

## II.2.1 Grundzüge der Objektorientierung

Mittwoch, 9. November 2016 08:30

Bislang: Datenstrukturen müssen mit Hilfe von Arrays dargestellt werden.

Vorteil: Arrays erlauben  
wahlfreien Zugriff (effizient)

Nachteile:

- Arrays fassen nur Werte des gleichen Typs zusammen.
- Mit der Erzeugung des Arrays liegt Elementzahl fest. Eignet sich nicht für dynamisch wachsende/schrumpfende Datenstrukturen.
- Array drückt nur eine Eigenschaft eines Objekts aus. Die verschiedenen Eigenschaften eines Objekts sind dann über mehrere Arrays verteilt.  
z.B. Rechteck  $O$  hat die Eigenschaften  $laenge[0]$ ,  $breite[0]$ ,  $strichst[0]$   
⇒ für Zuweisungen von Rechtecken braucht man 3 Befehle
- Arrays sind reine Daten-

strukturen, d.h., sie enthalten selbst keine Algorithmen zur Berechnung weiterer Objekt-Eigenschaften (wie z.B. fläche).

Besser: Objektorientierte Darstellung

Idee: • Objekt modelliert alle Eigenschaften eines Konzepts

• Klasse beschreibt, welche Eigenschaften ein Objekt hat (d.h. sie definiert einen Datentyp). Die Objekte sind die Elemente des Datentyps.

• Klasse definiert 2 Arten von Eigenschaften:

- Attribute ( $\hat{=}$  Variablen-deklarationen, z.B. *länge*, *breite*, *strickstärke*)

D.h. jedes Objekt  $r$  der Klasse Rechteck besitzt die Eigenschaften

$r.länge, r.breite, r.strickstärke$   
 $\uparrow \quad \uparrow \quad \uparrow$   
Typ double Typ int

- Methoden: berechnen Eigenschaften aus den Attributen des Objekts (z.B. flaeche())

Man kann daher  
r. flaeche() aufrufen.

• Klassen-Datentypen werden wie andere Typen benutzt. Erzeugung neuer (leerer) Objekte mit "new".

Bisher: Klasse verwendet als Teilprogramm (d.h. als Sammlung von Methoden).

Jetzt: Klasse als Datenstrukturen

Klasse legt fest, welchen Speicherplatz man für jedes Objekt braucht. z.B.: Rechteck-Objekte benötigen Platz für 2 double- und 1 int-Wert.

• Variablen von Klassentypen enthalten als Wert wieder

nur eine Referenz auf das Objekt (d.h. die Adresse des Speicherplatzes, an dem das Objekt steht).

- Ähnlich wie bei Arrays.

Zugriff auf die Objekt-Eigenschaften nicht mit `[0]`, ...

sondern mit `.laenge`, ...

- Wie bei Arrays:

- Seiteneffekte (Aliasing) möglich

(Änderung an `r` bewirkt auch Änderung an `s`).

- Garbage Collector löscht Objekte im Heap, die nicht mehr erreichbar sind.