

Programmieren BA



ANALYSE

SEQUENZ

Ablauf

DESIGN

```

    graph TD
      Start((Start)) --> A[Tue dies]
      A --> B[Tue das]
      B --> C[Tue jenes]
      C --> Ende((Ende))
  
```

SELEKTION

Entscheidung

ITERATION

ANALYSE

Schleife Wiederholung

KOPFGESTEUERT

```

    graph TD
      A[Tue dies] --> B{[solange BEDINGUNG?]}
      B -- [erfüllt] --> C[Tue wieder]
      C --> B
      B -- [nicht erfüllt] --> D{ }
      D --> E[Anzahl Iterationen: 0 ... ∞]
  
```

while ..

do .. while

SCHLEIFEN mit AB

```

    graph TD
      A[Tue dies] --> B{[solange BEDINGUNG?]}
      B -- [erfüllt] --> C[Tue wieder]
      C --> B
      B -- [nicht erfüllt] --> D{ }
      D --> E[Break]
      E --> F{ }
      F --> G{ }
  
```

ABSTRAKTION

Prozedur Funktion

ANALYSE

ABSTRAHRIERTER AUFRUF

KONKRETE AUSFÜHRUNG

DESIGN

(Aufruf hat evtl. Parameter)

(Funktion hat evtl. Rückgabewert)

```

    graph TD
      subgraph AD_Aufrufendes_Programm [AD: Aufrufendes Programm]
        Start1((Start)) --> Entscheide{Entscheide}
        Entscheide -- [ja] --> Machwas1[Aufruf MACHWAS]
        Entscheide -- [nein] --> Machwas2[Aufruf MACHWAS]
        Machwas1 --> Machwas2
        Machwas2 --> Ausgabe[Ausgabe]
        Ausgabe --> Machwas3[Aufruf MACHWAS]
        Machwas3 --> Machwas4[Aufruf MACHWAS]
        Machwas4 --> Ende1((Ende))
      end

      subgraph AD_Abstraktion_MACHWAS [AD: Abstraktion MACHWAS]
        Start2((Start)) --> MachwasIn[Machwas In]
        MachwasIn --> MachwasOut[Machwas Out]
        MachwasOut --> Ende2((Ende))
      end

      Machwas1 -.-> MachwasIn
      Machwas2 -.-> MachwasIn
      Machwas3 -.-> MachwasOut
      Machwas4 -.-> MachwasOut
  
```

