

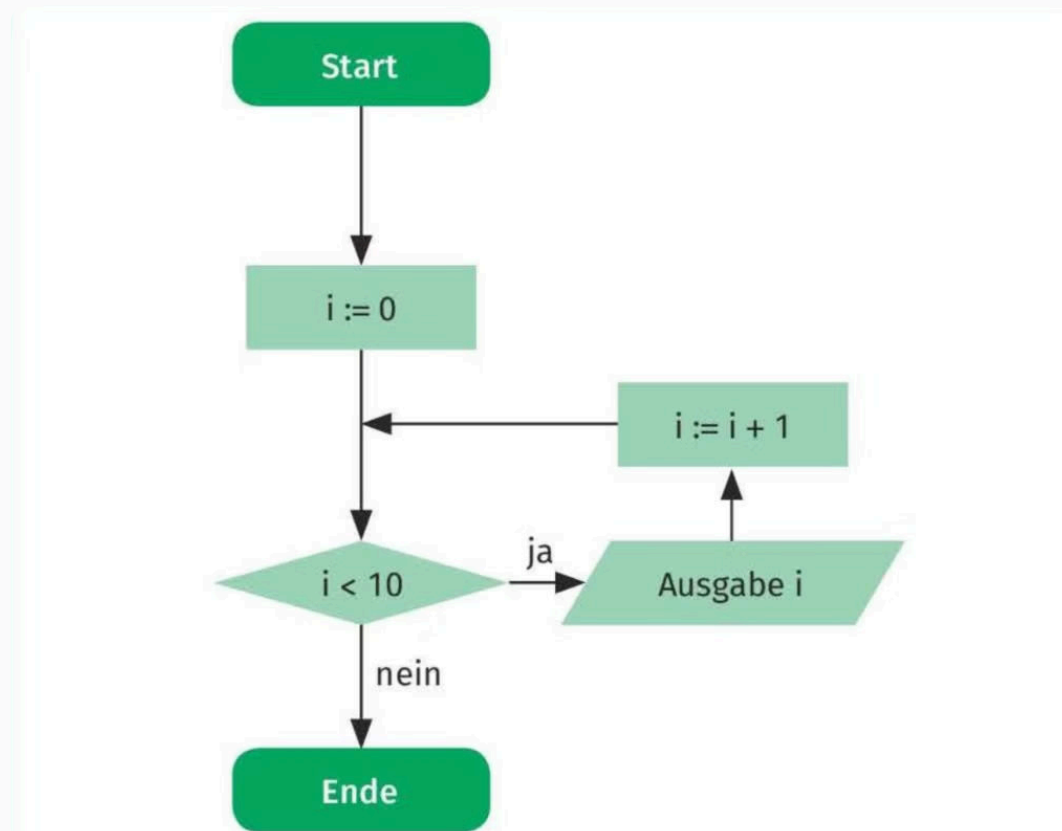
Programmablaufplan und Struktogramm: Grundlagen der strukturierten Programmierung

Programmablaufpläne und Struktogramme sind zwei wichtige grafische Modelle in der Softwareentwicklung, insbesondere im Bereich der strukturierten Programmierung. Beide Methoden dienen dazu, Algorithmen und Programmabläufe visuell darzustellen, unterscheiden sich jedoch in ihrer Struktur und Anwendung. Während Programmablaufpläne eine flexible Darstellung ermöglichen, bieten Struktogramme einen stärker strukturierten Ansatz. Diese Dokument erläutert die Grundlagen, Unterschiede und Anwendungsbereiche beider Modelle.

Programmablaufplan (PAP):

Der Programmablaufplan, auch als Flussdiagramm oder Programmstrukturplan bekannt, ist eine etablierte Darstellungsform für Programmabläufe in der imperativen Programmierung. Die Symbole für Programmablaufpläne sind nach der DIN 66001 genormt, was eine einheitliche Interpretation gewährleistet.

Ein wesentliches Merkmal von Programmablaufplänen ist ihre Flexibilität. Die Schritte werden symbolisch durch Sinnbilder dargestellt, während der Steuerungsablauf durch Ablauflinien visualisiert wird. Diese Pfeile zur Darstellung des Steuerungsablaufs unterliegen keinerlei Einschränkungen, was eine sehr freie Gestaltung ermöglicht.



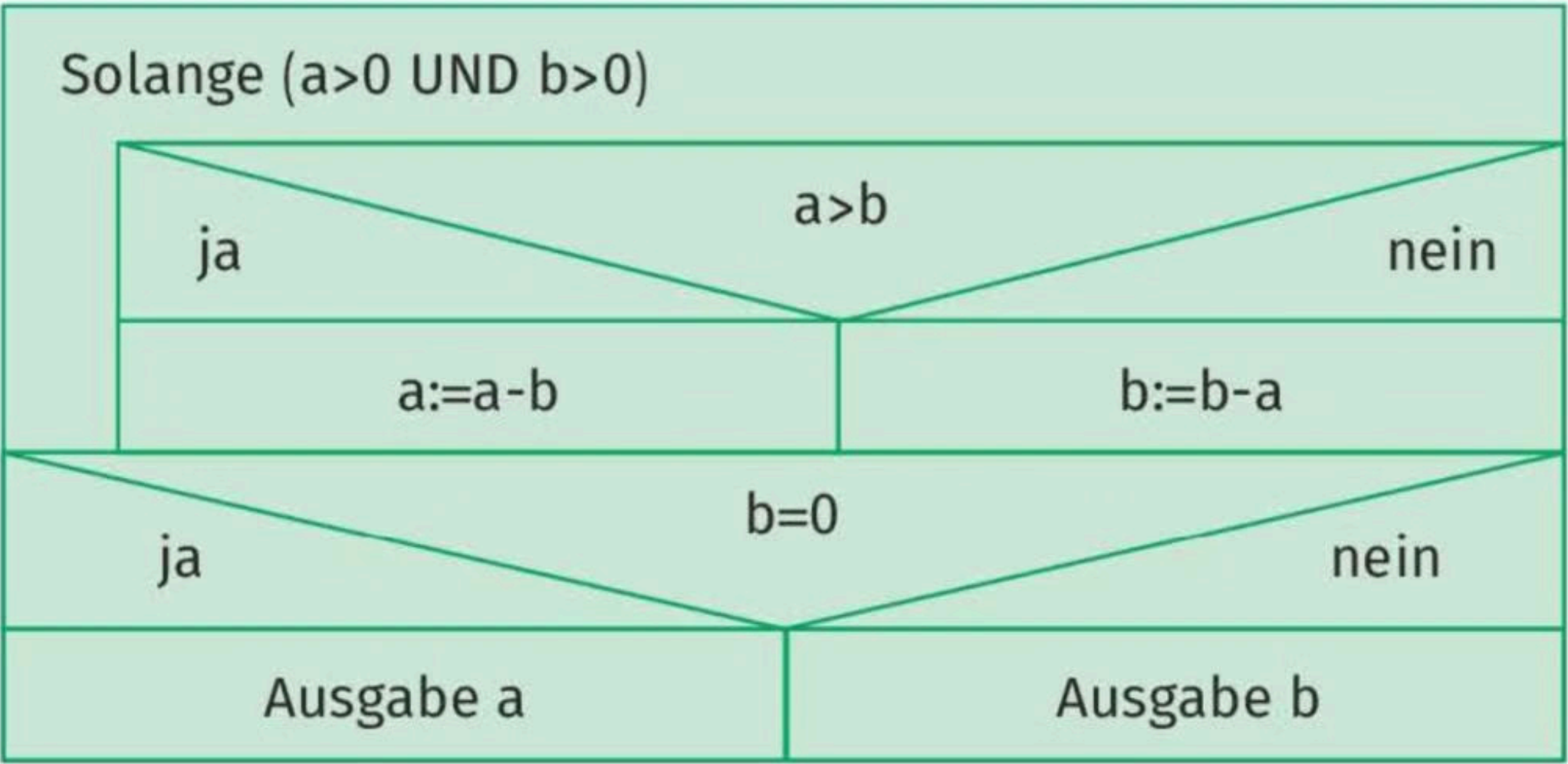
- 1 — Verarbeitung
Darstellung von Aktionen oder Berechnungen durch Rechtecke.
- 2 — Entscheidung
Verwendung von Rauten für bedingte Verzweigungen.
- 3 — Ein-/Ausgabe
Repräsentation durch Parallelogramme.
- 4 — Start/Ende
Markierung durch abgerundete Rechtecke.

Diese Flexibilität kann jedoch auch als Nachteil gesehen werden, da sie Ablaufstrukturen ermöglicht, die nicht den gängigen Regeln der strukturierten Programmierung entsprechen. Aus diesem Grund werden Programmablaufpläne in der modernen Softwareentwicklung seltener eingesetzt, bleiben aber ein wichtiges Werkzeug für das Verständnis grundlegender Programmierkonzepte.

Struktogramm:

Das Struktogramm, auch als Nassi-Schneiderman-Diagramm bekannt, wurde als Alternative zum Programmablaufplan entwickelt. Es dient ebenfalls zur Darstellung von Programmentwürfen im Rahmen der imperativen Programmierung, folgt jedoch einem stärker strukturierten Ansatz. Die Symbole für Struktogramme sind nach der DIN 66261 genormt.

Im Gegensatz zu Programmablaufplänen lassen Struktogramme keine beliebige Darstellung von Abläufen zu. Die grafische Darstellung erfolgt in einem geschlossenen Block mit einem definierten Eingangspunkt oben und einem definierten Ausgangspunkt unten. Diese Struktur fördert die Entwicklung von Algorithmen nach dem Top-Down-Prinzip, bei dem ein komplexes Ausgangsproblem schrittweise in übersichtlichere Teilprobleme zerlegt wird.



Sequenz

Darstellung
aufeinanderfolgender
Anweisungen durch
übereinander angeordnete
Rechtecke.

Verzweigung

Verwendung von geteilten
Rechtecken für bedingte
Ausführungen.

Schleife

Darstellung von
Wiederholungen durch
spezielle Rechteckformen
mit Bedingungsfeldern.

Unterprogramm

Kennzeichnung von Funktionsaufrufen durch doppelte vertikale Linien.

Struktogramme eignen sich besonders gut für den Einstieg in die Programmierung und sind nach wie vor ein fester Bestandteil des Lehrstoffes an Schulen und Universitäten. Sie helfen dabei, den Ablauf von Algorithmen zu verstehen und zu planen, bevor der eigentliche Code geschrieben wird.



Vergleich und Anwendungsbereiche

Sowohl Programmablaufpläne als auch Struktogramme gehören zu den ältesten grafischen Modellen für die Softwareentwicklung und sind durch DIN-Normen standardisiert. Beide Modelle lassen sich in die Ablaufstrukturen Verarbeitung, Folge, Auswahl und Wiederholung unterteilen, unterscheiden sich jedoch in ihrer Darstellungsweise und Flexibilität.

Programmablaufplan

- Flexible Darstellung durch Verwendung von Pfeilen
- Ermöglicht auch nicht-strukturierte Abläufe
- Gut geeignet für einfache Prozesse und Workflows
- Leicht verständlich für Nicht-Programmierer

Struktogramm

- Strukturierte Darstellung in geschlossenem Block
- Fördert strukturierte Programmierung
- Ideal für Top-Down-Entwicklung von Algorithmen
- Besonders nützlich für Programmierer und Studenten

Gemeinsame Merkmale

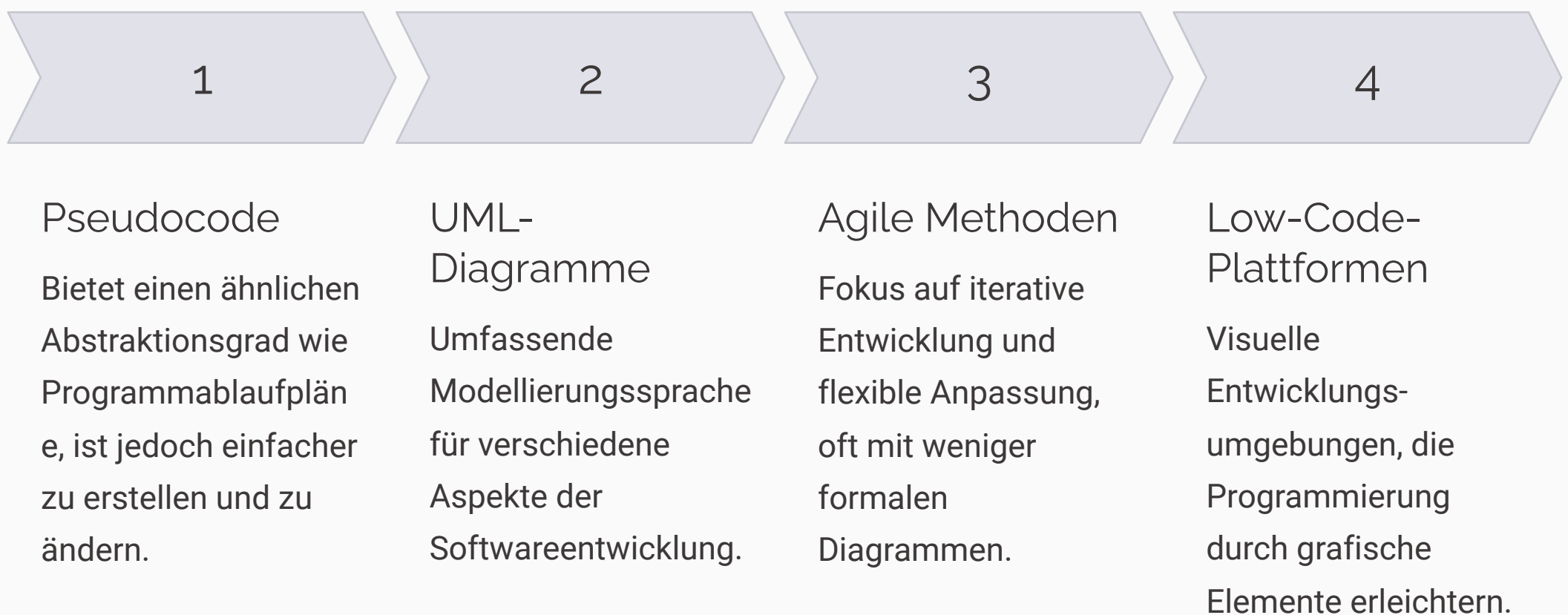
- Standardisierte Symbole (DIN-Normen)
- Visuelle Darstellung von Algorithmen
- Hilfreich beim Verstehen von Programmabläufen
- Wichtig für die Lehre der Programmierung

In der modernen Softwareentwicklung werden beide Methoden zwar seltener eingesetzt, bleiben aber wichtige Werkzeuge für das Verständnis und die Planung von Algorithmen, insbesondere in der Ausbildung und bei der Kommunikation mit Nicht-Programmierern.

Moderne Alternativen

Obwohl Programmablaufpläne und Struktogramme wichtige historische und didaktische Bedeutung haben, werden sie in der professionellen Softwareentwicklung zunehmend durch modernere Methoden ersetzt. Ein Hauptgrund dafür ist die Verfügbarkeit komplexerer Beschreibungssprachen und Modellierungstechniken, die den Anforderungen moderner Softwareprojekte besser gerecht werden.

Eine weit verbreitete Alternative ist die Unified Modeling Language (UML), die eine Vielzahl von Diagrammtypen für verschiedene Aspekte der Softwareentwicklung bietet. UML ermöglicht eine detailliertere und umfassendere Modellierung von Softwaresystemen, einschließlich Klassenstrukturen, Interaktionen und Zustandsübergängen.



Trotz des Rückgangs in der professionellen Anwendung bleiben Programmablaufpläne und Struktogramme wichtige Werkzeuge in der Informatikausbildung. Sie helfen Studierenden, grundlegende Konzepte der Algorithmenentwicklung zu verstehen und logisches Denken zu fördern. In Zukunft könnten diese klassischen Methoden durch interaktive, digitale Lerntools ergänzt werden, die eine dynamischere Darstellung und Manipulation von Algorithmen ermöglichen.

Pseudocode

Der Pseudocode ist eine umgangssprachliche Beschreibung eines Programmablaufs, die sich an eine Programmiersprache anlehnt. Pseudocode ist nicht genormt und kann daher relativ frei formuliert werden. In der Regel sollte sich die Syntax des Pseudocodes an die Programmiersprache anlehnen, in der das Programm geschrieben wird. Dabei können Variablen- und Konstantendeklarationen bereits enthalten sein. Möchte man den Pseudocode stärker programmiersprachenunabhängig formulieren, kann man auch Wörter der eigenen Muttersprache verwenden.

Struktur	Pseudocode	Beispiel
Verarbeitung	Anweisung1	durchschnitt := gesamt / anzahl
Folge (Sequenz)	Anweisung1 Anweisung2 ...	anzahl := 20 gesamt := 200 durchschnitt := gesamt / anzahl
Auswahl (Verzweigung)	Wenn <i>Bedingung</i> dann ... sonst ...	Wenn anzahl > 30 dann x := anzahl *2 y := anzahl /2 sonst Ausgabe: „Fehler“ Ende Verzweigung
Wiederholung (kopfgesteuerte Schleife)	Wiederhole solange <i>Bedingung</i> ...	Wiederhole solange x < 10 Ausgabe: x x := x + 1 Ende Schleife
Wiederholung (fußgesteuerte Schleife)	Wiederhole ... solange <i>Bedingung</i>	Wiederhole Ausgabe: x x := x + 1 solange x < 10
Wiederholung (Zählschleife)	Von <i>Anfangswert</i> Bis <i>Endwert</i> ...	Von i:=0 Bis 10 Ausgabe: x Ende Schleife