

# Darstellung von Algorithmen

## Python

- Korrekter, ausführbarer Python-Code
- Vorausgesetzt ist sprachspezifisches Wissen

### Anfang und Ende

- Gibt es in Python nicht

### Anweisungen

- Nach korrekter Python-Syntax
- Variabelzuweisungen: `summe = 5`
- Funktionsaufrufe `print(summe)`

```
summe = 5
print(summe)
```

### Verzweigungen

```
if summe == 13:
    print("Gewonnen!")
nur mehrstufig:
elif heute > 13:
    print("Verloren!")
zweiseitig und mehrstufig:
else:
    print("Weiterspielen möglich!")
```

### Schleifen

- Angenommen, die verwendeten Funktionen existieren:  
`wuerfeln()` gibt eine Augenzahl zurück  
`ueberpruefen()` überprüft, ob das Spiel gewonnen/verloren ist oder weitergehen kann.

```
summe = 0, anzahl_wuerfe = 3
```

#### kopfgesteuert

```
while summe <= 13:
    summe += wuerfeln()
```

#### zählergesteuert

```
for i in range(anzahl_wuerfe):
    summe += wuerfeln()
    ueberpruefen()
```

#### fussgesteuert

in Python nicht verfügbar

## Pseudocode

- Mischung aus natürlicher und formaler Sprache
- unabhängig von einer Programmiersprache
- abstrakt, strukturiert, ohne technische und sprachspezifische Details

### Anfang und Ende

- Nicht zwingend, teilweise in Anlehnung an Programmiersprachen mit **begin/end** oder **{}**

### Anweisungen

- Werden formal, in natürlicher Sprache oder gemischt beschrieben, unabhängig von der Programmiersprache

```
summe mit 5 initialisieren
summe ausgeben
```

### Verzweigungen

- In Worten beschrieben, Anfang und Ende gekennzeichnet oder Block eingerückt wie in Python

#### Einseitig- und zweiseitig

```
Falls summe = 13:
    „Gewonnen!“ ausgeben
```

#### Mehrstufig

```
Ansonsten, falls summe > 13:
    „Verloren!“ ausgeben
```

#### Zweiseitig und mehrstufig

```
Sonst:
    „Weiterspielen möglich“ ausgeben
```

### Schleifen

- In Worten beschrieben, Anfang und Ende gekennzeichnet oder Block eingerückt wie in Python

#### kopfgesteuert

```
Während summe < 13:
    würfeln und die Summe um die Augenzahl erhöhen
    summe überprüfen
```

#### zählergesteuert

```
dreimal wiederholen:
    würfeln und die Summe um die Augenzahl erhöhen
    summe überprüfen
```

#### fussgesteuert

```
Solange summe <= 13:
    würfeln und die Summe um die Augenzahl erhöhen
```

## Programmablaufplan

- Visualisierung des Programmablaufs durch grafische Elemente
- auch **Flussdiagramm** genannt

### Anfang und Ende

- Oval



### Anweisungen

- Kästchen
- Variabelzuweisungen: `mit =` oder `<—`



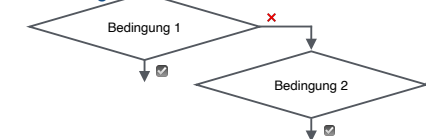
### Verzweigungen

- Rauten

#### Ein- und zweiseitig

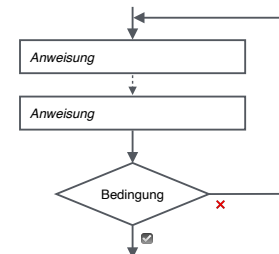


#### Mehrstufig



### Schleifen

- Mit einer Kombination von Bedingungsprüfungen (Verzweigungen) und Verbindungen, die weiter oben wieder ansetzen, können Schleifen generiert werden.



## Struktogramm

- Visualisierung der Programmstruktur durch grafische Elemente (Strukturblöcke)
- auch **Nassi-Shneiderman-Diagramm** genannt

### Anfang und Ende

- In der Regel wird das Struktogramm in einen Kasten geschrieben, als Titel fungiert der Name des Programms

### Anweisungen

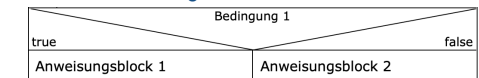
- Kästchen
- Variabelzuweisungen: `mit =` oder `<—`



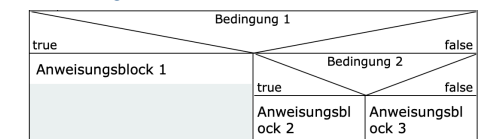
### Verzweigungen

- Dreieck, darunter geht es in den Spalten weiter

#### Ein- und zweiseitig

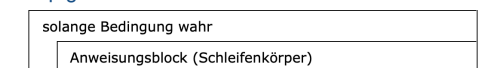


#### Mehrstufig

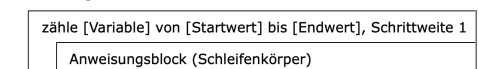


### Schleifen

#### kopfgesteuert



#### zählergesteuert



#### fussgesteuert

