


Software-Paradigmen



Begriffsklärung: Was ist ein Programmierparadigma?

Überblick über heute gebräuchliche Programmierparadigmen.

Was ist bei der Auswahl einer Programmiersprache zu beachten?

Begriffsklärung

Ein **Programmierparadigma** ist ein grundsätzlicher Programmierstil.

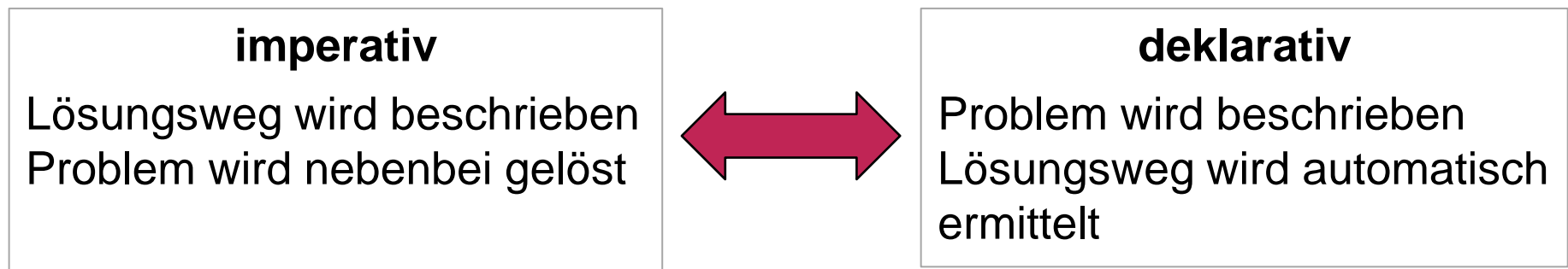
Die Unterschiede liegen im Umgang und der Repräsentation der Daten bzw. in der Darstellung des Kontrollflusses.

Die verschiedenen Paradigmen schließen sich nicht grundsätzlich gegenseitig aus. Sie können in den meisten Fällen untereinander kombiniert werden.

Überbegriffe

Überbegriffe

- Imperative Programmierung
- Deklarative Programmierung



imperativ (z.B. Pascal)

Code besteht aus einer Folge von Befehlen

Steuerung über Kontrollstrukturen

- Entscheidungen
- Schleifen

```
procedure quicksort(l,r : integer);  
var x,i,j,tmp : integer;  
begin  
  if r>l then  
    begin  
      x:=a[l]; i:=l; j:=r+1;  
      repeat  
        repeat i:=i+1 until a[i]>=x;  
        repeat j:=j-1 until a[j]<=x;  
        tmp:=a[j]; a[j]:=a[i]; a[i]:=tmp;  
      until j<=i;  
      a[i]:=a[j]; a[j]:=a[l]; a[l]:=tmp;  
      quicksort(l,j-1);  
      quicksort(j+1,r)  
    end  
  end;  
end;
```

deklarativ (Haskell)

Problembeschreibung steht im Vordergrund

- Lösungsweg ermittelt der Compiler/ Interpreter

```
quicksort [] = []  
quicksort (x:xs) = quicksort [n | n<-xs, n<x] ++ [x] ++ quicksort [n | n<-xs, n>=x]
```



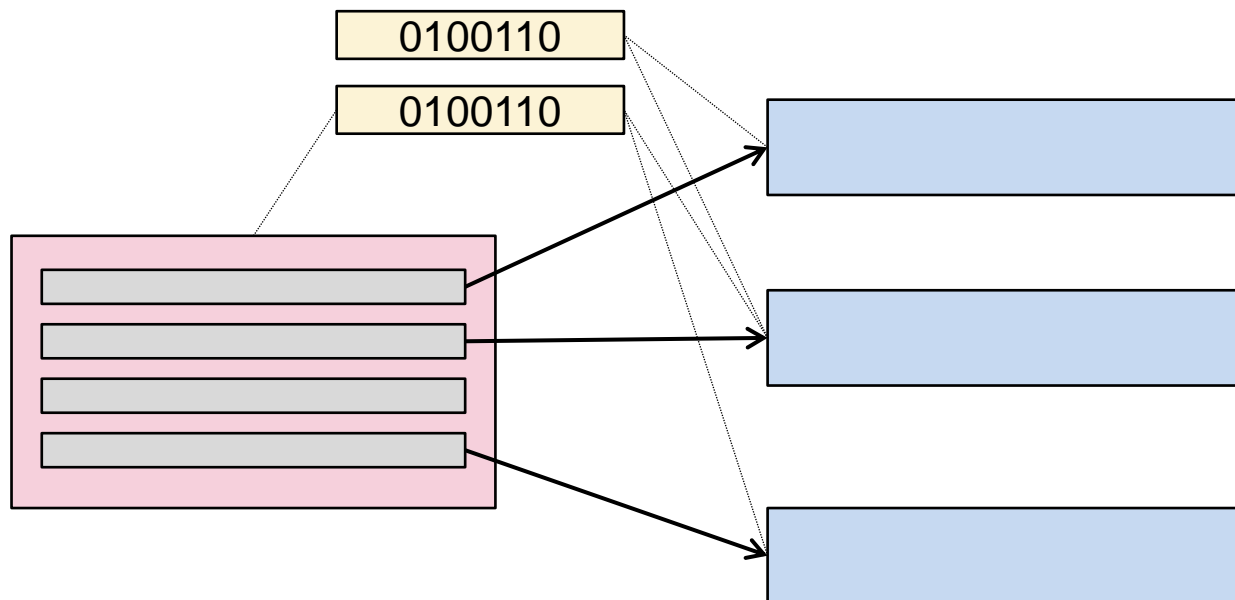
Programmierparadigmen



Prozedurale Programmierung

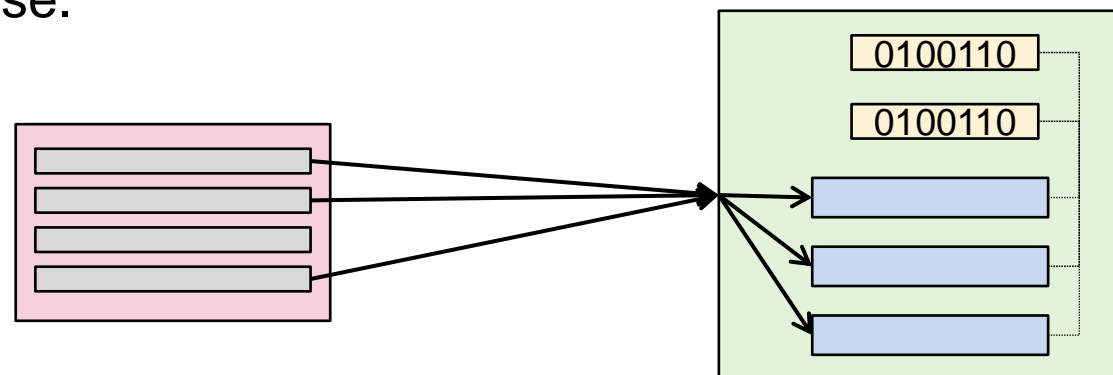
Zerlegung eines Algorithmus in überschaubare Teile

- Prozeduren
- Funktionen



Objektorientierte Programmierung

- Orientiert sich am menschlichen Alltag. Die Umwelt wird in Form von Objekten wahrgenommen.
- Abstraktion
 - Daten und Funktionen werden zusammengefasst (Klasse)
- Kapselung
 - Daten werden nach außen hin gekapselt, nicht zum Objekt gehörende Methoden dürfen nicht auf diese zugreifen.
- Vererbung
 - Abgeleitete Klassen erben Daten und Funktionen der Basisklasse.



Hauptunterschied objektorientiert und prozedural

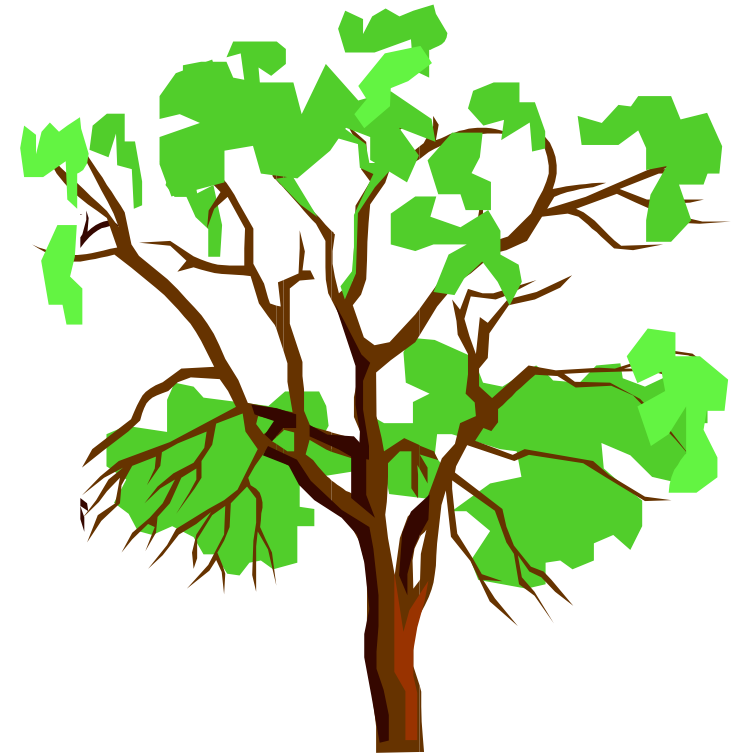
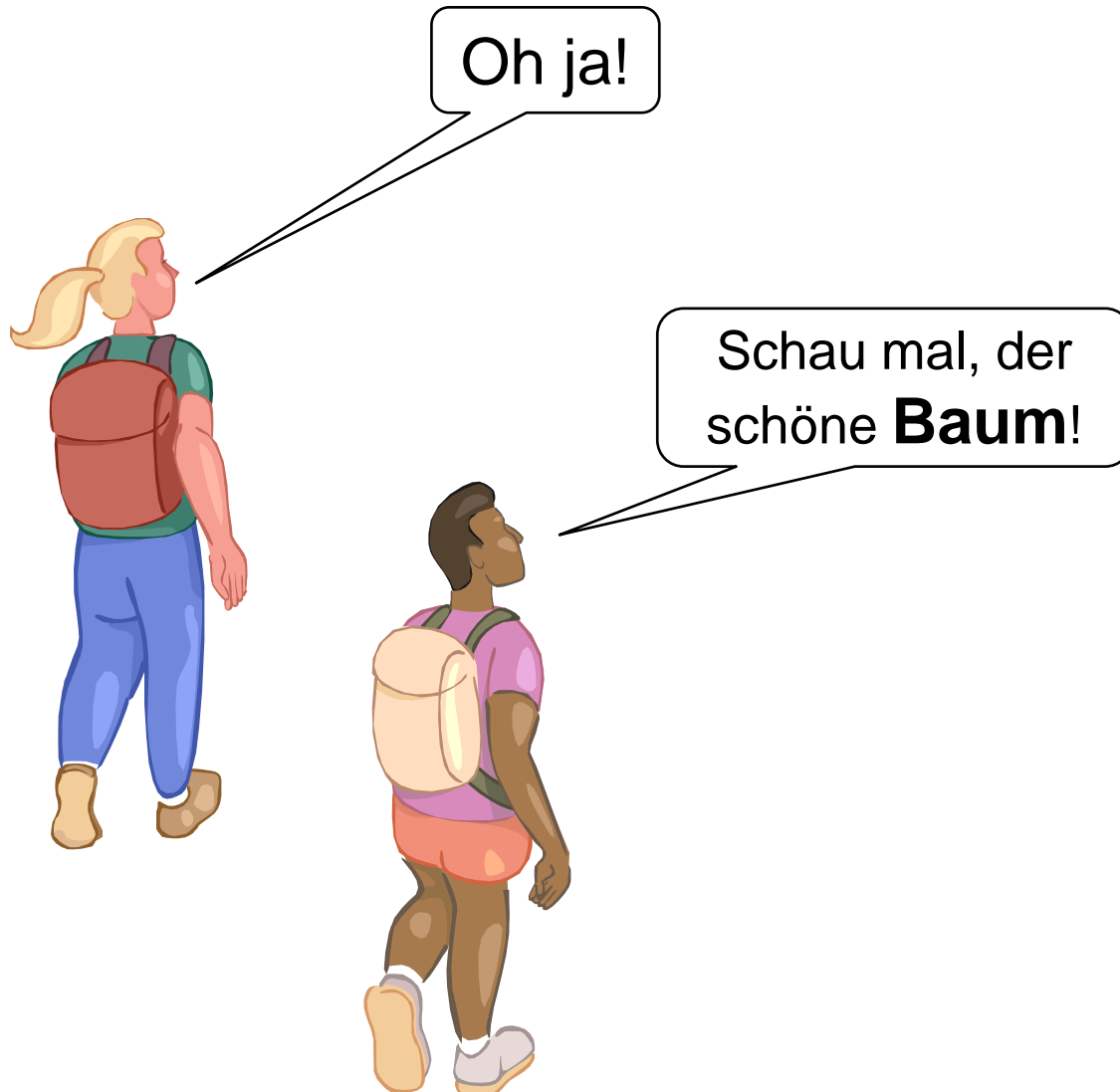
- **Objektorientiert:**
Zusammenfassung von Daten und Funktionen in Objekten.
Funktionen werden über das Objekt implizit auf die zugehörigen Daten angewandt.

- **Prozedural:**
Es besteht kein sichtbarer Zusammenhang zwischen Daten und Funktionen.

Prozedural

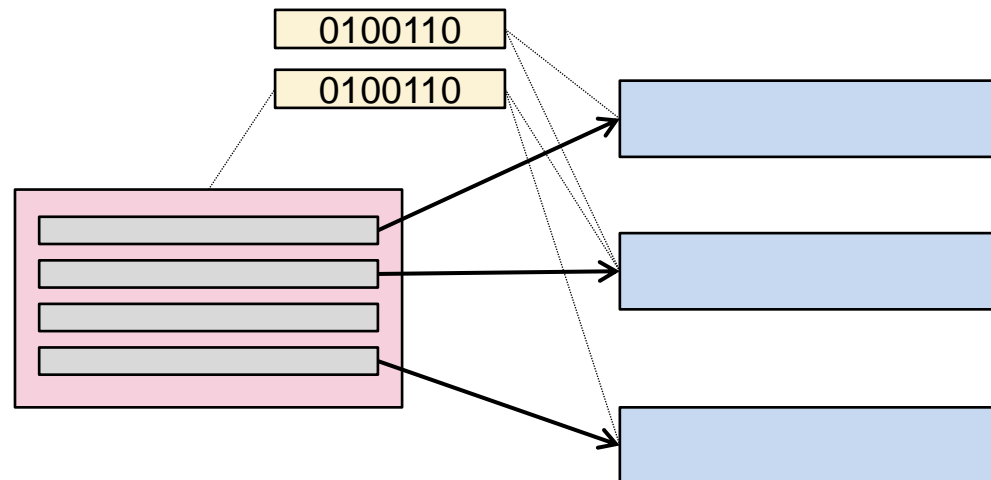


Objektorientiert



Strukturierte Programmierung

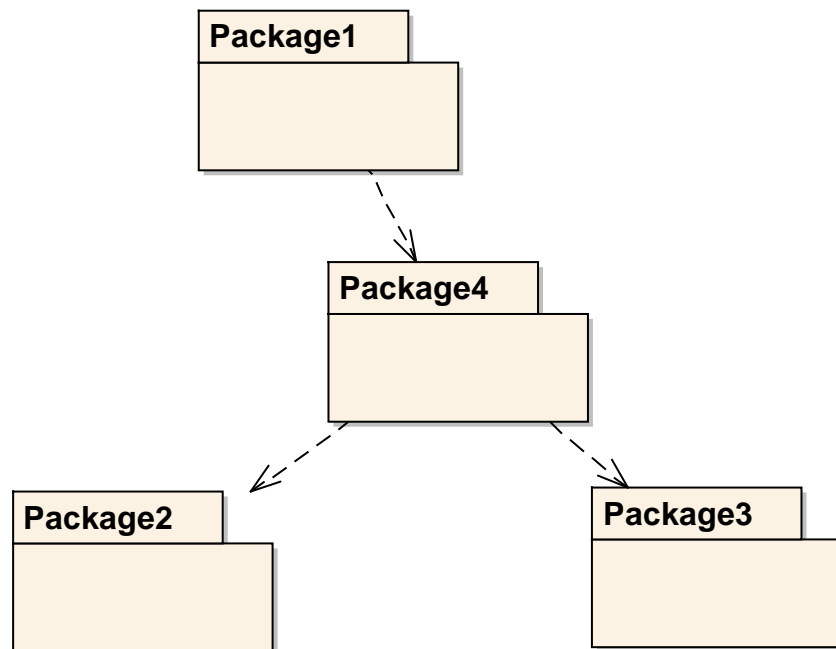
- baumartige Zerlegung in Teilprogramme (Prozeduren)
- Beschränkung auf drei Kontrollstrukturen
 - Sequenz (Anweisungen, die hintereinander ausgeführt werden)
 - Verzweigung (Entscheidung)
 - Wiederholung (Schleife)
- Vermeidung von **goto**
- ist heute selbstverständlich, neuere Programmierparadigmen bauen darauf auf



Modulare Programmierung

Aufteilung von Programmen in Teilblöcke (Module)

- Module können einzeln entworfen, entwickelt und getestet werden
- Wiederverwendung von Modulen
- Einfachere Fehlersuche



Aspektorientierte Programmierung

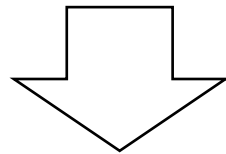
- Verwendung von generischen Funktionen über mehrere Klassen hinweg.
- Für Funktionen, die nicht über Vererbung verteilt werden können.
- Beispiele:
 - Protokollierung (Logging)
 - Fehlerbehandlung
 - Persistenz

Funktionale Programmierung

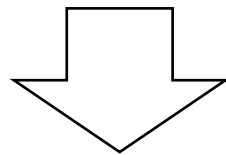
- Basiert auf dem mathematischen Funktionsbegriff.
- Hauptfunktion wird aus anderen Funktionen zusammengesetzt.
- Parameter und Rückgabewerte können Funktionen sein.
- Variablen haben immer denselben Wert
- Es gibt keine Zuweisungen und Schleifen
→ wird durch Rekursionen ersetzt.

Funktionale Programmierung

$$X = X + 1$$

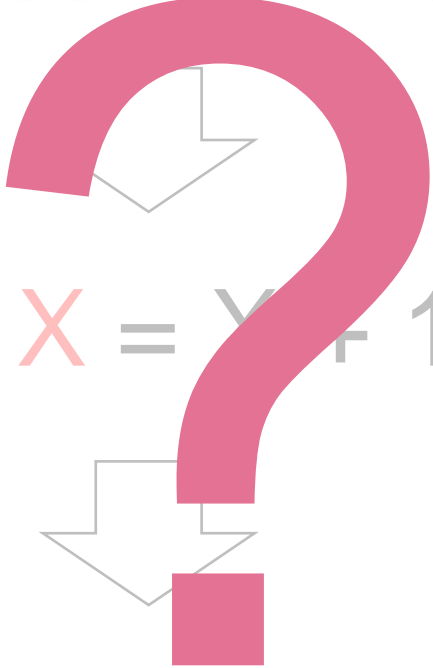


$$X - X = X + 1 - X$$



$$0 = 1$$

Funktionale Programmierung

$$X = X + 1$$


$$X - X = X + 1 - X$$

$$0 = 1$$

Auswahl einer Programmiersprache

Wie wird eine Programmiersprache ausgewählt?

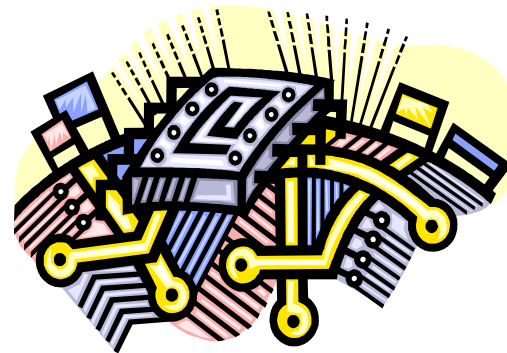
- Geschieht oft aus dem Bauch heraus
- Wird von alteingesessenen Entwicklern bestimmt
- Der Chef hat da mal etwas gehört
- Nur nichts Neues machen

Besser:

Nutzung von
objektiven Kriterien
zur Auswahl der
Programmiersprache!

Zielsystem(e)

- Embedded Target
- PC
- Nur ein Ziel-Betriebssystem
- Applikation soll auf mehreren Plattformen laufen
 - mit neu-Übersetzen
 - ohne neu-Übersetzen



Randbedingungen

- Echtzeitanforderungen
- Verfügbare Programmiersprachen
- Verfügbare Bibliotheken
 - frei
 - kommerziell
- Support
- Communities



Team

- Ausbildungsstand
- Vorlieben der Entwickler → größtmöglicher Konsens
- Motivation



Es gibt nicht DIE Programmiersprache!

- Programmierparadigmen wurden lösungsbezogen entwickelt und sind bei falschem Einsatz unproduktiv.
 - z.B. imperativ \leftrightarrow deklarativ
- Andere Paradigmen bauen aufeinander auf und ergänzen sich gegenseitig.
 - z.B. prozedurale und strukturierte Programmierung

Die objektorientierte Entwicklung ist heutzutage Stand der Technik und wird in vielen Projekten eingesetzt.

Oft wird die Auswahl einer Programmiersprache durch die Plattform stark eingeschränkt (z.B. embedded Programmierung)

Es ist sinnvoll, eine auf der Zielplattform etablierte Sprache zu verwenden, Exoten sind Risikobehaftet