**Algorithmus-Definition und Eigenschaften**

**Was ist ein Algorithmus?**

Das Wort “Algorithmus” leitet sich vom Namen Muhammed al-Chwarizmi (ca. 783–850) ab. Sein arabisches Lehrbuch über das Rechnehbhjvhjvjhvhvhjblkbvdxknjmn mit indischen Ziffern (um 825) beginnt in der mittelalterlichen lateinischen Übersetzung mit den Worten “Dixit Algorismi” (Algorismi hat gesagt).

Ein Bild, das Entwurf, Porträt, Selbstporträt, Mann enthält.

Automatisch generierte Beschreibung

* **Algorithmus**: Eine genau definierte Handlungsvorschrift zur Lösung eines Problems oder einer bestimmten Art von Problemen in endlich vielen Schritten.

**Alltagsbeispiele**

Algorithmen sind allgegenwärtig:

* Gebrauchsanweisungen und Anleitungen
* Rezepte zum Kochen
* Bastelanleitungen
* Regeln, denen wir folgen (oder auch nicht)
* Prozessabläufe (z. B. Verwendung einer Waschmaschine)
* Wegbeschreibungen

**Eigenschaften von Algorithmen**

1. **Endliche Länge**: Algorithmen bestehen aus einer endlichen Anzahl von Schritten.
2. **Termination**: Ein terminierender Algorithmus endet in endlich vielen Schritten.
3. **Determinismus**:
   * Deterministische Algorithmen haben einen eindeutig vorgeschriebenen Ablauf.
   * Determinierte Algorithmen liefern immer dasselbe Ergebnis.
   * Nicht jeder determinierte Algorithmus ist notwendigerweise determiniert.
4. **Sequentialität / Parallelität**: Algorithmen können sequenziell oder parallel ausgeführt werden.

**Beispiel: Algorithmus von Euklid**

Der Algorithmus von Euklid findet den größten gemeinsamen Teiler (ggT) zweier Zahlen:

1. Solange der Rest der Division von (x) durch (y) nicht null ist:
   * Wenn (x > y), ziehe (y) von (x) ab.
   * Andernfalls ziehe (x) von (y) ab.
2. Das Ergebnis nach der Terminierung ist der ggT ((x) oder (y)).

**Fragen zu Algorithmen**

* **Voraussetzungen**:
  + Unter welchen Bedingungen funktioniert der Algorithmus?
  + Was passiert bei falscher Eingabe?
* **Terminationsverhalten**:
  + Beendet der Algorithmus sich für alle möglichen Eingaben? (Beweis!)
  + Kann der Algorithmus in eine Endlosschleife geraten?
* **Korrektheit**:
  + Löst der Algorithmus das gestellte Problem?
  + Liefert der Algorithmus die gewünschte Ausgabe basierend auf der Eingabe?
* **Effizienz**:
  + Speicherplatzbedarf?
  + Ausführungszeit?
  + Abhängigkeit von der Eingabegröße?

Algorithmen spielen eine grundlegende Rolle in der Informatik und beim Problemlösen. Das Verständnis ihrer Eigenschaften und Entwurfsprinzipien ist für effektives Programmieren und Problemlösen unerlässlich.<!-- I have formatted the scanned text as requested. If you need any further adjustments or have additional requests, feel free to let me know! -->

Hier ist der formatierte Text der gescannten PPT:

**Struktur -> bestimmt den Ablauf**

* Sequentielle Ausführung der Anweisungen
* Bedingte Ausführung einer Anweisung (wenn … dann …)
* Alternative Ausführung zweier Anweisungen (wenn… dann … sonst …)
* Alternative Ausführung mehrerer Anweisungen (… falls … dann …, falls … dann …, falls … dann…)
* Wiederholte Ausführung von Anweisungen
  + Feste Anzahl von Wiederholungen (führe … mal aus: …)
  + Unbestimmte Anzahl mit Abbruchbedingung (solange … führe … aus)

**Prozedurale und objektorientierte Programmierung — Tankred Grosch**

* Seite 6
* Seite 7

**Programmablaufplan (PAP)**

* Ablaufdiagramm für einen Algorithmus
* auch Flussdiagramm oder Programmstrukturplan
* graphische Darstellung zur Umsetzung eines Algorithmus
* beschreibt die Folge von Operationen zur Lösung einer Aufgabe.

Ein Bild, das Text, Screenshot, Schrift, Design enthält.

Automatisch generierte BeschreibungEin Bild, das Diagramm, Text, Entwurf, Plan enthält.

Automatisch generierte Beschreibung

**Operation allgemein**

* Unterprogrammaufruf
* Ein-/Ausgabe
* Verzweigung
* Übergangsstelle
* Grenzstelle
* Ablauflinien
* Schleifenbegrenzung für zählergesteuerte Wiederholung

**Beispiel**

* Programmablaufplan (PAP) D
* Start
* Beispiel
* Eingabe
* Ende

**Algorithmus von Euklid**

* zur Bestimmung des größten gemeinsamen Teilers (ggT)
* solange x ≠ v ist, wiederhole:
  + wenn x > v, dann:
    - ziehe v von x ab und weise das Ergebnis x zu.
  + andernfalls:
    - ziehe x von v ab und weise das Ergebnis v zu.
* ENDE solange
* x (bzw. v) ist der gesuchte ggT

**Aussagen über Algorithmen**

* Voraussetzungen
  + unter welchen Bedingungen arbeitet der Algorithmus?
  + Menge aller erlaubten Eingaben
  + Menge aller möglichen Ausgaben (bei erlaubten Eingaben)
  + Was geschieht bei falscher Eingabe?
* Terminationsverhalten
  + endet ein Algorithmus für alle möglichen Eingaben? Beweis!
  + Ist es möglich, den Algorithmus in einen nicht-endenden Zyklus zu bringen?
* Korrektheit
  + löst der Algorithmus das gestellte Problem?
  + liefert der Algorithmus bei Ausführung die gewünschte Ausgabe als Funktion der Eingabe?

**Aussagen über Algorithmen (Fortsetzung)**

* Aufwand/Effizienz
  + Speicherplatzbedarf?
  + Ausführungszeit?
  + Abhängigkeit der Ausführungszeit von der Eingabe?