Algorithmen und Datenstrukturen

5 Projektphasen der Softwareentwicklung

1.) Auflistung der **Anforderungen** an die Software

2.) **Entwurf mit Modellierungshilfen**

3.) **Implementation (Umsetzung mit einer Programmiersprache)**

4.) **Überprüfung**/Test/Verbesserung

5.) **Wartung/Weiterentwicklung**

Dieses klassische lineare Vorgehensmodell nennst Du

➡ Wasserfallmodell

2. Schritt: Entwurf mit Modellierungshilfen

Modellierungshilfen für den Entwurf einer Software sind z. B.

- Aktivitätsdiagramm
- Struktogramm
- Projektablaufplan (PAP)

Alle diese Diagramme zählen zur UML =
Unified Modeling Language

Die UML ist eine grafische Modelliersprache zur Spezifikation, Konstruktion, Dokumentation und Visualisierung von Software-Teilen und anderen Systemen.

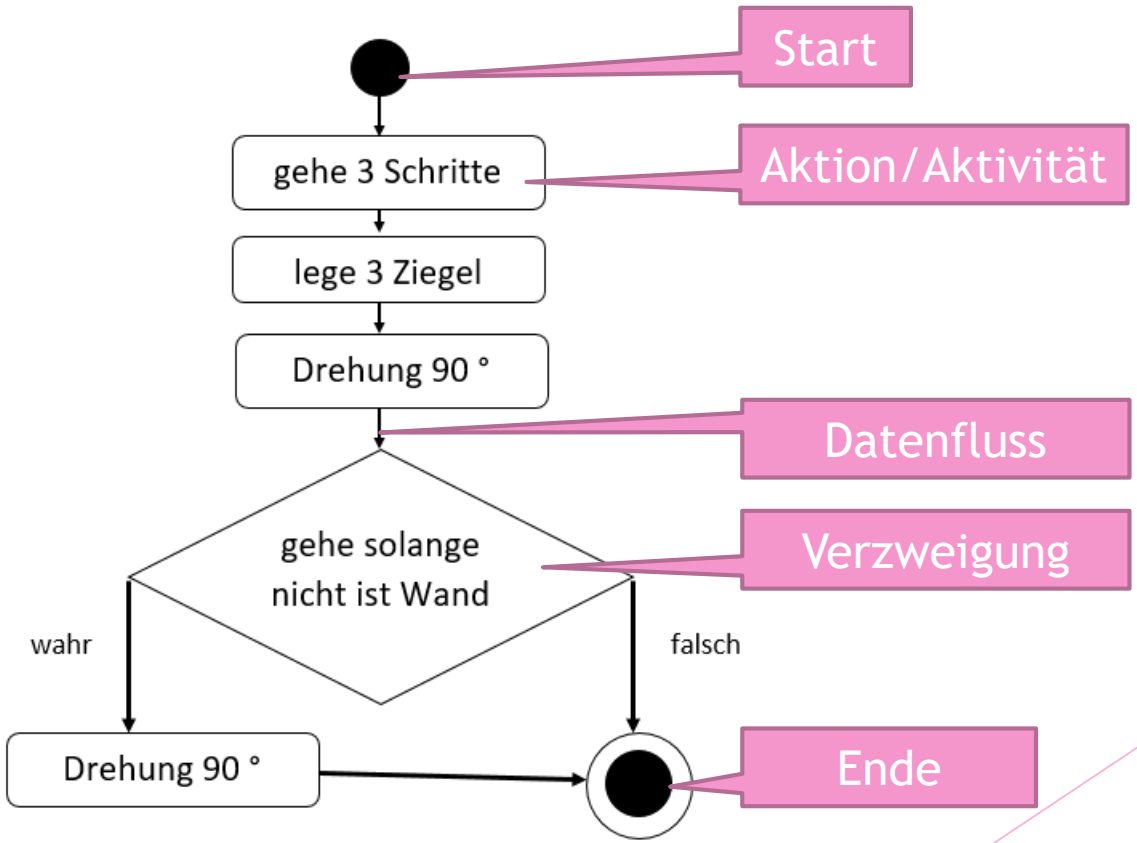
2. Schritt: Entwurf mit Aktivitätsdiagramm

Ein **Aktivitätsdiagramm** ist ein Verhaltensdiagramm der UML und stellt die Vernetzung von elementaren Aktionen und deren Verbindungen mit Kontroll- und Datenflüssen grafisch dar.

Beispiel:

Eine detaillierte und dennoch lesbare Sprache nennst Du

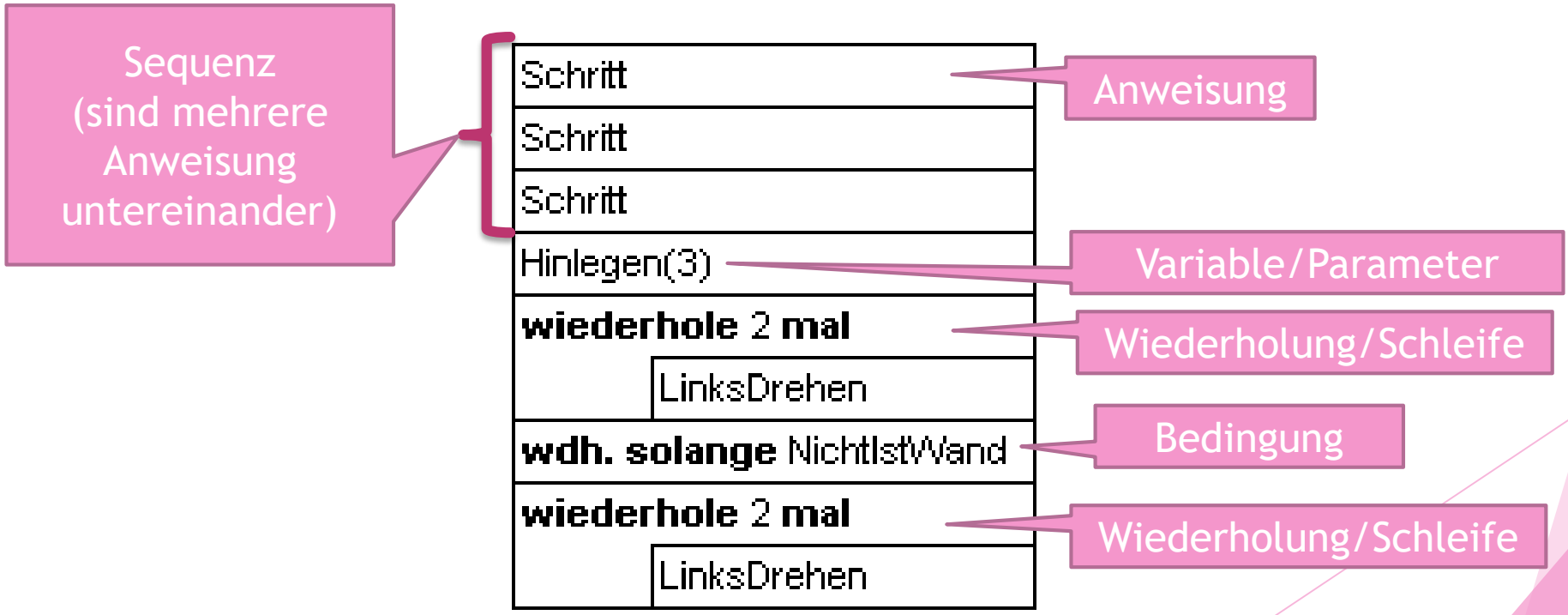
Pseudocode.



2. Schritt: Entwurf mit Struktogramm

Ein **Strukturgramm** ist ebenfalls eine **Modellierhilfe** für Algorithmen ohne der Verwendung einer Programmiersprache. Sie werden nach ihren Entwicklern auch Nassi-Shneiderman-Diagramme genannt. In Deutschland sind sie genormt in der DIN 66261

Beispiel:

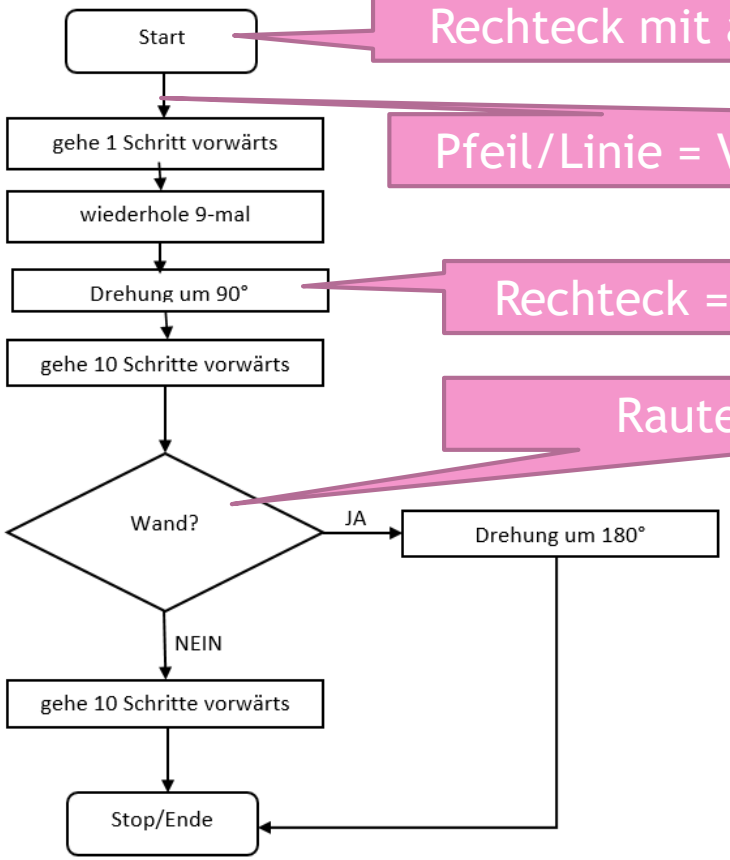


2. Schritt: Entwurf mit Projektablaufplan

Der Programmablaufplan wird zur Dokumentation und Softwareplanung eingesetzt. Informatiker, Programmierer usw. verständigen sich auf diese Art und Weise.

Die Symbole für Programmablaufpläne sind in der DIN 66001 vorgegeben.

Beispiel:



Rechteck mit abgerundeten Ecken = Start und Stop

Pfeil/Linie = Verbindung zum nächstfolgenden Element

Rechteck = Tätigkeit (Operation)

Raute = Entscheidung

3. Schritt: Implementation mit einem Programm

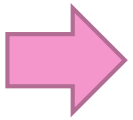
Die **Umsetzung** der Inhalte aus den grafischen Modellierhilfen (Aktivitätsdiagramm, Struktogramm usw.) erfolgt **mit** einer **Software**.
Jede Software hat ihre eigene **Programmiersprache**.

Programmiersprachen sind z. B.:

Eine Programmiersprache erlaubt es, Algorithmen präzise zu beschreiben.

3. Schritt: Implementation mit einem Programm

Was ist ein Algorithmus?



Ein Algorithmus ist eine **präzise Handlungsvorschrift** in **endlich vielen Schritten** zur **Lösung eines Problems.**

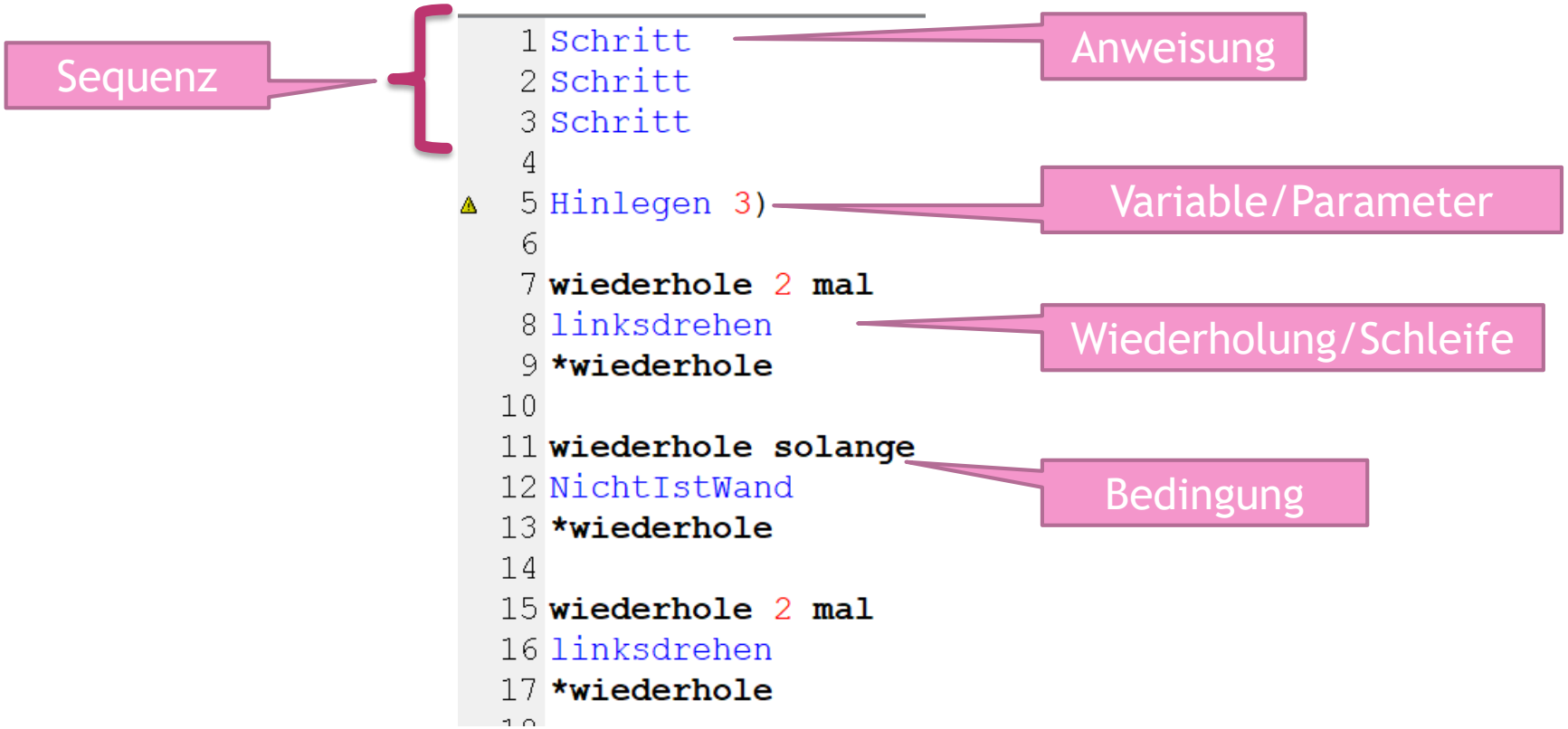
3. Schritt: Implementation mit einem Programm

Welche Merkmale hat ein Algorithmus?

- **Terminierung** — Er kommt zum Ende.
- **Determinismus** — Es gibt nur einen Weg.
- **Determiniertheit** — Unterschiedliche Reihenfolge, aber gleiches Ergebnis.

3. Schritt: Implementation mit einem Programm

Welche Grundbausteine besitzen Algorithmen einer Programmiersprache?



3. Schritt: Implementation mit einem Programm

Was verstehst Du unter **Syntax**, **Debuggen** und **Semantik**?

Die **Syntax** ist die **Schreibweise** einer **Programmiersprache**.

Du erhältst einen **Syntax-Fehlerhinweis**, wenn Du die falsche Schreibweise gewählt hast.

Ein **Bug** ist ein Fehler. **Debuggen** bedeutet, den **Fehler beseitigen.**

Die **Semantik** beschreibt den **Sinn** eines Programms. Die Syntax ist korrekt, aber das gewünschte Ergebnis ist falsch.

```
 1 Schritt
 2 Schritt
 3 Schritt
 4
 5 Hinlegen 3)
 6
 7 wiederhole 2 mal
 8 linksdrehen
 9 *wiederhole
10
11 wiederhole solange
12 NichtIstWand
13 *wiederhole
14
15 wiederhole 2 mal
16 linksdrehen
17 *wiederhole
```