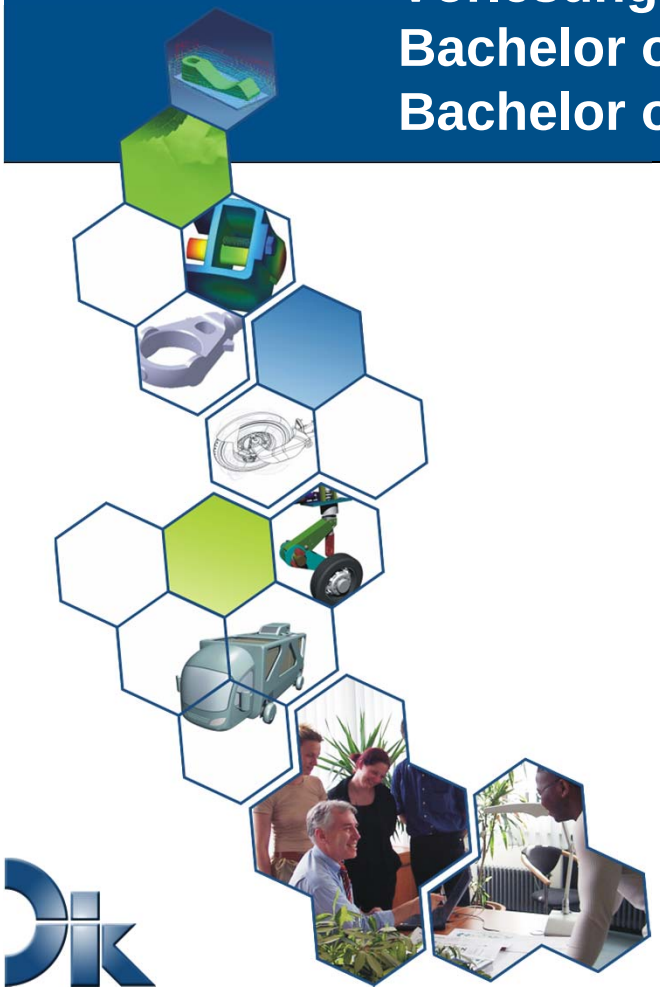
Grundlagen der elektronischen Datenverarbeitung

Vorlesung für Bachelor of Science MPE, Mechanik, Bachelor of Education

TECHNISCHE UNIVERSITÄT DARMSTADT



Die Vorlesung wird zusätzlich in den Raum S3|11-006 (1 Stockwerk tiefer) übertragen.

Kapitel 3

Programmiersprachen- und techniken

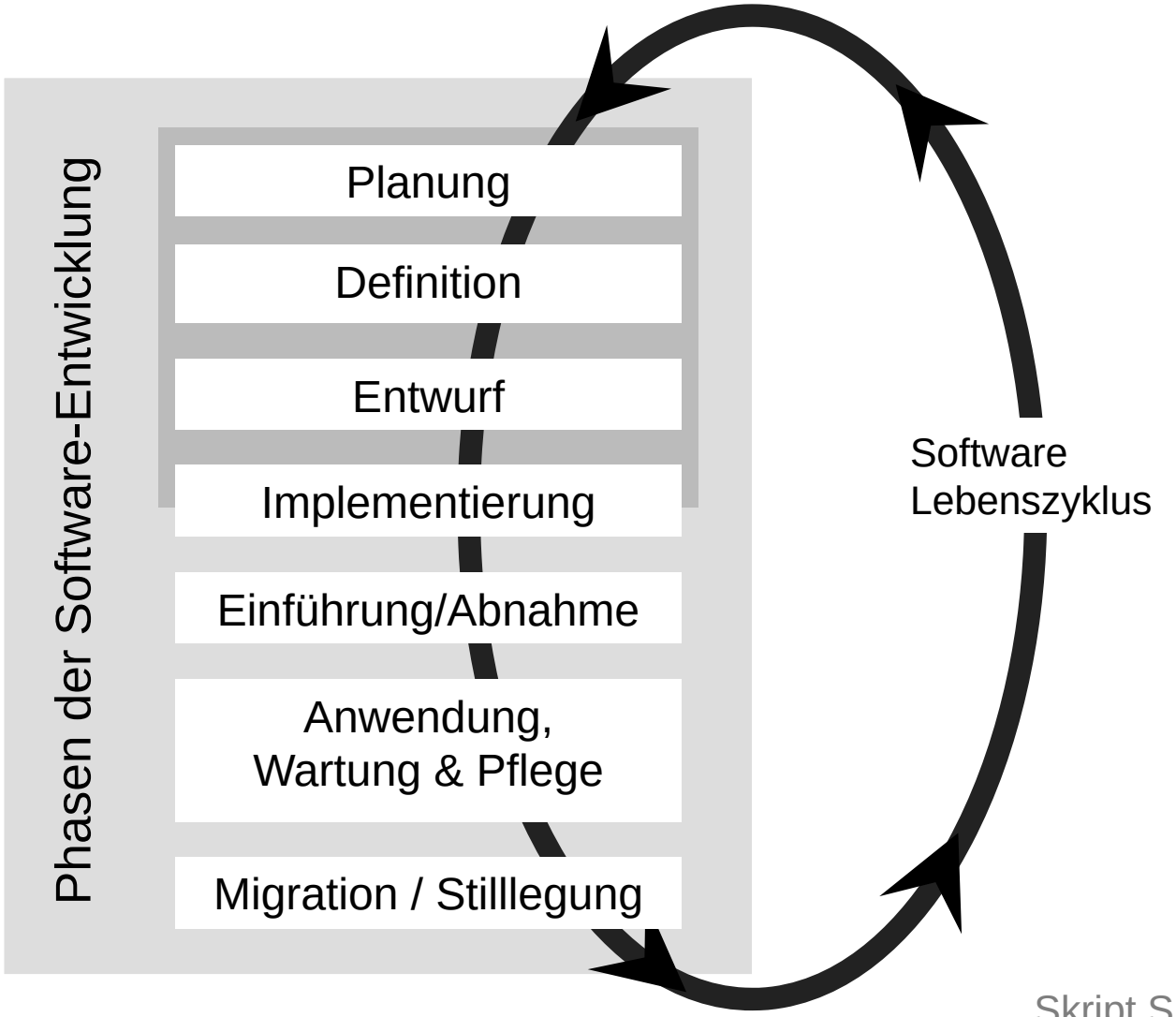
Gliederung 4. Vorlesungsstunde GeDV

TECHNISCHE UNIVERSITÄT DARMSTADT

Kapitel 3

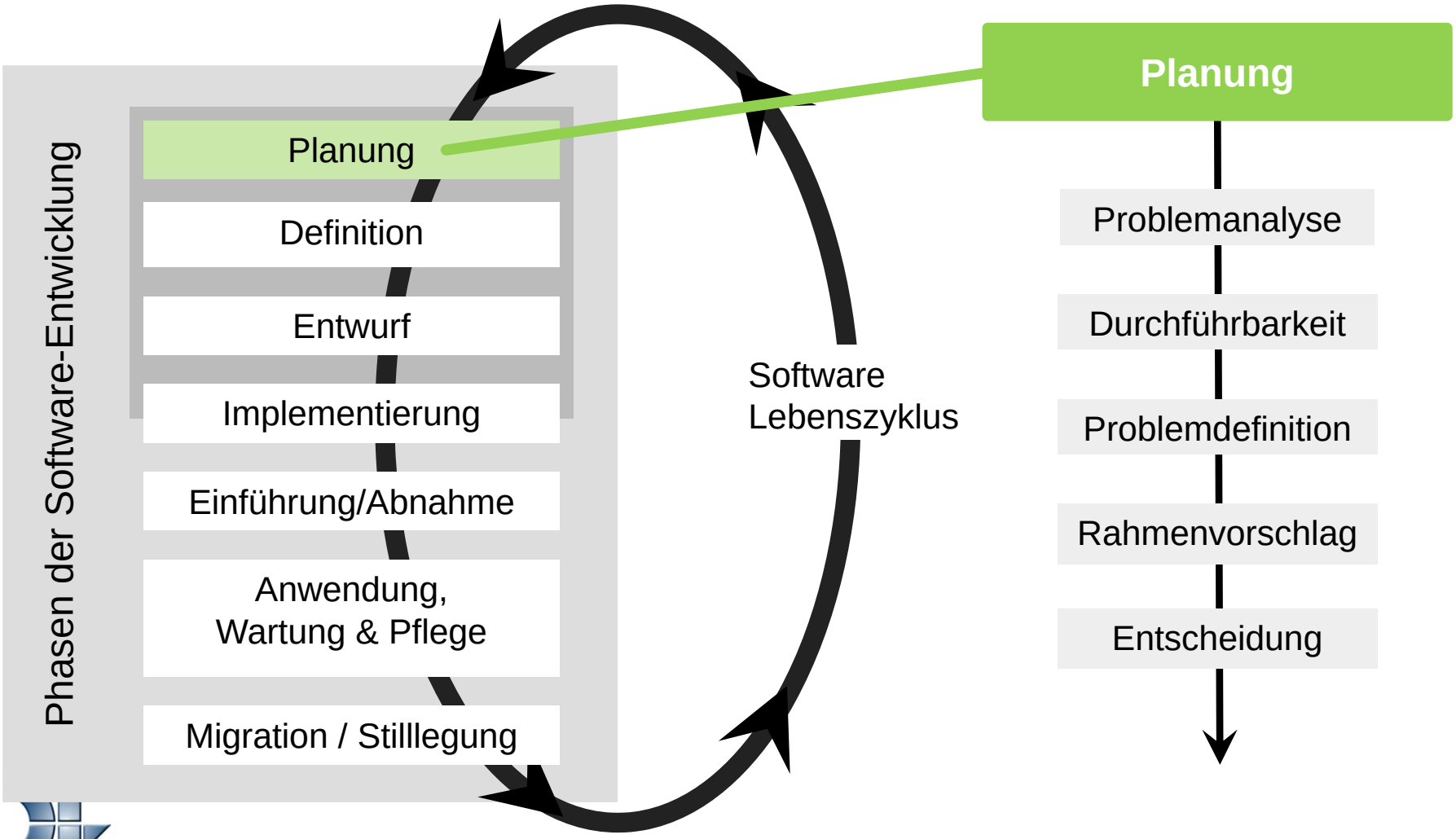
- Programmiersprachen und Techniken
- Softwarelebenszyklus
- Programmkonstrukte
- Struktogramme und Programmablaufplan
- Programmiersprachen
- Einführung in die objektorientierte Programmierung

Der Softwarelebenszyklus



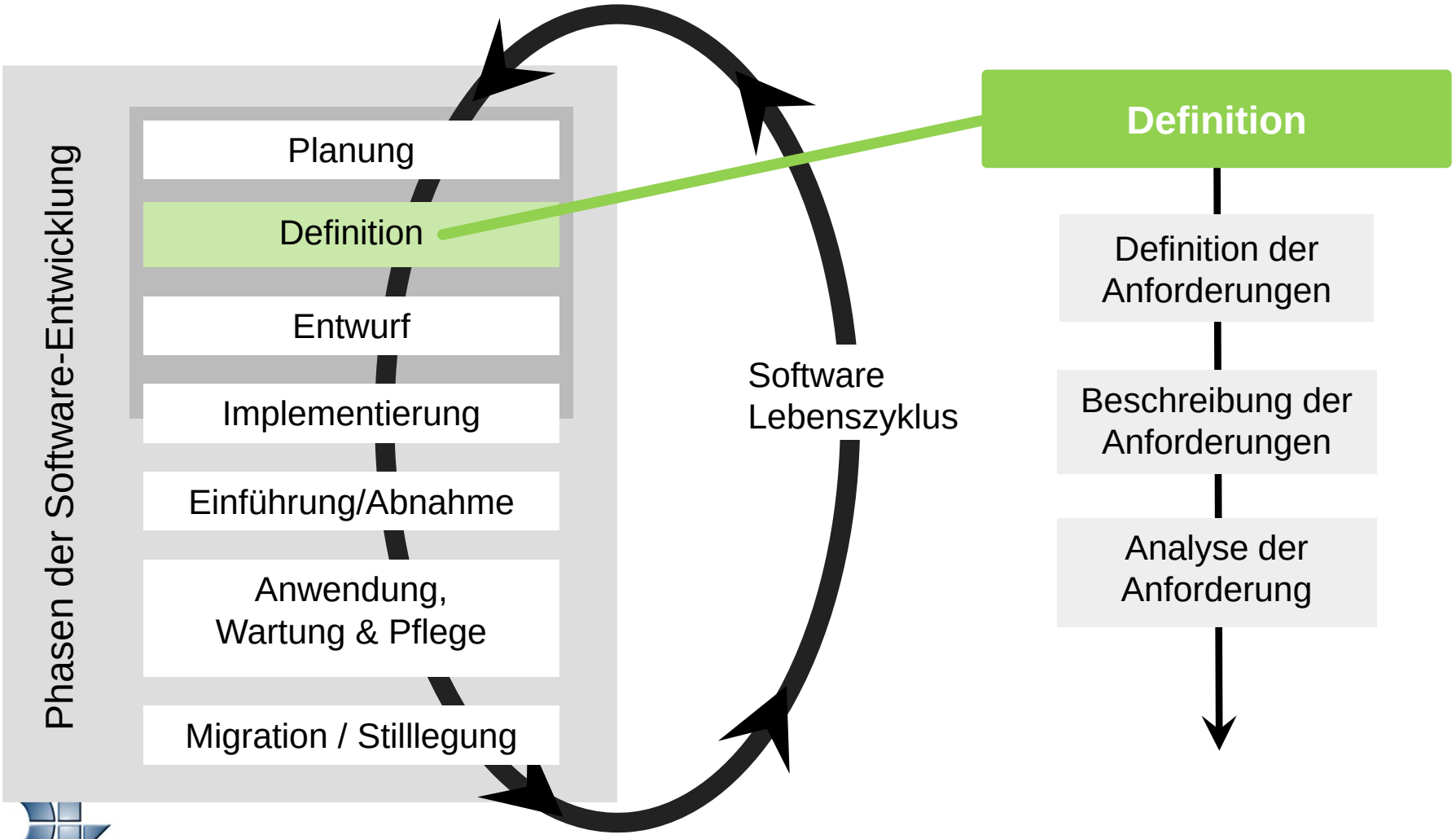
Skript S. 70 ff

Die Planungsphase



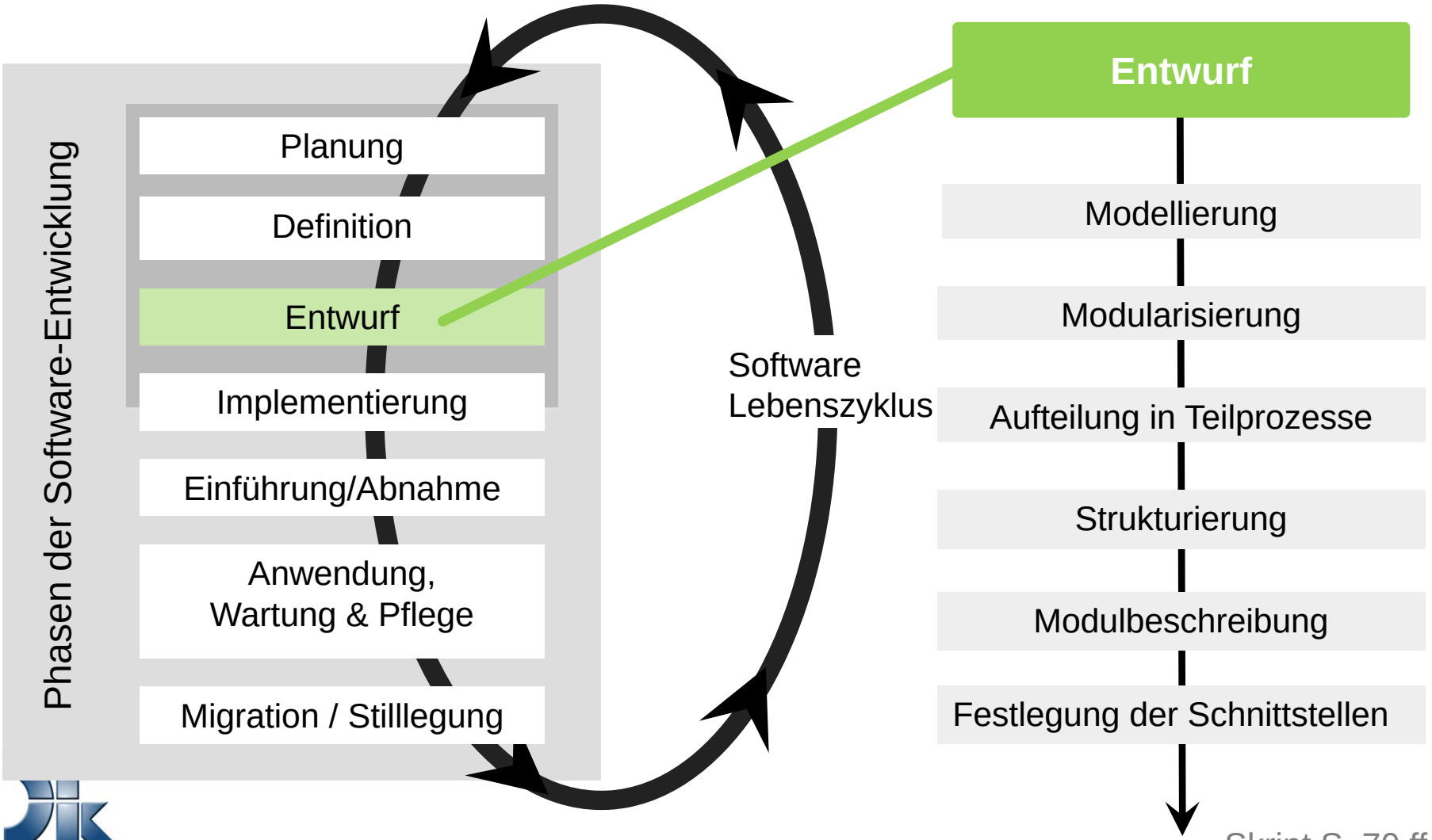
Skript S. 70 ff

Die Definitionsphase



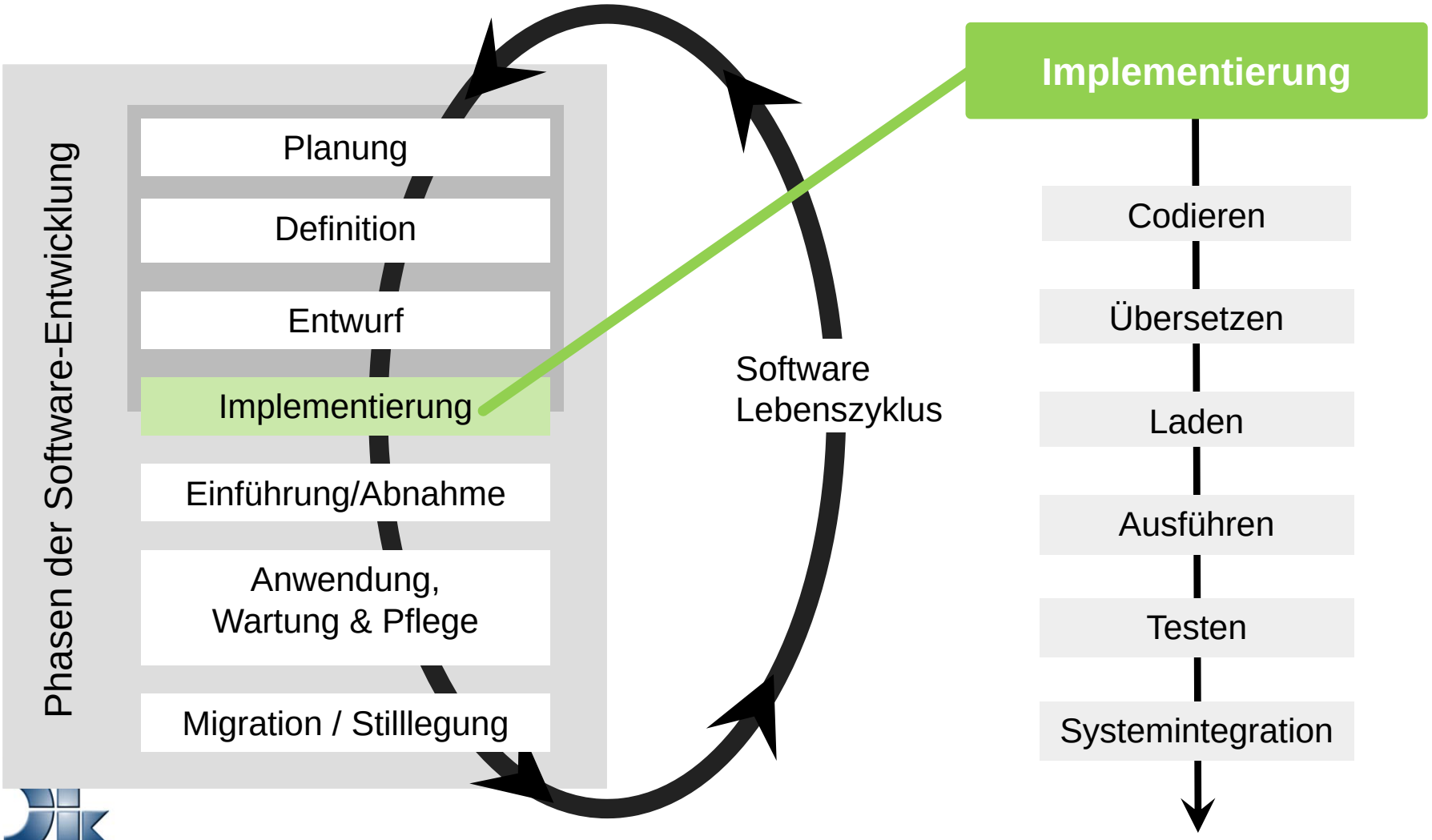
Skript S. 70 ff

Die Entwurfsphase



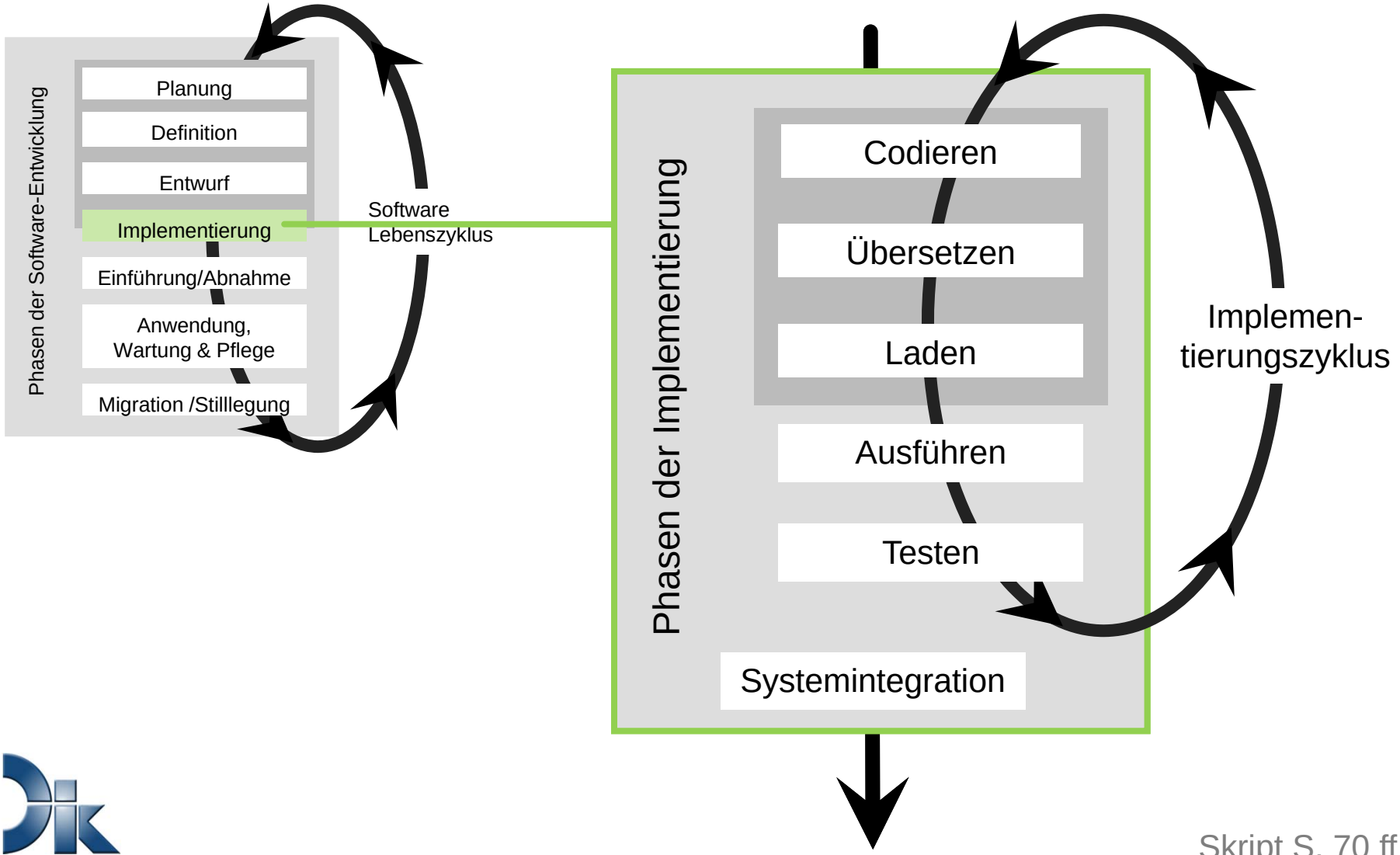
Skript S. 70 ff

Die Implementierungsphase



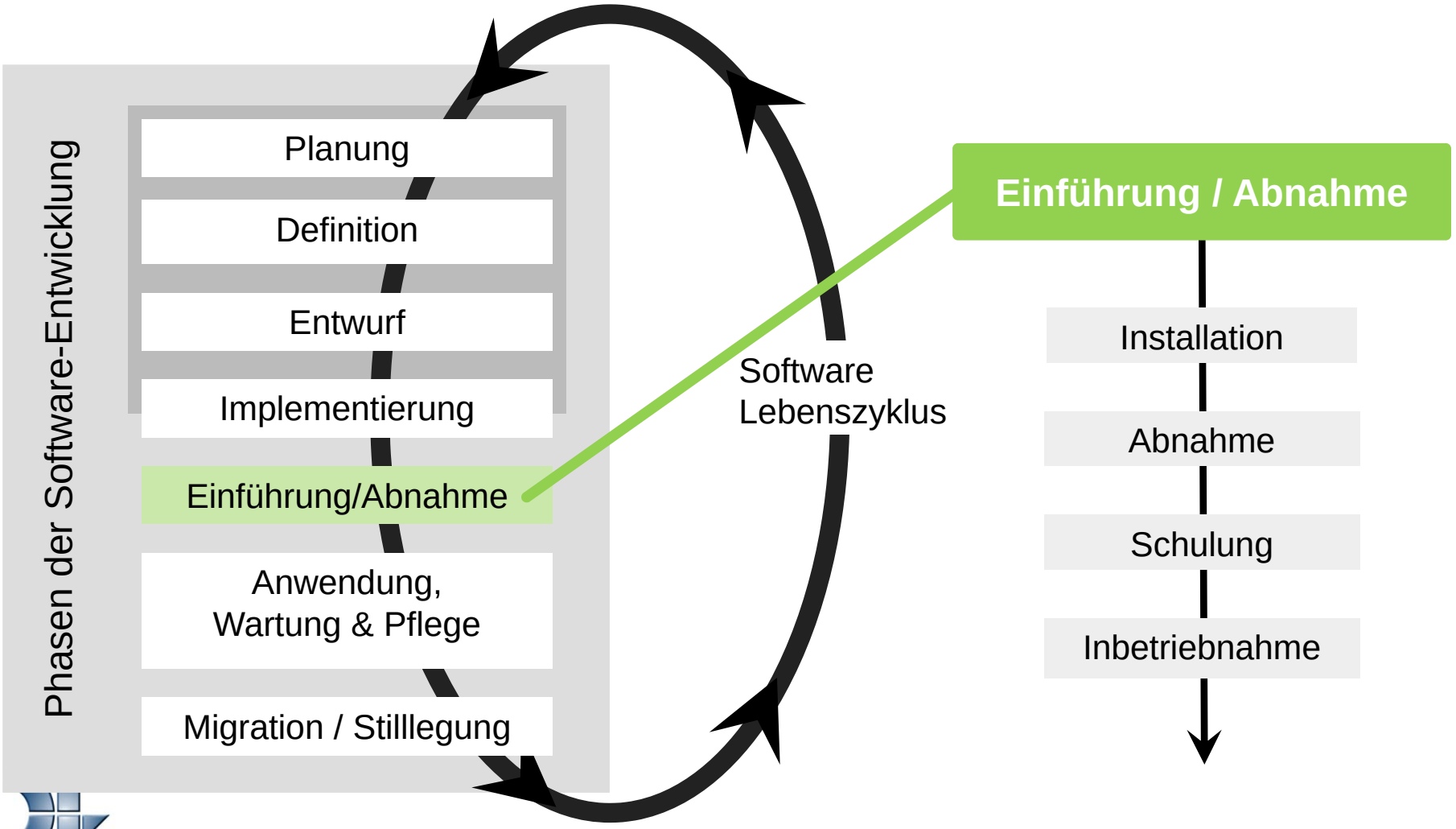
Skript S. 70 ff

Prozesskette Implementierung



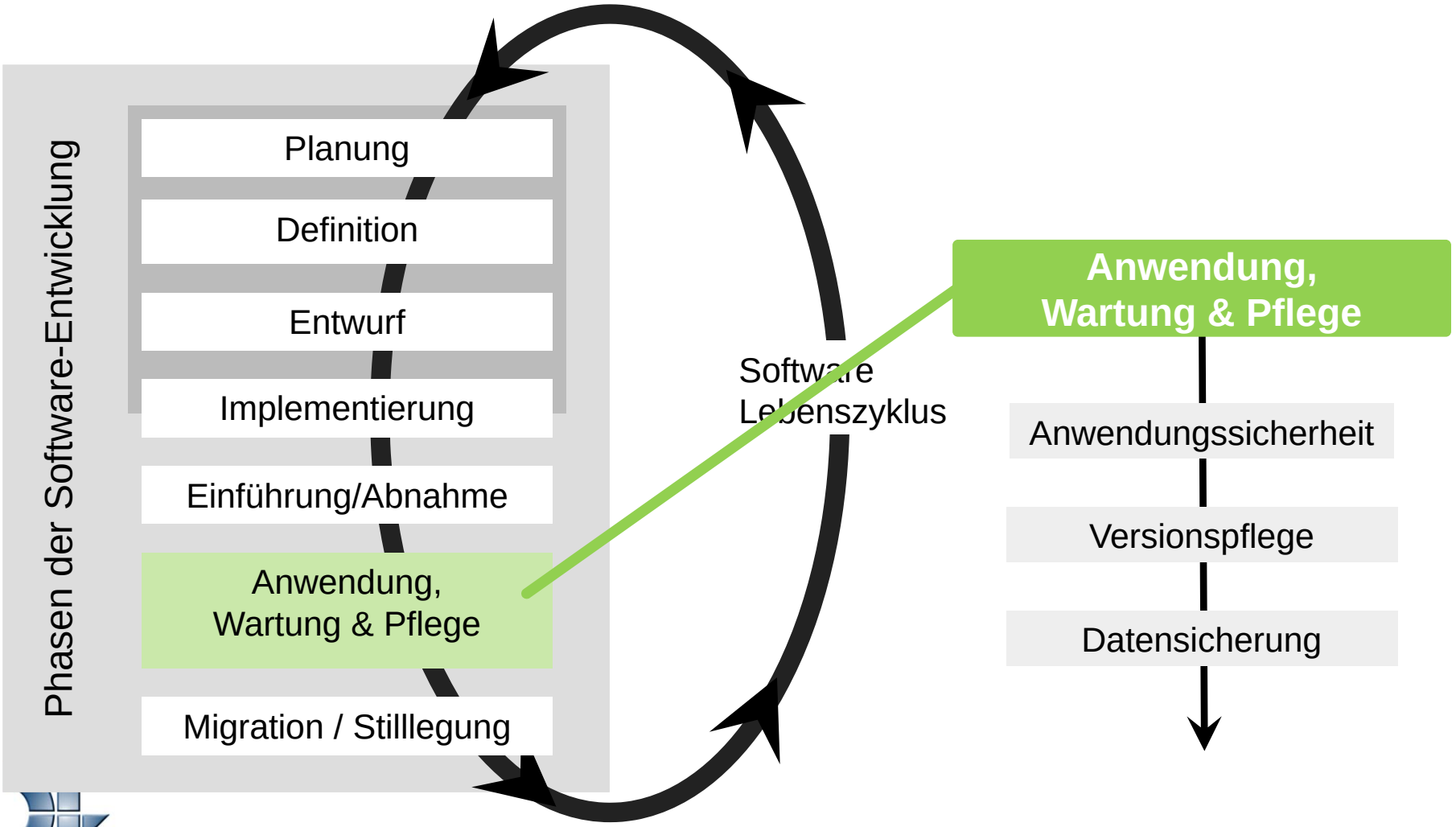
Skript S. 70 ff

Die Abnahme- und Einführungsphase



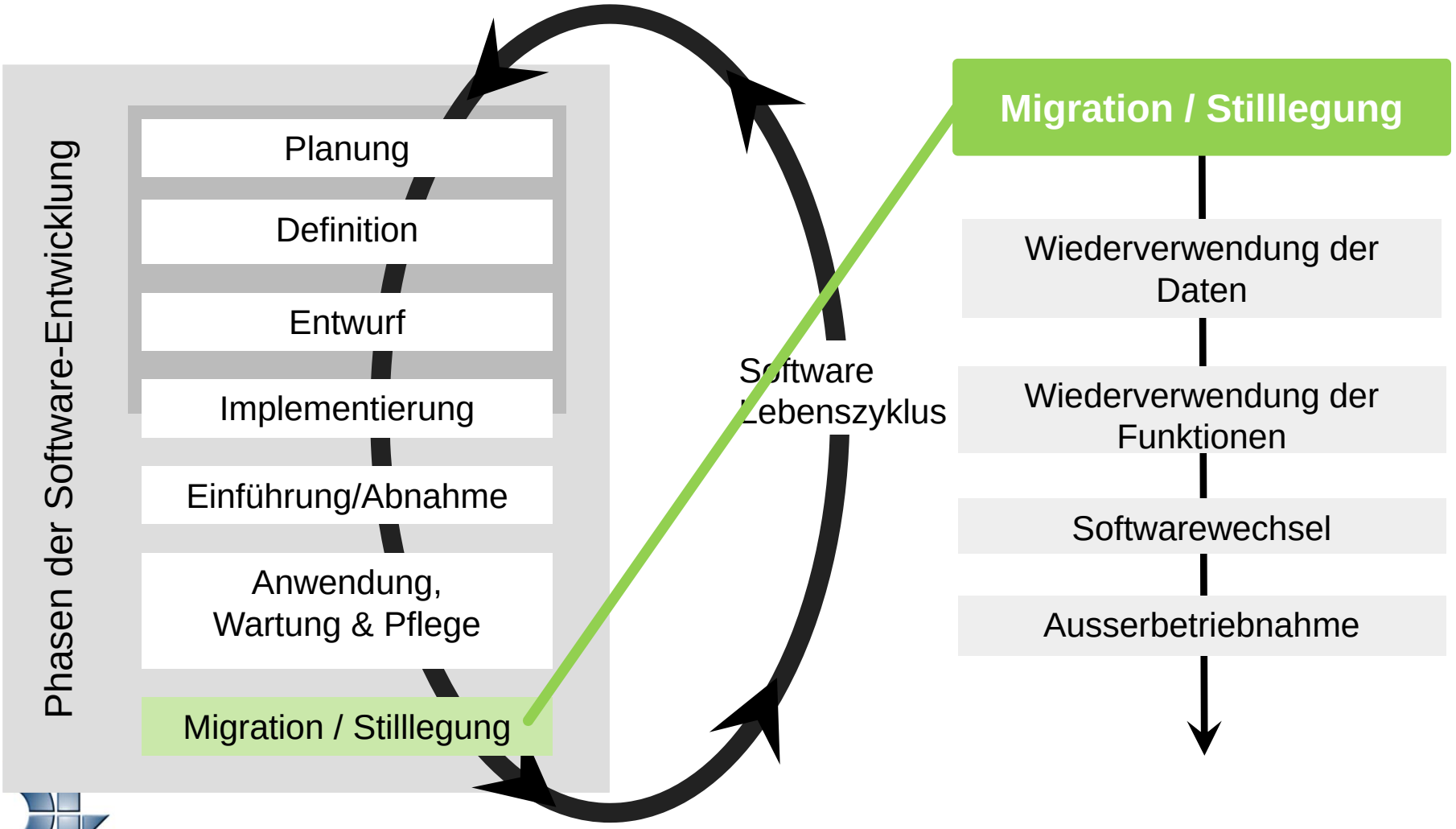
Skript S. 70 ff

Die Anwendungs-, Wartungs- und Pflegephase



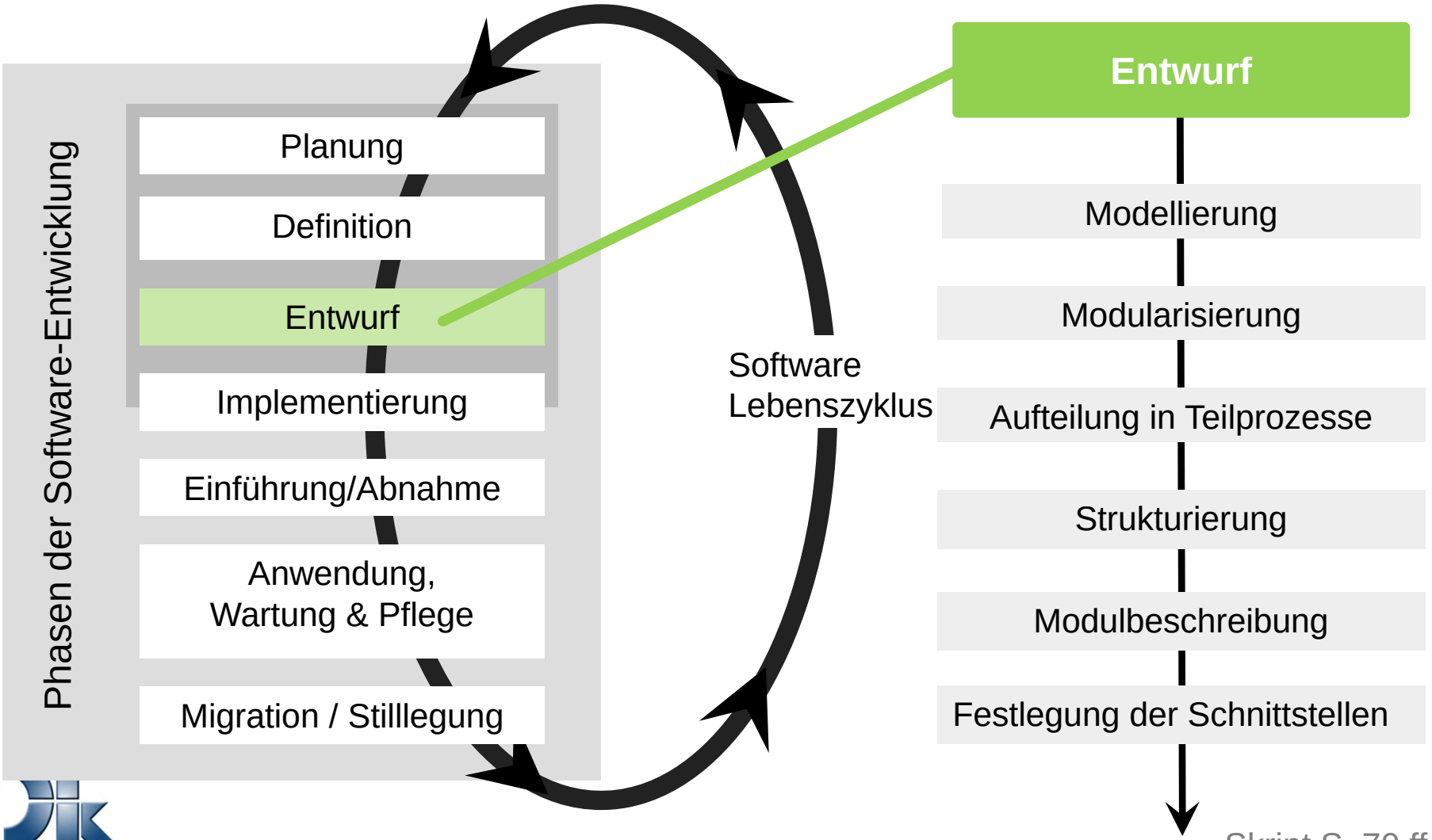
Skript S. 70 ff

Die Migrations- und Stilllegungsphase



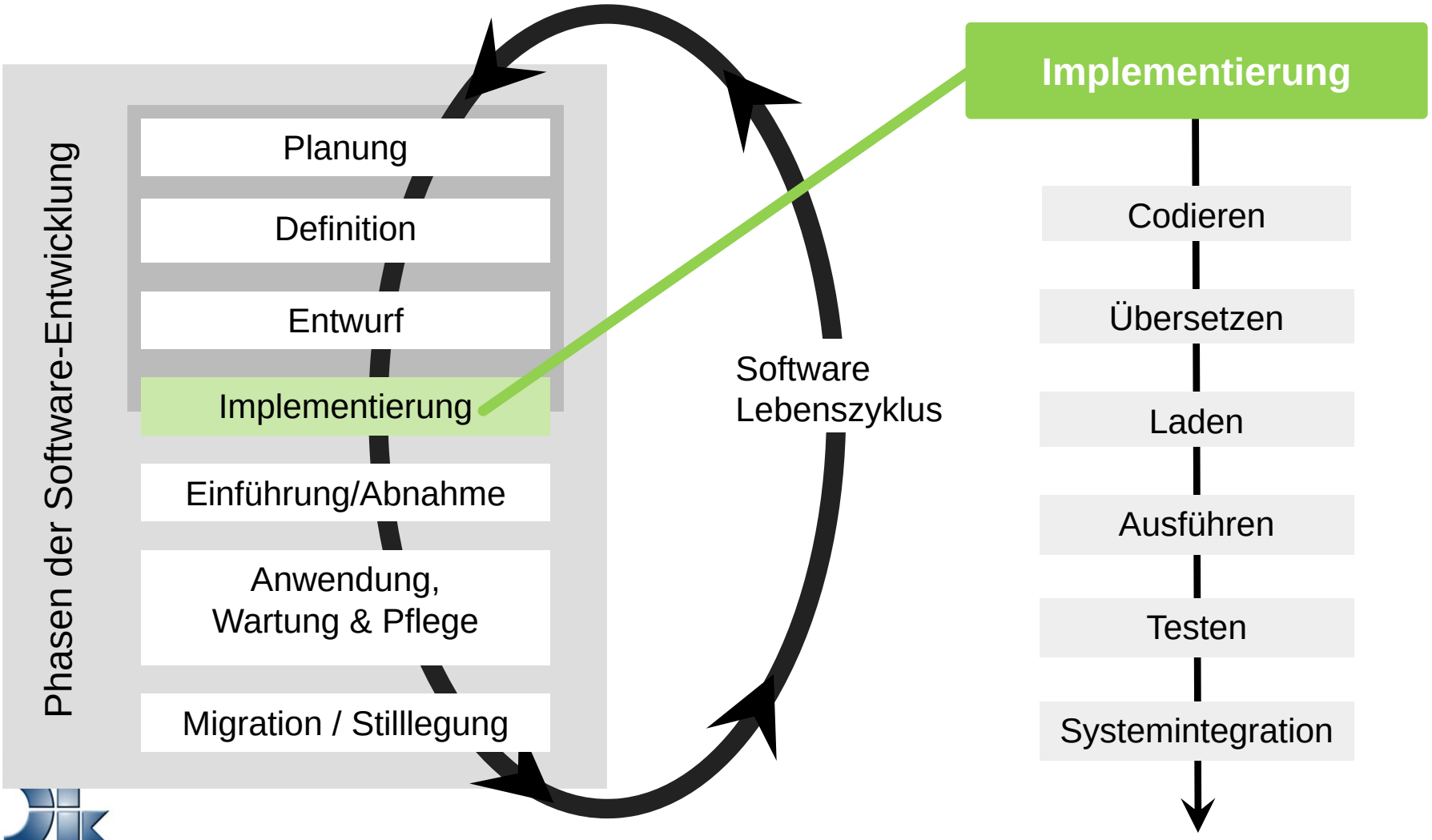
Skript S. 70 ff

Die Entwurfsphase



Skript S. 70 ff

Die Implementierungsphase



Skript S. 70 ff

Was ist ein Programm?

Skript S. 33 f

Grundsätzlicher Aufbau eines Programmes

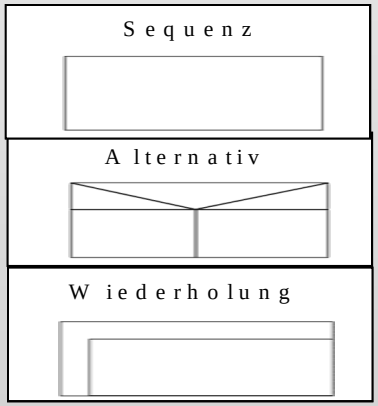
TECHNISCHE UNIVERSITÄT DARMSTADT

Prozedural

Variable, definiert über Datentypen

A = [....]

Verarbeitungskonstrukte



Einführung in Variablendeklaration

Skript S. 34

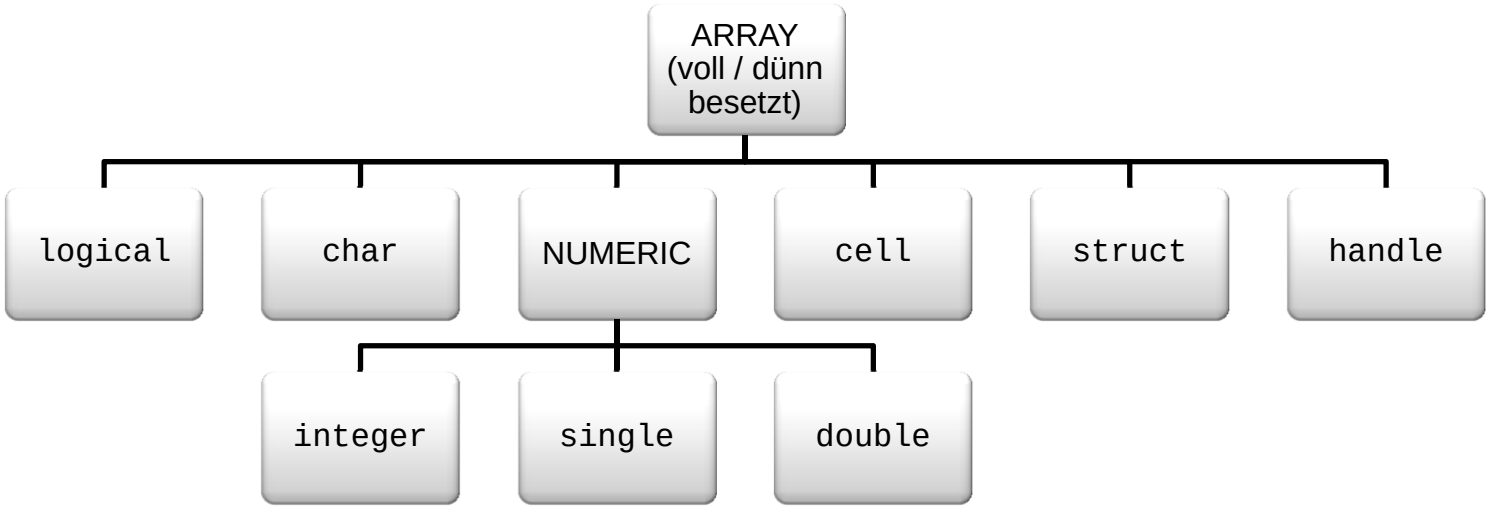
Variablendeklaration in Matlab: Namenskonvention und Datentypen

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Implizite Deklaration der Variablen bei der Initialisierung**
  - Durch eine Wertzuweisung wird der Datentyp der Eingabe von MatLab interpretiert und eine entsprechende Variable erzeugt.

- **Regeln:**
  - Unterscheidung zwischen Groß- und Kleinbuchstaben (case sensitiv)
  - Ein Variablenname darf keine Sonderzeichen außer dem Unterstrich ‚_‘enthalten.
  - Das erste Zeichen muss ein Buchstabe sein.
  - Vordefinierte Funktionen sollten nicht überschrieben werden mit gleichnamigen Variablen.

- **Die wichtigsten fundamentalen Datentypen in Matlab:**



Einführung in Programmkonstrukte

Skript S. 33

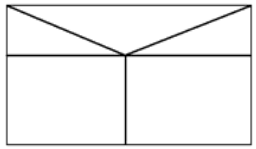
Bausteine eines Programms

TECHNISCHE UNIVERSITÄT DARMSTADT

Sequenz

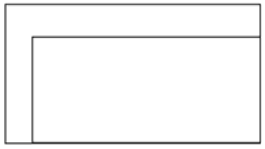
Erst wenn eine Anweisung abgearbeitet ist, darf mit der nächsten begonnen werden, was eine Parallelverarbeitung ausschließt. Diese sequentielle Arbeitsweise wird in den meisten Programmiersprachen durch das Unter- bzw. Hintereinanderschreiben der einzelnen Befehle erreicht.

Alternativ



Aufgrund von logischen Bedingungen kann eine Wahl zwischen mehreren Teilprozessen getroffen werden. Dies ermöglicht es, bestimmte Teilprozesse in Abhängigkeit des Zustandes der bearbeiteten Objekte auszulassen oder einzufügen.

Wiederholung



Ein bestimmter Teilprozess wird mehrfach abgearbeitet. Der Zustand der beteiligten Objekte lässt sich dabei durch Anweisungen ändern. Die Anzahl der Wiederholungen kann somit vom Zustand der Objekte abhängig gemacht werden, oder es wird eine feste Anzahl von Wiederholungen vorgegeben.

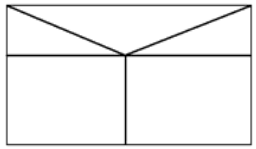
Skript S. 33

Programmkonstrukte

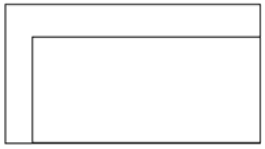
TECHNISCHE UNIVERSITÄT DARMSTADT

Sequenz

Alternativ



Wiederholung

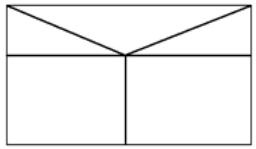


Programmkonstrukte

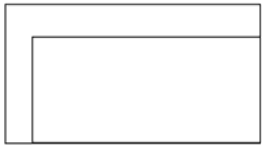
TECHNISCHE UNIVERSITÄT DARMSTADT

Sequenz

Alternativ

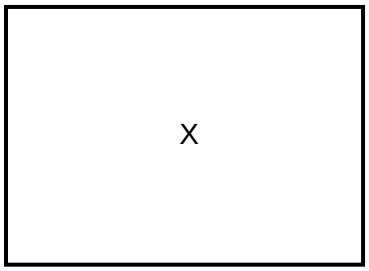
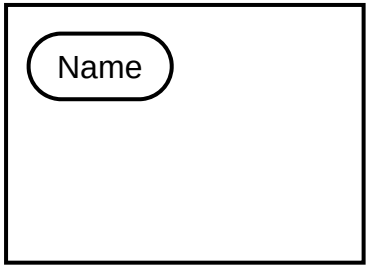
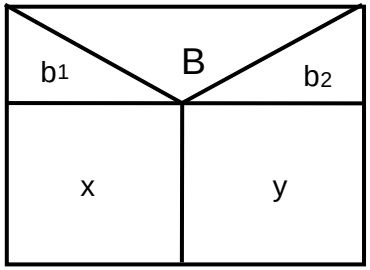
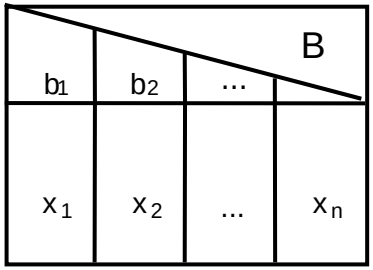
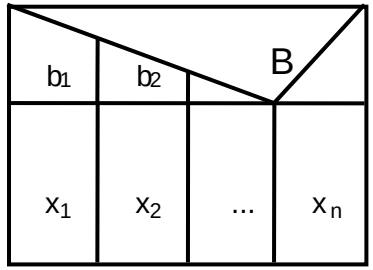
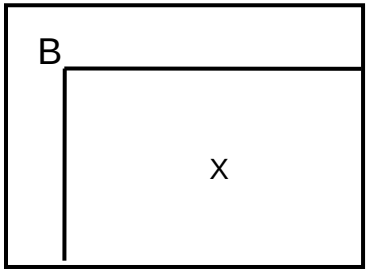
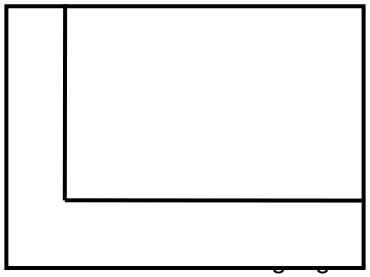
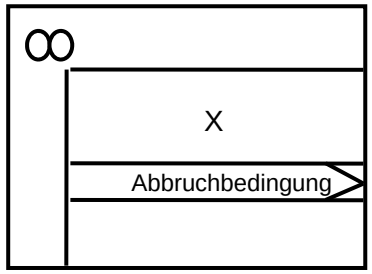


Wiederholung



Die Symbole eines Struktogrammes

TECHNISCHE UNIVERSITÄT DARMSTADT

	Grundtyp	1. Modifikation	2. Modifikation
Sequenz	 Folge (Sequenz)	 Prozeduraufruf	
Auswahl	 Verzweigungen je nach Bedingung (x oder y können leer sein)	 mehrfache Verzweigung	 mehrfache Verzweigung (keine Bedingung für die letzte Auswahl)
Wiederholung	 B = Ausführungsbedingung (Überprüfung zu Beginn)	 B = Abbruchbedingung (Überprüfung am Ende)	 Unendliche Schleife mit integrierter Abbruchbedingung

Skript S. 81 ff

Beispiel Messwertprüfung - Problembeschreibung

TECHNISCHE UNIVERSITÄT DARMSTADT

Problemstellung:

In der Qualitätssicherung sollen Bauteile geprüft werden. Dazu soll ein Programm entwickelt werden, welches die Bauteilabmessungen erfasst und prüft, ob diese in einem gegebenen Toleranzbereich liegen.

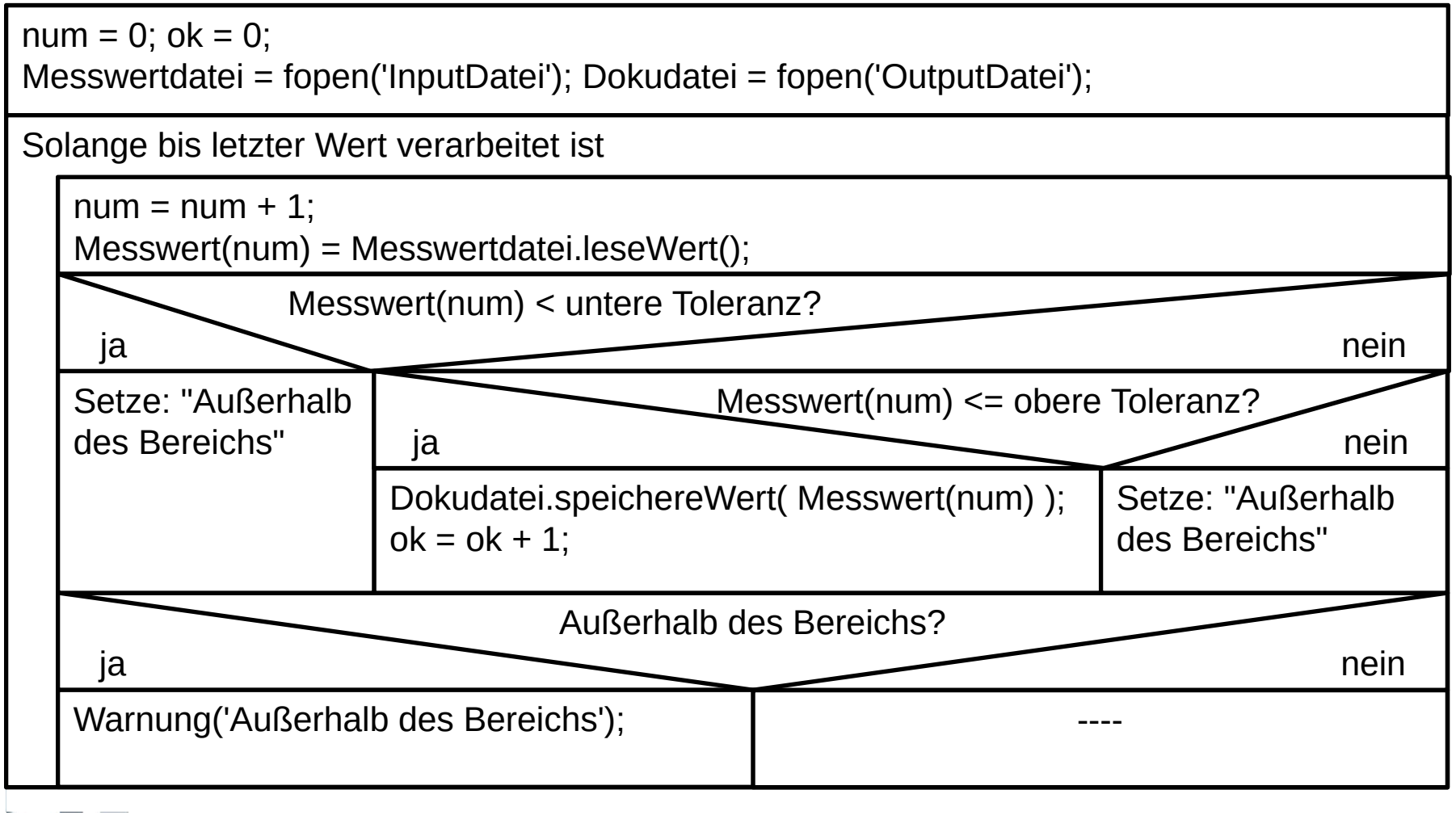
Die vorgegebenen Messwerte sollen aus einer Datei eingelesen werden und es soll jeweils eine Warnung ausdruckt werden, falls ein Wert nicht im vorgegebenen Bereich liegt. Ansonsten sollen die Messwerte zur Dokumentation abgespeichert werden, wenn sie im Toleranzbereich liegen.

Beispiel Messwertprüfung – Methode „verarbeiteMesswert“ als Struktogramm

Struktogramm - Symbolik

TECHNISCHE UNIVERSITÄT DARMSTADT

verarbeiteMesswert



Struktogramm – Symbolik –Alternative Lösung

TECHNISCHE UNIVERSITÄT DARMSTADT

verarbeiteMesswert

num = 0; ok = 0;
Messwertdatei = fopen('InputDatei'); Dokudatei = fopen('OutputDatei');

Solange bis letzter Wert verarbeitet ist

num = num + 1;
Messwert(num) = Messwertdatei.leseWert();

Messwert(num) < untere Toleranz?
oder
Messwert(num) > obere Toleranz?

ja

Warnung('Außerhalb des Bereichs');

nein

Dokudatei.speichereWert(Messwert(num));
ok = ok + 1;

Konstrukte in MatLab

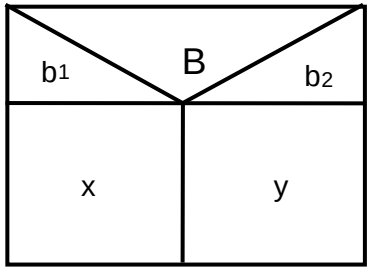
TECHNISCHE UNIVERSITÄT DARMSTADT

- Sequenz

```
v = g = 9;
t = [0:0.01:1];
x = v * t;
y = 0.5*g*t.^2 + v * t;
plot(x,y,'r')
```

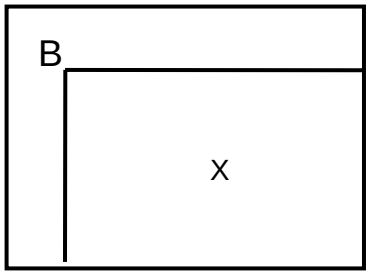
- Auswahl (`if, switch`)

```
if (a>=0)
      disp('positive');
else
      disp('negative');
end
```

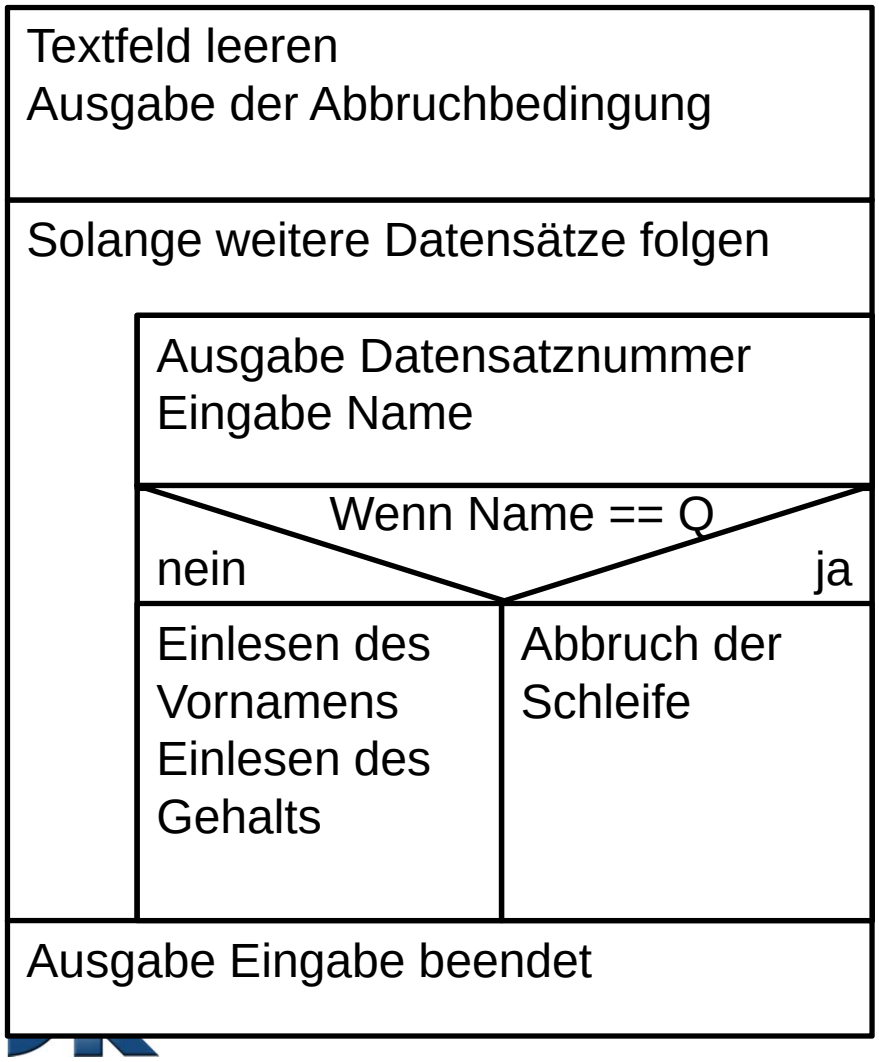


- Wiederholung (`for, while`)

```
for ii=1:5
      disp('Hallo')
end
```



Arbeitnehmerdatenbank




```
clc
disp(' ')
disp('Abbruch der Eingabe mit ''q'' ')

Arbeitnehmer = struct('');

ii = 0;
schalter = true;
while schalter
    ii = ii+1;
    disp(' ')
    disp(['Datensatznummer: ' num2str(ii)])

    S = input('Name   : ','s');
    if (S=='q') | (S=='Q')
        schalter = false;
    else
        Arbeitnehmer(ii).Name = S;
        Arbeitnehmer(ii).Vorname = ...
                    input('Vorname : ','s');
        Arbeitnehmer(ii).Gehalt  = ...
                    input('Gehalt  : ','s');
    end
end

disp('Eingabe beendet!,)
```

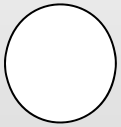
Symbole eines Programmablaufplans

TECHNISCHE UNIVERSITÄT DARMSTADT

Grenzstelle

Kennzeichnet den Anfang bzw. das Ende des Programmablaufplans

Marken



Kennzeichnet Verzweigungs-punkte. Hat mehrere Eingänge aber nur einen Ausgang

Ein-Ausgabeanweisung

Hat einen Eingang und einen Ausgang

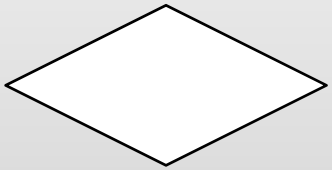
Ablauflinien

Verbinden Symbole und legen durch Pfeilrichtung den Ablauf fest

Anweisungsteil

Hat einen Eingang und einen Ausgang

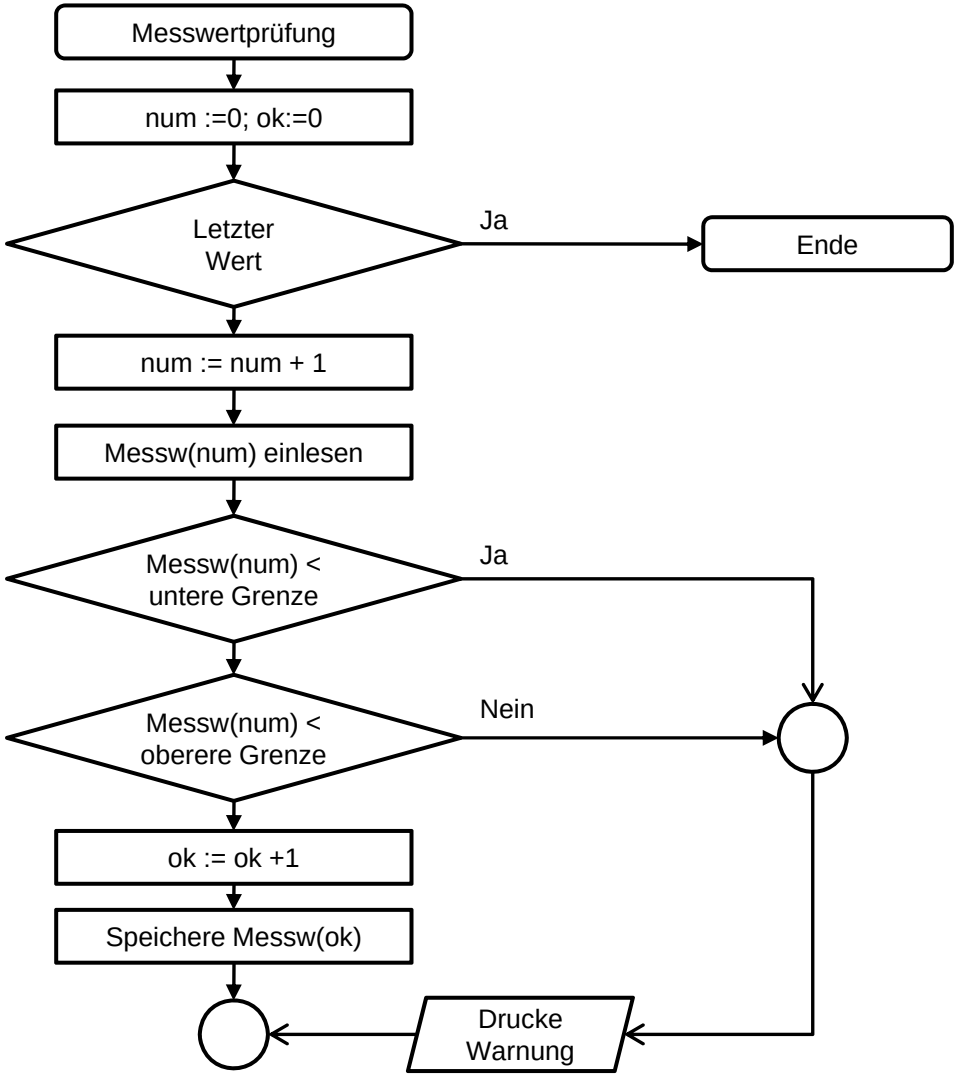
Verzweigung mit Bedingung



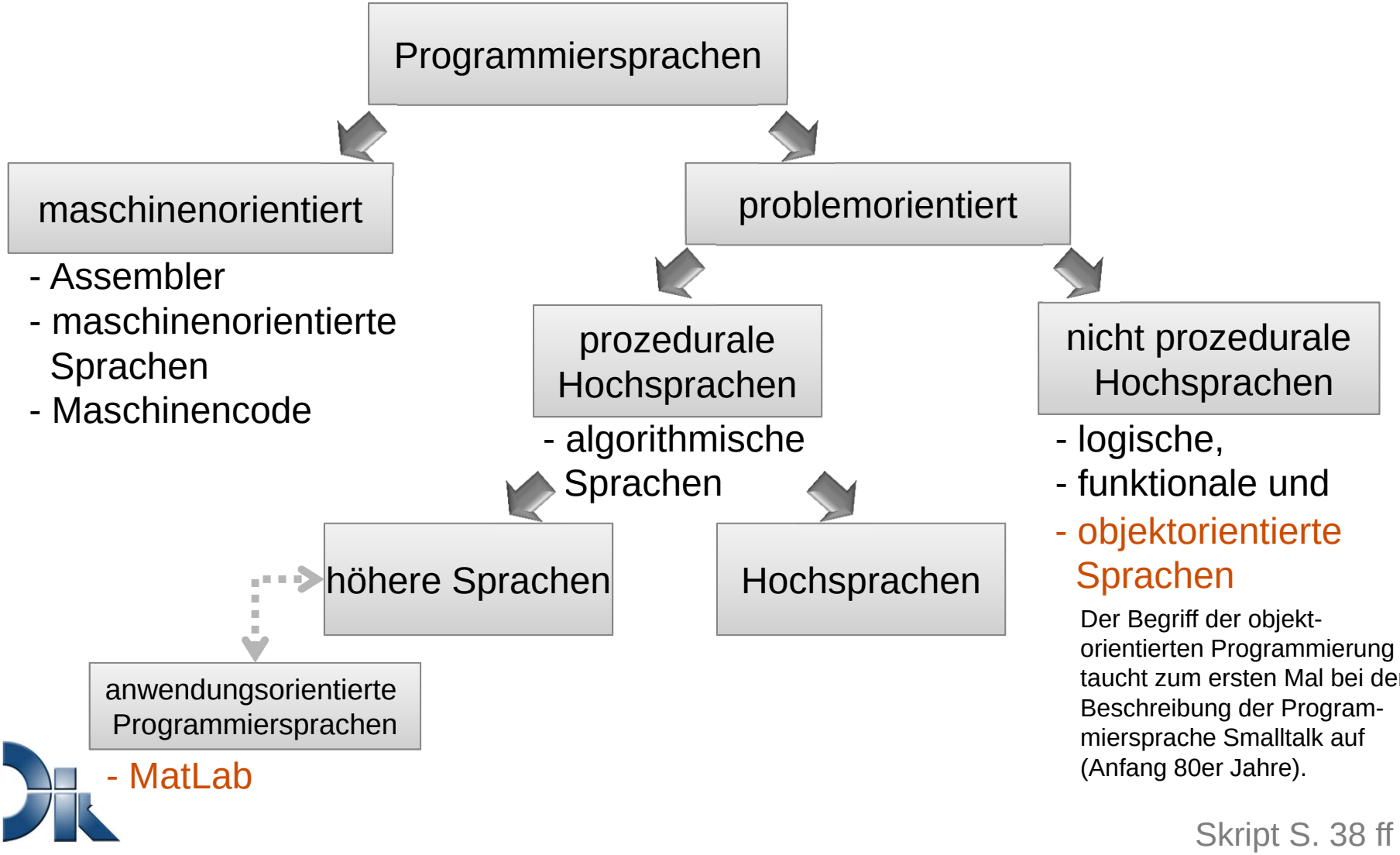
Die beiden Ausgänge sind mit ja bzw. nein zu kennzeichnen

Skript S. 80 f

Beispiel Messwertprüfung als Programmablaufplans



Arten von Programmiersprachen



Der Begriff der objekt-orientierten Programmierung taucht zum ersten Mal bei der Beschreibung der Program-miersprache Smalltalk auf (Anfang 80er Jahre).

Skript S. 38 ff

Skript S. 49 ff

Bausteine eines Programms

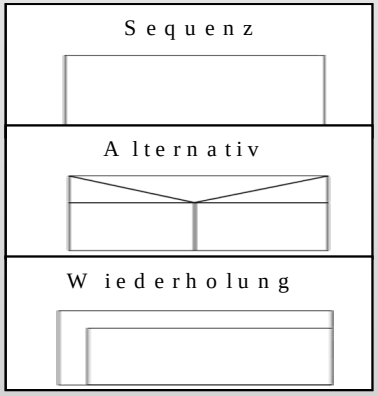
TECHNISCHE UNIVERSITÄT DARMSTADT

Prozedural

Variable, definiert über Datentypen

A = [….]

Verarbeitungskonstrukte



Objektorientiert

Objekt: Rad

Attribute

Raddurchmesser = 19“

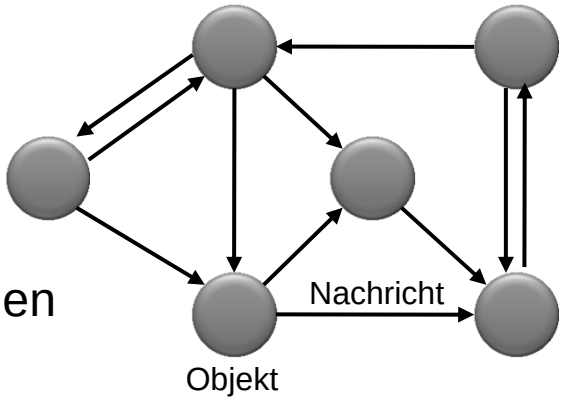
Operationen

Drehen um die Achse
Einlenken
….

Objektorientierte Programmierung

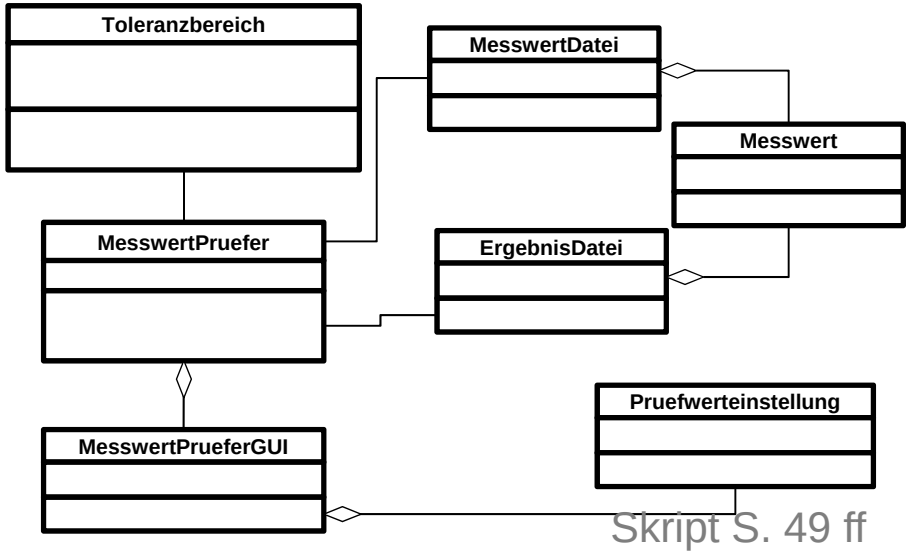
TECHNISCHE UNIVERSITÄT DARMSTADT

Objektorientierte Programmierung ist die Identifizierung von einer Menge von (dynamischen) *Objekten* und deren *Interaktionen*.



Sie unterstützt eine Sichtweise zur Behandlung von Problemen in der :

- real vorhandenen *Strukturen* im Vordergrund stehen (im Gegensatz zu den *Verfahren*),
- diese in Objekte zerlegt und in bestimmten *Beziehungen* zueinander gebracht werden.

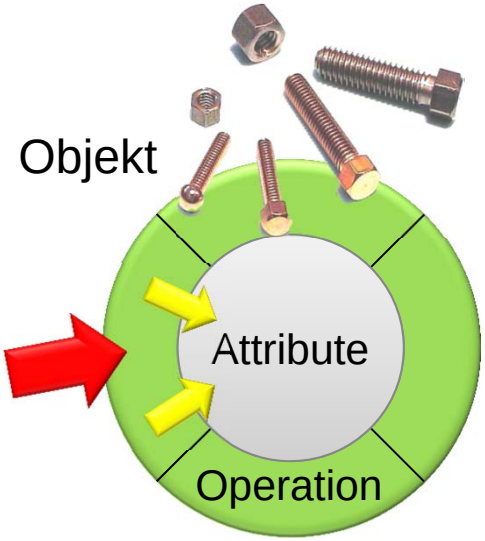


Skript S. 49 ff

Grundprinzipien des Objekts

TECHNISCHE UNIVERSITÄT DARMSTADT

- Ein **Objekt** ist eine in sich abgeschlossene Komponente mit einem gekapselten, i.allg. nicht direkt sichtbaren Zustand.
- Ein Objekt besteht aus **Attributen** und **Operationen**.
  - Die Attribute bilden den Zustand eines Objekt.
  - Der Zustand beeinflusst das Resultat der Operationen.
  - Das Verhalten eines Objektes wird charakterisiert durch die Operationen.
- Ein Objekt hat eine eindeutige ***Identität*** (Objektname)
  - Die Identität eines Objektes ist unabhängig von seinen anderen Eigenschaften.
  - Es können mehrere Objekte mit identischem Verhalten und inneren Zustand existieren (z.B. ***Autos***).



Skript S. 49 ff

Merkmale der objektorientierten Programmierung

Skript S. 49 ff und 55 ff

Beispiel für den Entwurf einer Klasse

TECHNISCHE UNIVERSITÄT DARMSTADT

Toleranzbereich
obere_grenze: double **untere_grenze: double**
prüfeToleranzB(): boolean **setzeToleranz(): void**

Name und Typ der Klasse

Attribute der Klasse

Operationen der Klasse

Name und Typ des Objektes

Attribute des Objektes

Operationen des Objektes

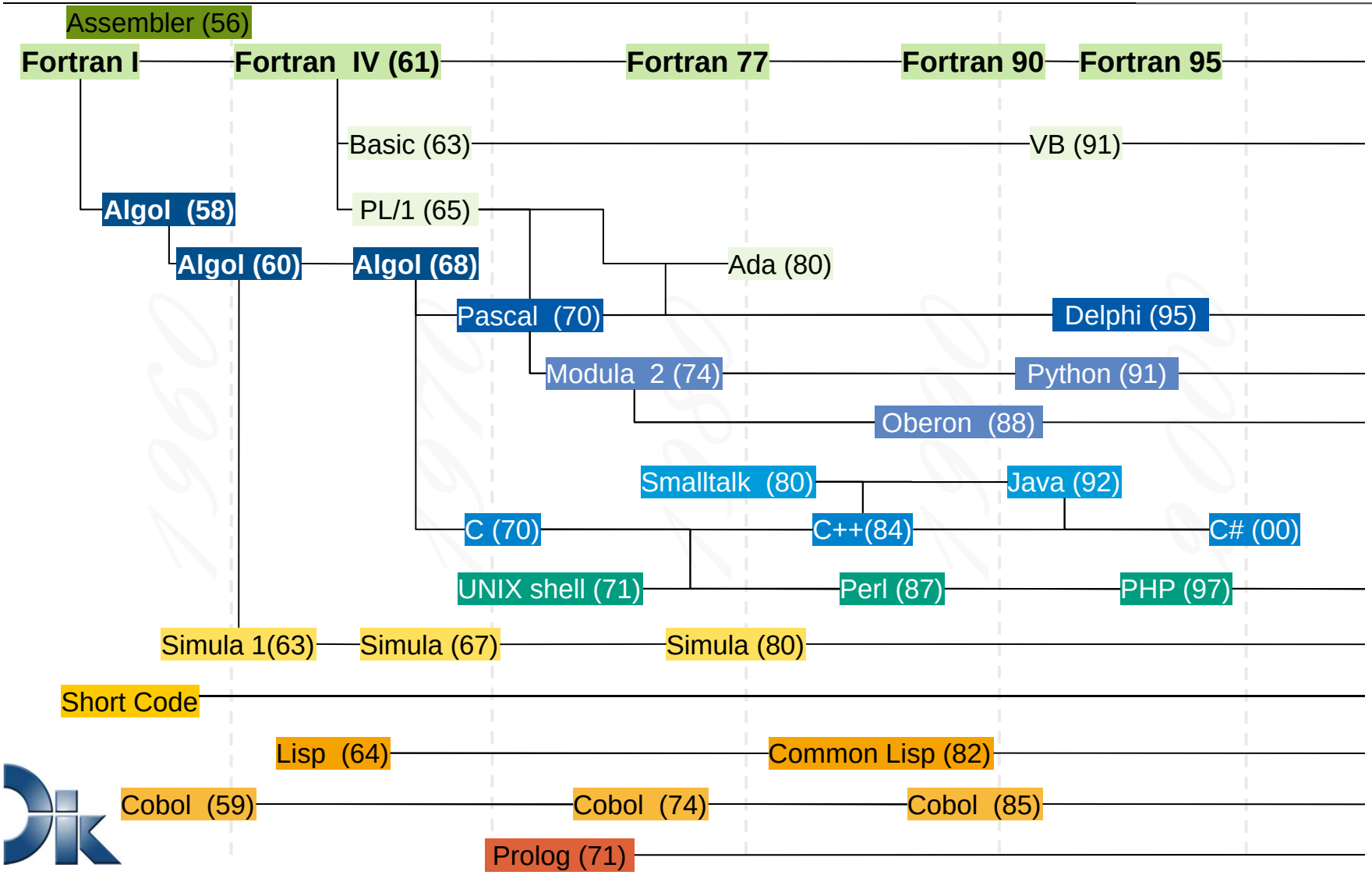
Toleranzbereich: TB1
obere_grenze: double= 10 **untere_grenze: double= -5**
prüfeToleranzB(): boolean **setzeToleranz(): void**

Übersicht Programmiersprachen

Skript S. 38 ff

Chronologische Entwicklung der Programmiersprachen

TECHNISCHE UNIVERSITÄT DARMSTADT



Skript S. 38