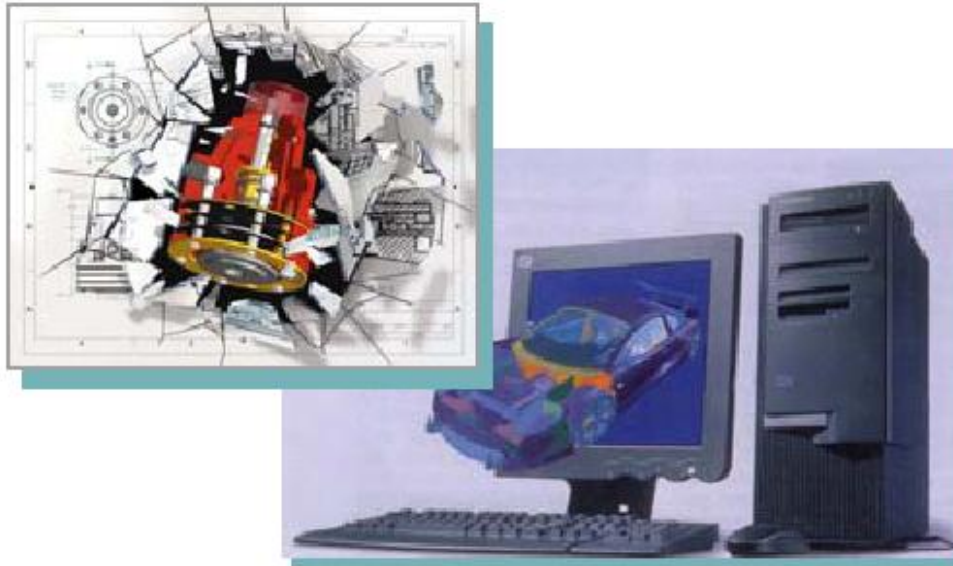




Vom Problem zum Programm

Einige Worte zum Software Engineering
Phasen der SW Entwicklung

Der Traum wurde wahr



“Denn es ist eines ausgezeichneten Mannes nicht würdig, wertvolle Stunden wie ein Sklave im Keller der einfachen Rechnungen zu verbringen. Diese Aufgaben könnten ohne Besorgnis abgegeben werden, wenn wir Maschinen hätten”.



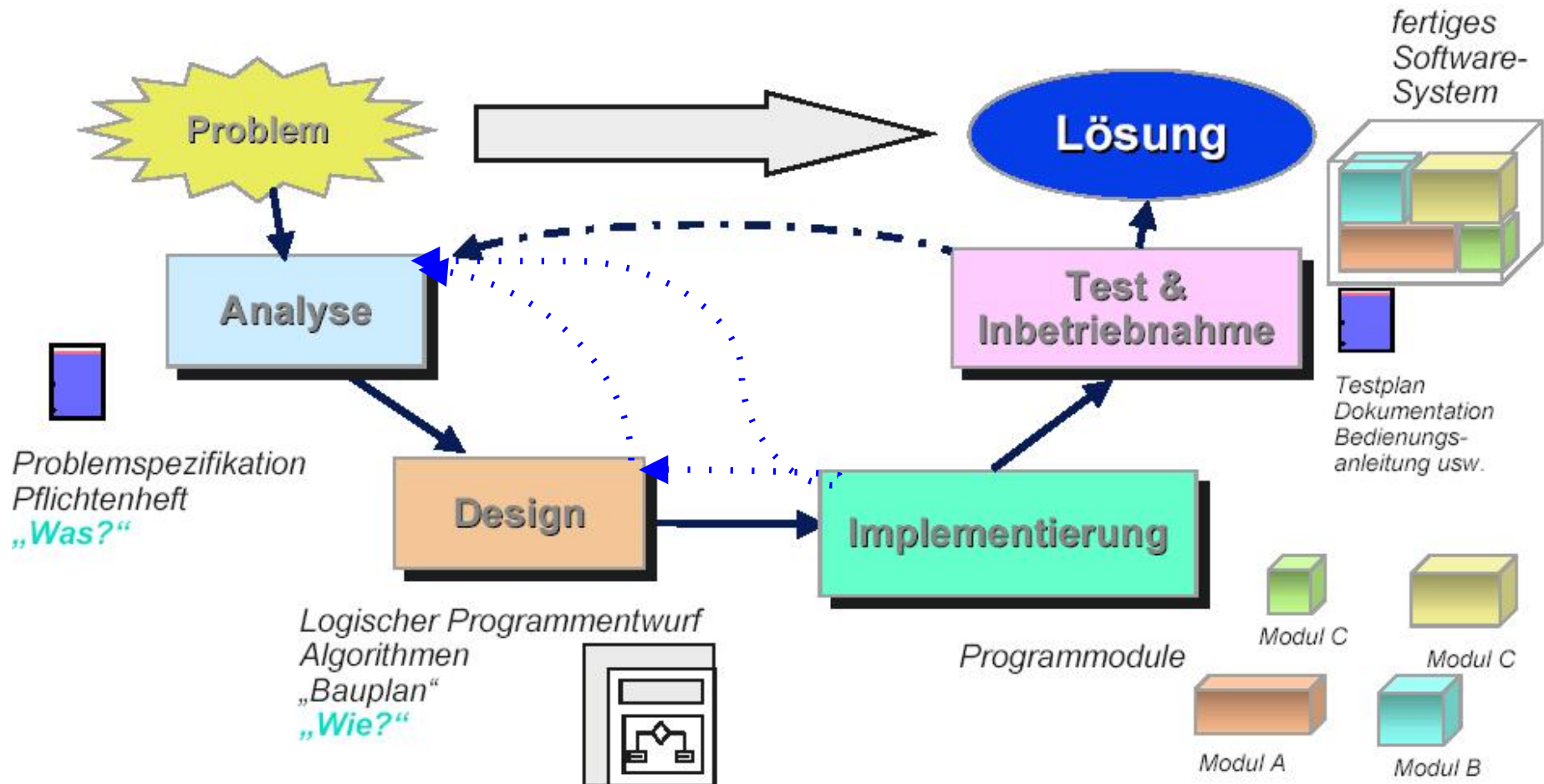
Gottfried Wilhelm Leibniz, 1646-1716

Die SW Entwicklung in der Krise

- Software zählt zu den komplexesten Artefakten der Menschheit
- 5 von 6 SW-Projekten sind nicht erfolgreich
- 1/3 aller SW-Projekte wird irgendwann abgebrochen
- Die restlichen Projekte benötigen ein vielfaches an Zeit/ Geld
- Die Entwicklungszyklen werden immer kürzer (time to market)

- Software Entwicklung braucht ingenieurmäßiges Vorgehen

Phasen der SW Entwicklung

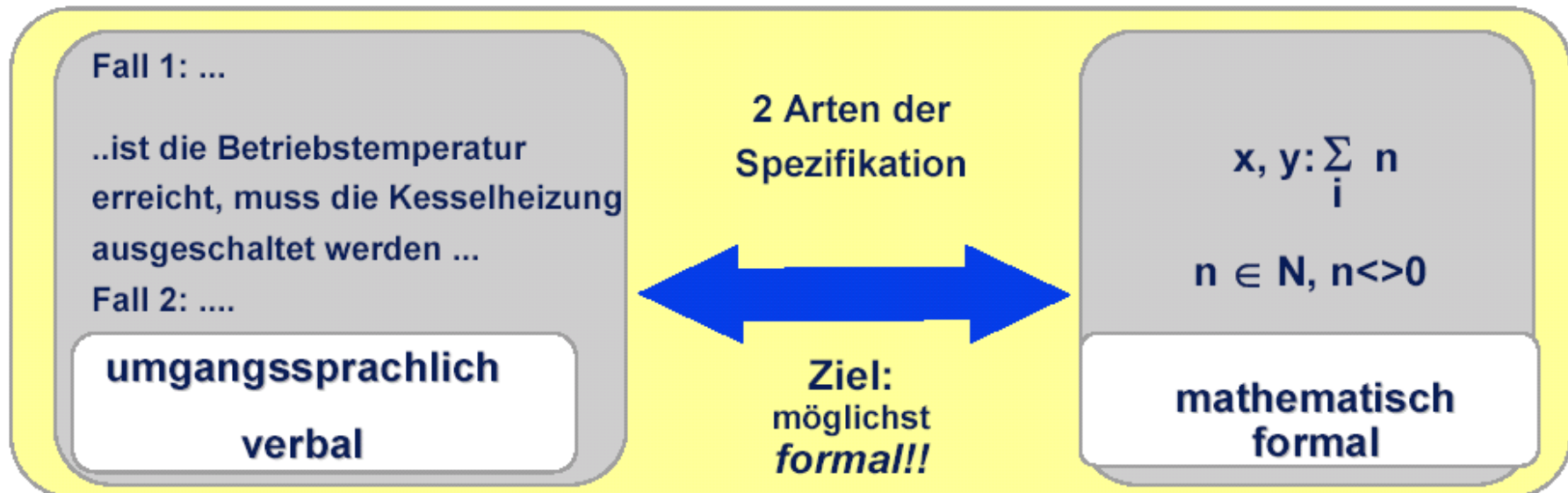


Wichtig ist:

- o Vollständigkeit
- o Eindeutigkeit
- o Widerspruchsfreiheit

Analyse- Phase

- Im Mittelpunkt steht die Frage nach dem WAS
- Ergebnis:
 - Problembeschreibung (Spezifikation)
 - Pflichtenheft (bei größeren Projekten „Vertragscharakter“)



Analyse- Phase : Beispiel

Problemstellung

- Im Vorlesungssaal befinden sich ca. 60 Studenten.
- Wie alt ist die jüngste Person?

Analyse- Phase : Beispiel

Problemstellung

- Im Vorlesungssaal befinden sich ca. 60 Studenten.
- Wie alt ist die jüngste Person?

Gegeben:

$a_1, a_2, \dots, a_n, n, a_i \in \mathbb{N}$

a_i : Alter der i-ten Person

n : Anzahl der Personen

Gesucht:

$m \in \mathbb{N}$

Mit m ist Alter der jüngsten Person(en)

Funkt. Zusammenhang:

$m = \min(a_1 \dots a_n)$

Design - Phase

- Im Mittelpunkt steht die Frage nach dem WIE?
- Ergebnis
 - Lösungsweg in Form einer möglichst allgemeinen Rechenvorschrift (Kochrezept)



Algorithmus

Definition „Algorithmus“:

Ein Algorithmus ist eine Verarbeitungsvorschrift, die so präzise formuliert ist, dass sie von einem mechanisch oder elektrisch arbeitenden Gerät abgearbeitet werden kann. Er besteht aus einer Reihe von Regeln für ein endliches Verfahren, um eine spezifische Ausgabe von spezifischen Eingaben zu erhalten.

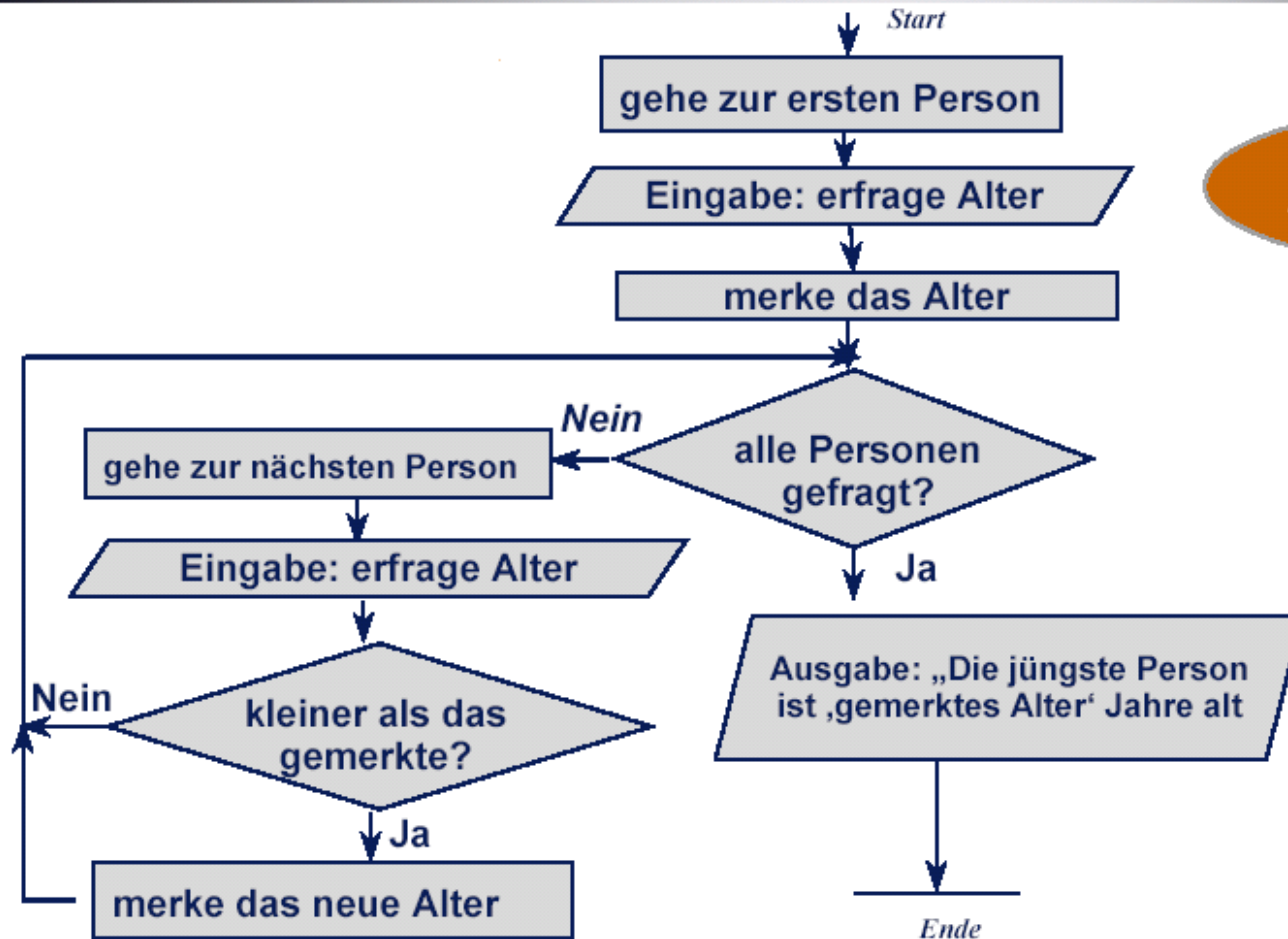
Design - Phase

1. Version eines Kochrezepts

1. Gehe zur ersten Person
2. Erfrage das Alter
3. Merke das Alter
4. Solange noch nicht alle Personen gefragt sind wiederhole die Schritte:
 1. Gehe zur nächsten Person
 2. Erfrage das Alter
 3. Wenn das Alter kleiner als das gemerkte Alter ist, merke das neue Alter
5. Jüngste Person ist „gemerkttes Alter“ Jahre alt!



Design – Phase : Flussdiagramm



Algorithmus

2. Version
eines Kochrezepts

Design-Phase

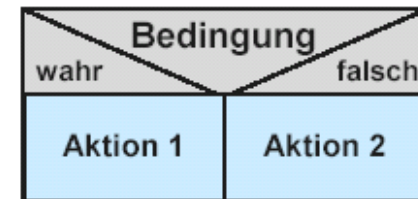
Grundelemente des Nassi-Shneidermann-Diagramms



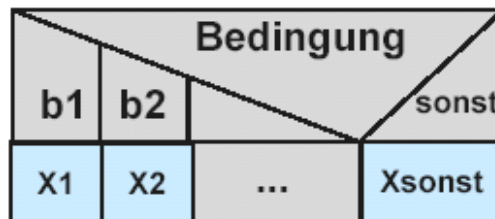
Anweisung
Strukturblock



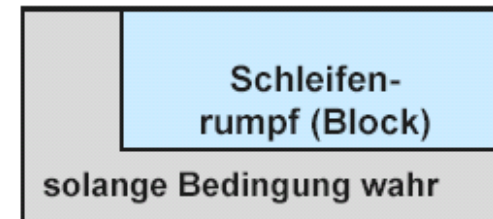
Folge (Sequenz)
von Anweisungen



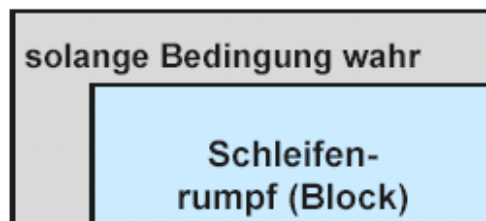
Bedingte Anweisung



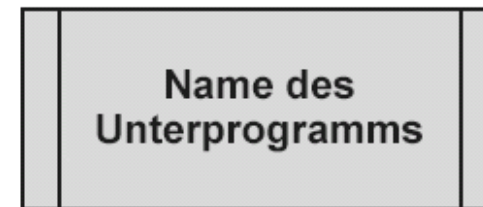
mehrfache Verzweigung



Schleife, Prüfung nach dem Rumpf



Schleife, Prüfung vor dem Rumpf

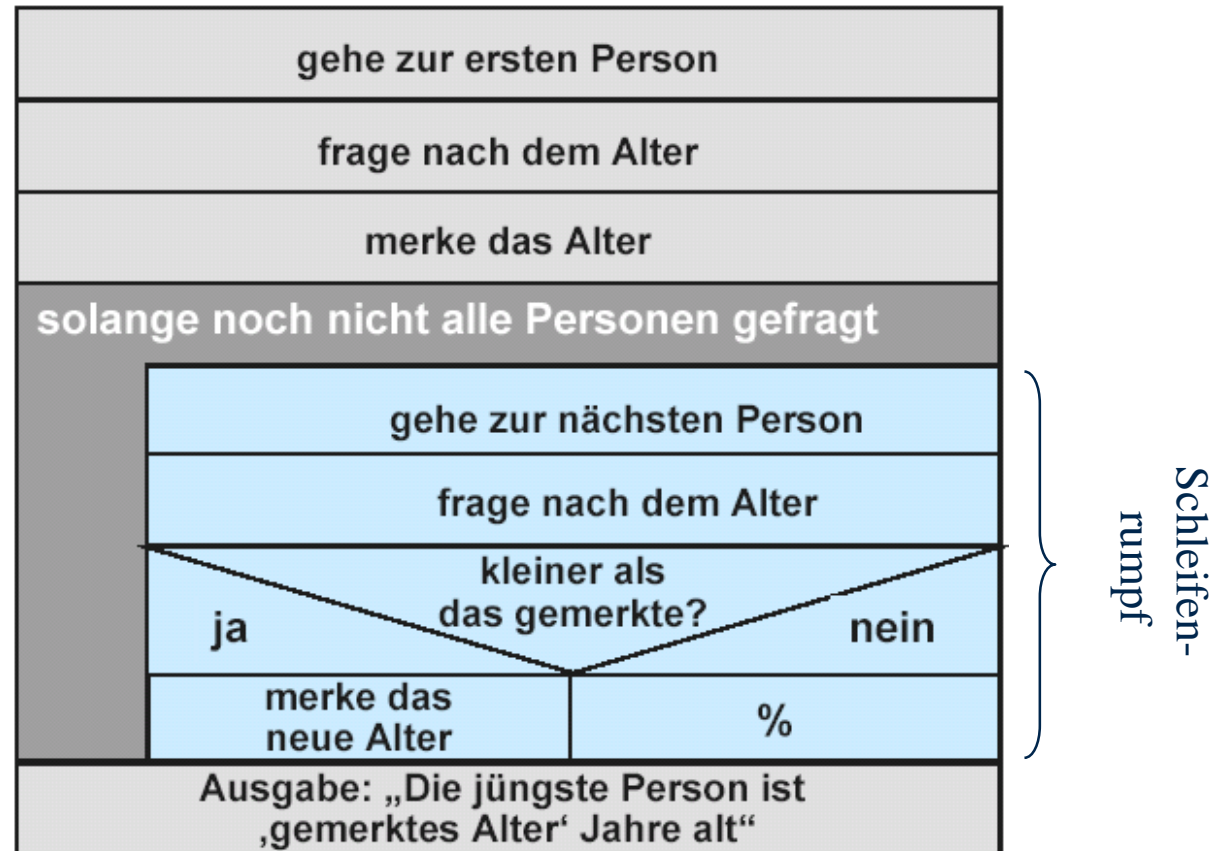


Unterprogrammaufruf

Design-Phase

3. Version

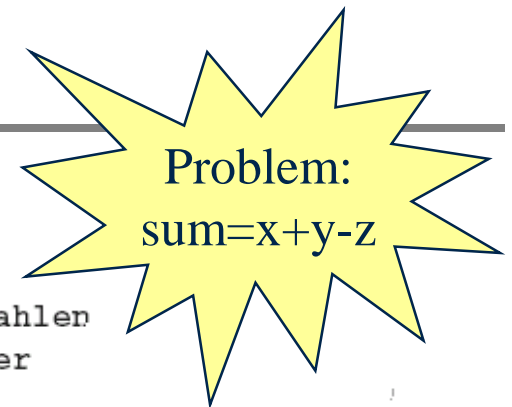
Nassi-Shneidermann-Diagramm



Implementierungsphase

- **Ausgangspunkt:**
 - Softwarearchitektur & Spezifikation der Systemkomponenten
- **Ergebnis**
 - (dokumentierte) Programme
 - Testplanung und Testprotokolle
- **Implementierung umfasst**
 - Entwicklung von Klassen, Datenstrukturen, Algorithmen
 - Kommentieren von Lösungsschritten (Zeit, Platz,...)
 - Umsetzung in Programmiersprache
 - Test bzw. Testplanung

Implementierungsphase



“höhere Ebene” [intuitiv]

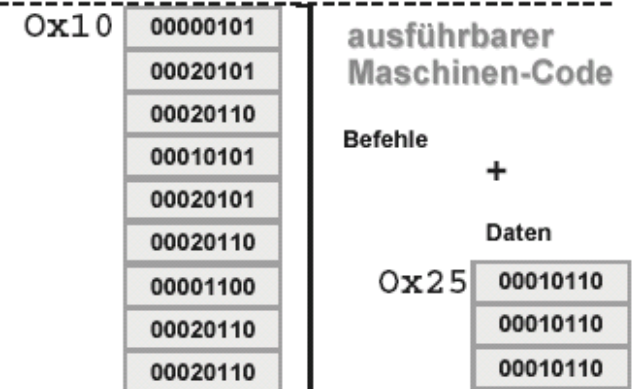
x, y, z und sum sind ganze Zahlen
hol dir x,y und z vom Bediener
über Tastatur ab
sum ergibt sich dann aus x+y-z
gib den Summenwert sum als Ergebnis
auf dem Bildschirm aus

Assembler-Ebene

```
Assembler-Programm
start: store  sum,x;
      add    sum,y;
      sub    sum,z;

sum   :      ?
x     :      ?
y     :      ?
z     :      ?
```

Maschinen-Ebene



Testphase

- Ziele:
 - Validation: System entspricht den Kundenwünschen?
 - Verifikation: System entspricht der Spezifikation?
- Testen ist ein NICHT unerheblicher Anteil
 - Ca. 30% der Entwicklungskosten
 - Daher Testen bereits in vorherigen Phasen integrieren
 - „early prototyping“
- Besser: Tester ≠ Entwickler

Testphase

- Testen einer Basiseinheit (klassisch: Prozedur, Module OO: Klasse)
 - Bereits in der Implementierungsphase (durch Entwickler)
- Integrationstest
 - Zusammenspiel der Einheiten, der Prozesse
- Systemtest
 - Gesamtsystem incl. Umfeld
- Regressionstest
 - Z.B. nach Änderungen / Weiterentwicklungen
- Anforderungsbasiert Test
 - Überprüfung einzelner Anforderungen / Prozesse

Testphase

- **Operationstest**
 - Systemtest im Normalbetrieb
- **Performancetest / Lasttest / Überlasttest**
 - Überprüft Laufzeitverhalten für einzelne Use-cases
 - Welche Reaktion zeigt gesamtsystem bei Überlast (Crash, sanftes Herunterfahren)
 - Meistens Tools erforderlich (z.B. Loadrunner,...)
- **Backup / Recovery / High Availability**
 - Test der definierten Verfahren

Testphase

- Ergonomietest
 - Interaktion Mensch / Maschine (oft schon durch Prototyp möglich)
- Test der Dokumentation / Hilfe
 - Stimmt die Benutzerdokumentation mit den impl. Funktionen überein?
- User Acceptance Test
 - Durch Auftraggeber (α -Test)
 - Bei Standard-Software: z.B. mit Pilot-Kunden (β -Test)

Phasen der SW Entwicklung

