

# PAP / Programmablaufpläne

## Den Diagrammtyp PAP beschreibend einordnen, Vor- und Nachteile benennen:

Ein Programmablaufplan (PAP) ist eine grafische Darstellung eines Algorithmus oder eines Programmablaufs. Es ist ein Diagramm, das den logischen Fluss eines Programms oder eines Prozesses darstellt.

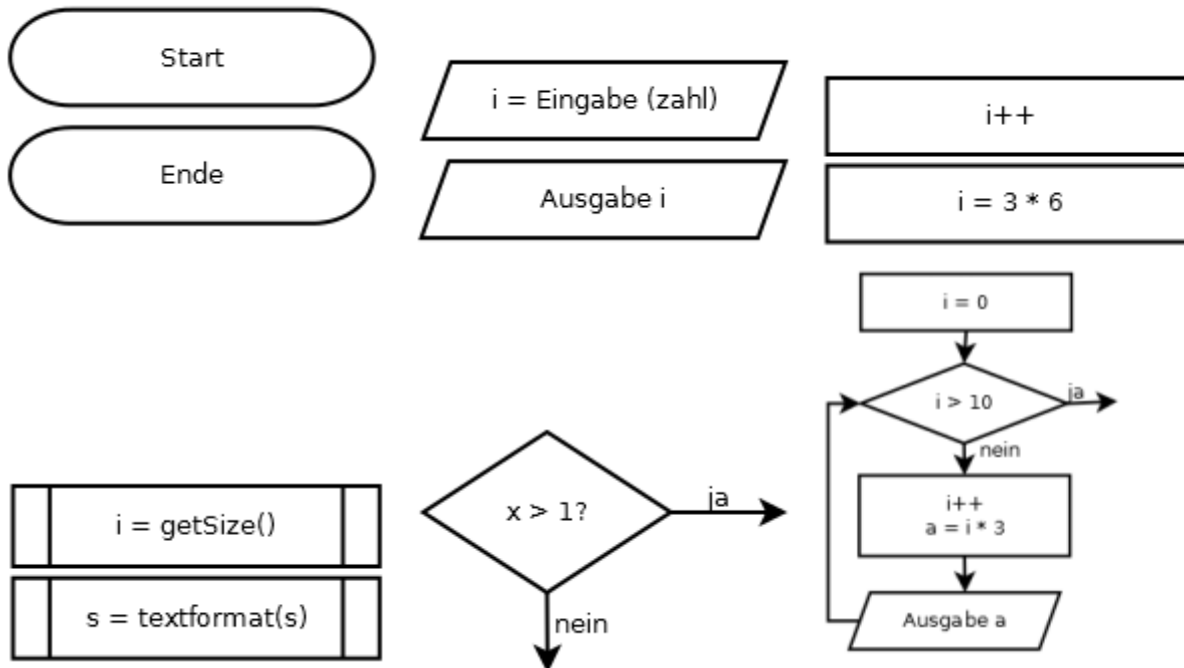
### Vorteile:

- Klare Darstellung des Ablaufs
- Strukturierte Planung
- Bessere Kommunikation in einem Team
- Es kann helfen bei der Wartung

### Nachteile:

- Je komplexer, desto unübersichtlicher
- Begrenzte Darstellungsmöglichkeit

## Programmablaufpläne lesen, Schreibtischtest, entwickeln und zeichnen:



# Struktogramme

## Den Diagrammtyp Struktogramm beschreibend einordnen, Vor- und Nachteile benennen:

Ein Struktogramm ist eine grafische Darstellung eines Programmablaufs ähnlich wie ein Programmablaufplan (PAP). Es wird häufig in der Softwareentwicklung verwendet, um den logischen Fluss eines Programms zu beschreiben.

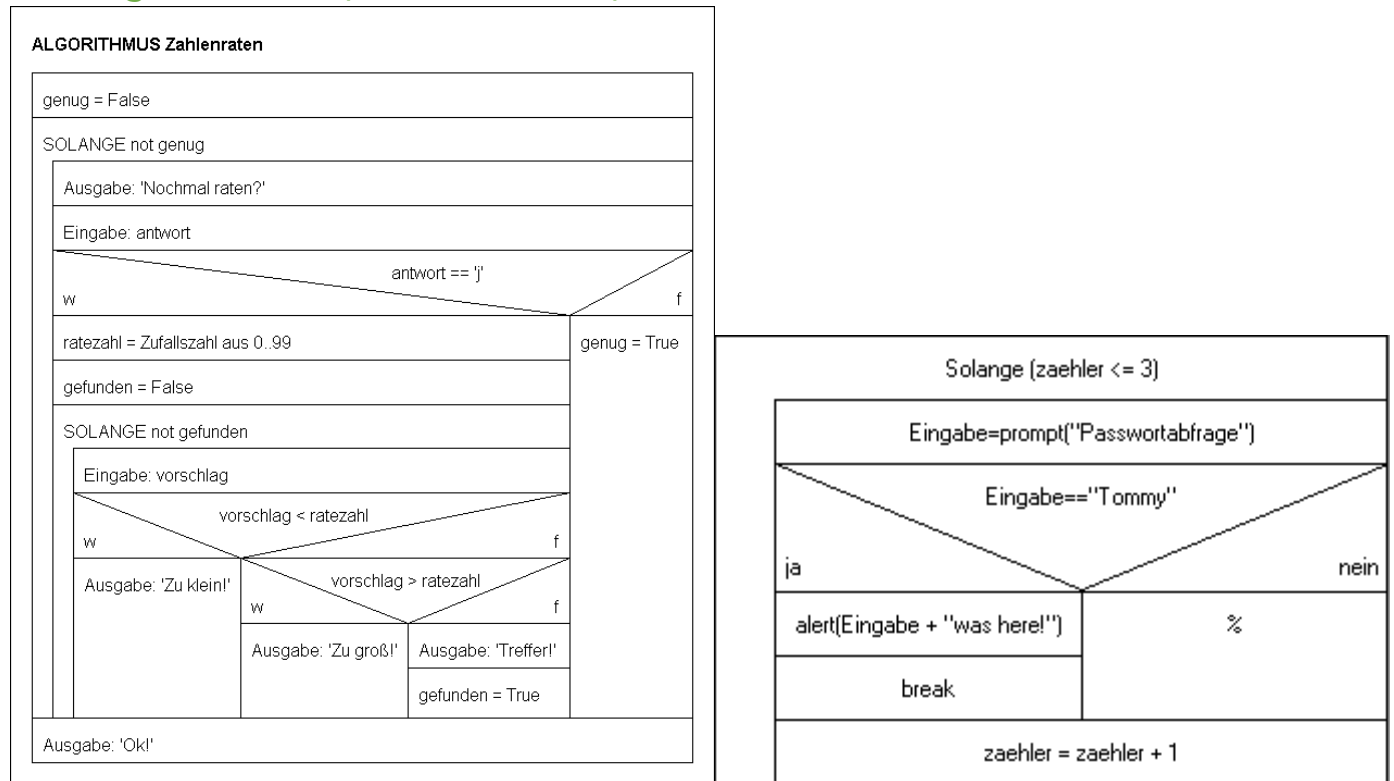
### Vorteile:

- Klare Struktuiierung
- Effektive Planung
- Übersichtlich, kann helfen bei der Wartung

### Nachteile:

- Begrenzte Flexibilität
- Erfordert eine gewisse Lernkurve

## Struktogramme lesen, Schreibtischtest, entwickeln und zeichnen



# Theorie-Wissen

## Fragen zu den theoretischen Grundlagen / zu den Fachbegriffen beantworten:

### Compiler:

#### Was ist ein Compiler?

Ein Compiler ist ein Softwareprogramm, das Quellcode in Maschinencode oder eine Zwischensprache wie Bytecode, übersetzt.

#### Wie funktioniert ein Compiler?

Ein Compiler generiert einen optimierten Zwischencode oder Maschinencode, der von dem Computer ausgeführt werden kann.

#### Vorteile eines Compilers:

- Der erzeugte Maschinencode ist i.d.R. effizienter als der ursprüngliche Quellcode.
- Unabhängig, solange ein Compiler für die Zielsprache verfügbar ist.
- Syntaxfehler und einige Semantikfehler im Quellcode werden erkannt.

### Interpreter:

#### Was ist ein Interpreter?

Ein Interpreter führt Code Zeile für Zeile ausführt.

#### Wie funktioniert ein Interpreter?

Anders als ein Compiler übersetzt ein Interpreter den Quellcode nicht vollständig, sondern interpretiert und führt ihn Zeile für Zeile aus.

#### Vorteile eines Interpreters:

- Quellcode auf verschiedenen Plattformen und Systemen ist portabel, solange der Interpreter verfügbar ist.
- Ein Interpreter ermöglicht schnellere Entwicklungszyklen und Iterationen.

### Bytecode:

#### Was ist Bytecode?

Bytecode ist eine Zwischensprache, die von einigen Programmiersprachen und Plattformen verwendet wird. Es handelt sich um einen kompakten, plattformunabhängigen Satz von Instruktionen.

#### Wie wird Bytecode verwendet?

Bytecode wird oft als Zwischenschritt im Kompilierungsprozess verwendet. Der Quellcode wird zunächst in Bytecode übersetzt, der dann von einer virtuellen Maschine ausgeführt werden kann. Beispiele für Sprachen, die Bytecode verwenden, sind Java und C#.

#### Vorteile von Bytecode:

- Unabhängig, weil Bytecode von einer virtuellen Maschine ausgeführt wird, kann derselbe Bytecode auf verschiedenen Plattformen und Systemen ausgeführt werden.
- Kann in einer sicheren Umgebung ausgeführt werden, da die virtuelle Maschine den Zugriff auf Ressourcen kontrollieren kann und potenziell gefährlichen Code isoliert.

# Aktivitätsdiagramme

Den Diagrammtyp Aktivitätsdiagramm entwickeln und zeichnen:

