hier genügt es zu verstehen, dass wir das Bild digital abbilden müssen.



Wie gelangt das Bild in den Computer? Dies ist im Modell nicht näher definiert und für die Problembeschreibung auch nicht wesentlich. Das Bild muss lediglich irgendwie in den Arbeitsspeicher des Programms `count_plants` gelangen. Dies kann auf verschiedene Arten geschehen: Es kann von der Festplatte gelesen werden, über eine drahtlose Verbindung wie Bluetooth direkt übertragen und verarbeitet werden, oder das Programm `count_plants` läuft direkt auf der Drohne und greift unmittelbar auf deren Kamera zu. Die technische Umsetzung ist für unser Modell irrelevant.

Unser Fokus liegt vielmehr darauf, wie die benötigten Informationen für die Ein- und Ausgabe eines Programms in digitaler Form dargestellt werden können. Es geht uns um die **Informationsrepräsentation** in einem digitalen Computer.



Problemlösungsstrategien

Teile und Herrsche (*Divide and Conquer*)

Verteile und Parallelisiere (*Distribute and Parallelize*)

Manche Probleme lassen sich effizienter lösen, wenn mehrere Personen gleichzeitig daran arbeiten. Anstatt eine einzelne Person mit der gesamten Aufgabe zu betrauen, verteilen wir die Arbeit auf mehrere Schultern und arbeiten parallel an der Lösung. Allerdings eignet sich nicht jedes Problem für diesen Ansatz.

Ein gutes Beispiel ist das Aufräumen eines Zimmers: Eine einzelne Person müsste nacheinander verschiedene Bereiche aufräumen, während mehrere Personen gleichzeitig unterschiedliche Ecken des Raums in Angriff nehmen können. Je größer das Zimmer, desto mehr Personen werden benötigt, um es in der gleichen Zeit aufzuräumen. Ein Gegenbeispiel, bei dem diese Strategie nicht funktioniert, ist das Lösen einer mathematischen Gleichung. Hier müssen die einzelnen Rechenschritte aufeinander aufbauen, weshalb das Problem nicht gleichzeitig an mehrere Personen übergeben werden kann, die unabhängig daran arbeiten.

Etwas technischer lässt sich die Strategie des Verteilens und Parallelisierens, die wir im Englischen *Distribute and Parallelize* nennen, wie folgt beschreiben: Wir zerteilen ein großes Problem in unabhängig voneinander lösbare Teile und weisen jeder Ressource (etwa einer Person oder einem Computer) einen Teil zur Bearbeitung zu. Jede Ressource erzeugt ein Ergebnis für ihr Teilproblem, und die Prämisse ist, dass sich am Ende alle Teile zur Gesamtlösung des ursprünglichen Problems zusammenfügen lassen.

- Beispiel Wörter zählen in einem Buch