

Begriffe am Beispiel

- Objekte
- Klassen



Definition: Objekt

Ein **Objekt**

- ist *eindeutig* identifizierbar,
- gehört zu einer *Klasse*,
- implementiert [...] *Methoden* (→ sein *Verhalten*)
und
- hat einen änderbaren *Zustand* in Form von Werten seiner *Feldern* (→ seine *Struktur*).

Die Begriffe am Beispiel

verschiebeDich (x, y)

Eindeutigkeit:

genau
dieser
Kreis

Kreis
Mittelpunkt $x=4, y=6$
Radius $r=3$

Klassenzugehörigkeit:

alle Kreise "wissen",
dass sie Kreise sind

nicht
dieser!

Struktur:

diese
drei
Felder

berechneDeineFläche ()

Verhalten:

diese beiden
Methoden

Zustand:

Das Kreisobjekt befindet sich
am Punkt 4, 6 und hat einen
Radius von 3

WIRTSCHAFT

Kreis
Mittelpunkt $x=4, y=6$
Radius $r=3$



Der Programmcode dazu

```
class Kreis {  
  
    private int x, y, r;  
  
    Kreis ( int xKoordinate, int yKoordinate, int radius ) {  
        x = xKoordinate;  
        y = yKoordinate;  
        r = radius;  
    }  
  
    double berechneDeineFläche () {  
        return r * r * 3.14;  
    }  
  
    void verschiebeDich ( int dX, int dY ) {  
        x += dX;  
        y += dY;  
    }  
}
```

Definition: Klasse

Eine **Klasse** definiert die (Daten-) *Struktur*
und das *Verhalten* aller

aus ihr instanziierten /

von ihr abgeleiteten /

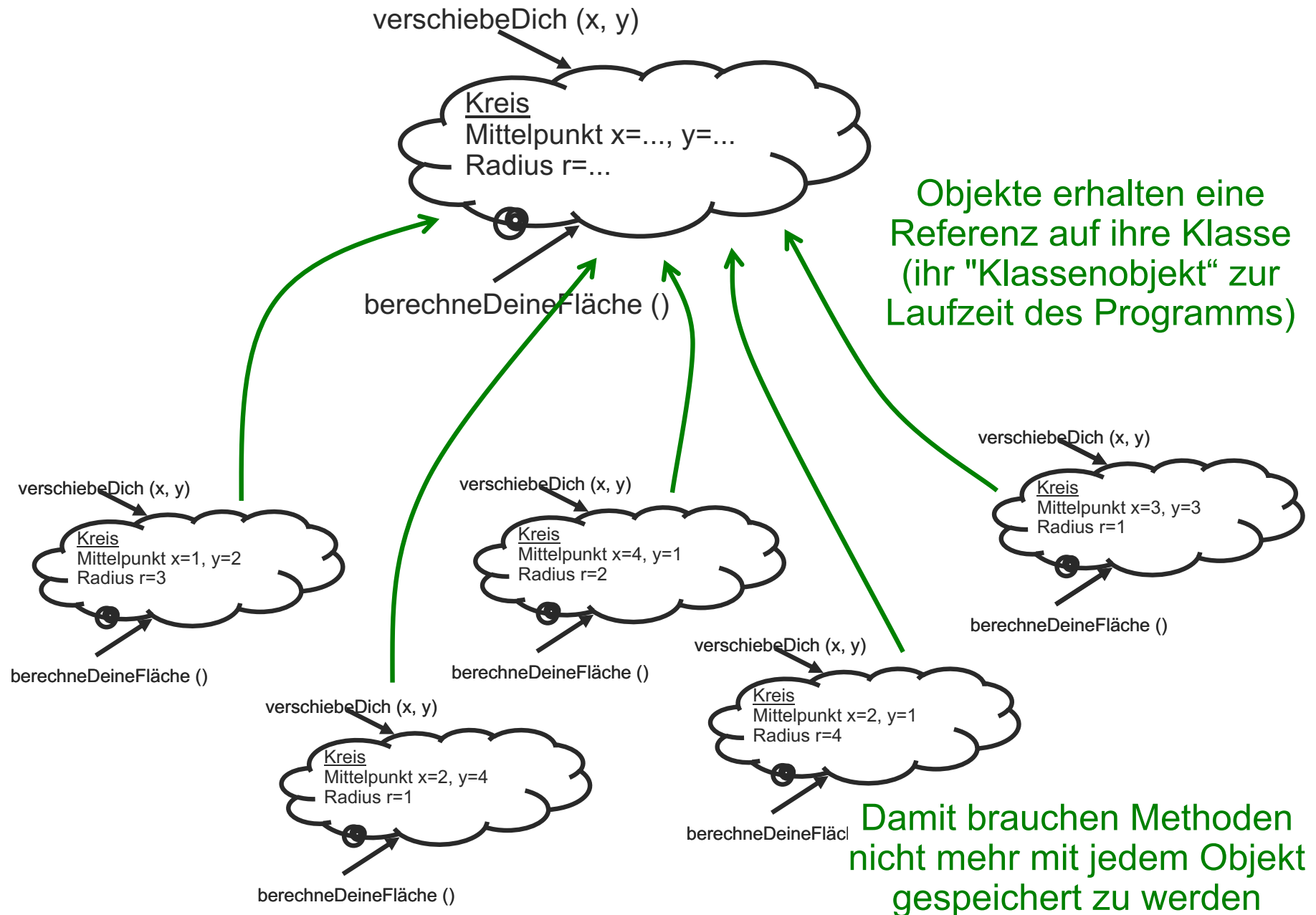
nach ihrem Bauplan erstellten

Objekte

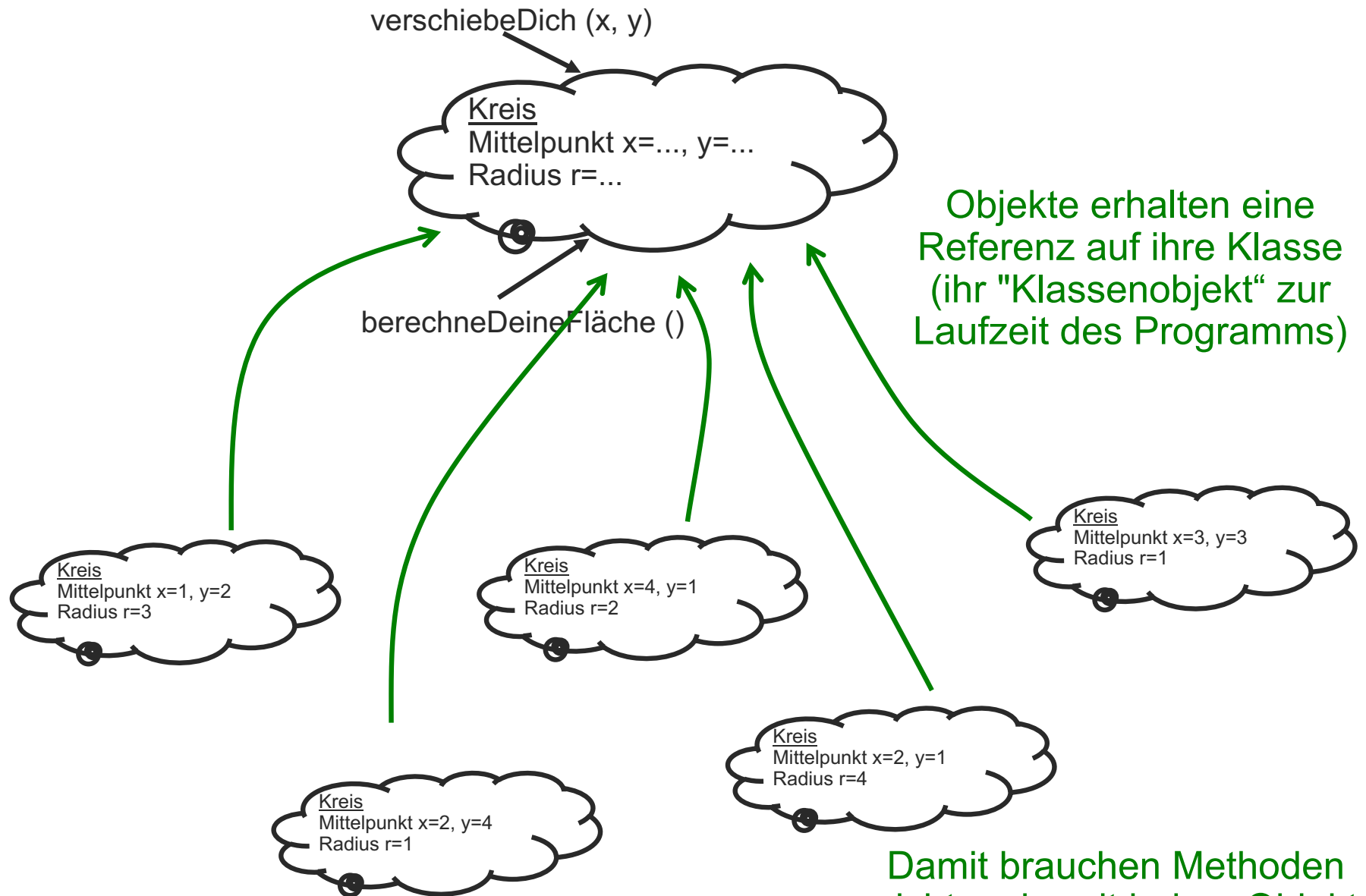
Verwendung von Klassen

- Zur Erinnerung aus der Definition von *Objekt*:
Ein Objekt
 - ... gehört zu einer *Klasse*, ...
- Beim Entwickeln eines Programms:
Objekte mit der gleichen Struktur (= den gleichen Feldern)
und dem gleichen Verhalten (= implementierte Methoden)
werden durch eine Klasse beschrieben
- Zur Laufzeit eines Programms:
Gleichartige Objekte werden erzeugt, indem sie aus der
gleichen Klasse instanziiert (auch: abgeleitet) werden
- *Eine mögliche* Implementierung durch die VM:
 - Felder werden mit jedem Objekt gespeichert,
 - Methoden werden zentral in einem "Klassenobjekt" gespeichert
 - Objekte halten eine Referenz auf ihr "Klassenobjekt"

Beispiel für Klassen und Objekte



Beispiel für Klassen und Objekte



Objekte erhalten eine Referenz auf ihre Klasse (ihr "Klassenobjekt" zur Laufzeit des Programms)

Damit brauchen Methoden nicht mehr mit jedem Objekt gespeichert zu werden

Klassen und Objekte 1/2

- Klassen-Deklarationen sind **benutzerdefinierte** (Daten-) **Typ-Vereinbarungen**
- **Benutzerdefinierte** Datentypen sind zu verstehen als Ergänzung zu den **vordefinierten** Datentypen wie z.B. int, boolean, String
- Benutzerdefinierte Datentypen können, sobald sie einmal deklariert sind, wie vordefinierte Datentypen verwendet werden, um zum Beispiel Variablen dieses Typs zu deklarieren

```
String hausnummer;  
CD meineCd;  
Datum meinGeburtsdatum;
```

- Die Werte, die eine Klasse annehmen kann, bezeichnet man als **Objekte** oder **Instanzen**: Wir sprechen von einem Buch-Objekt, einem Auto-Objekt, einem Adress-Objekt etc.
- Daher spricht man von der **objektorientierten Programmierung (OOP)** und von **objektorientierten Programmiersprachen** (wie zum Beispiel Java, C++, C#)



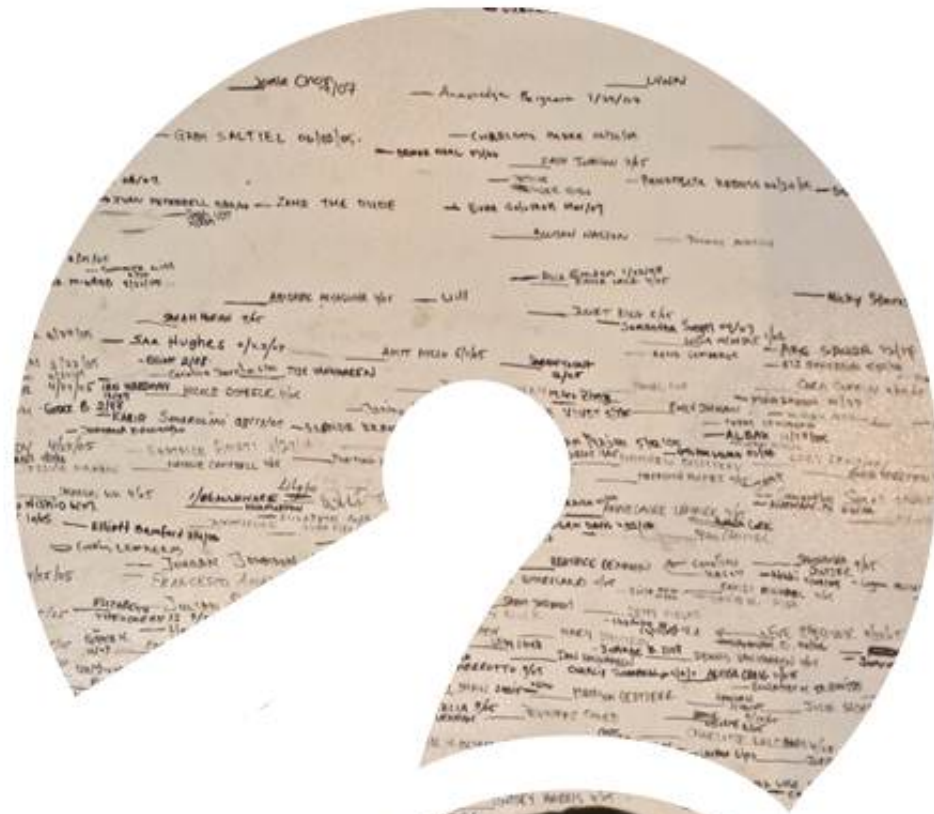
Klassen und Objekte 2/2

- Objekte, also Werte von Klassen-Typen, werden, genau wie Arrays, durch einen **Konstruktor**-Aufruf mit dem Schlüsselwort **new** erzeugt (oder kreiert, abgeleitet, instanziiert)

```
meineCd = new CD();
```

```
meinGeburtsdatum = new Datum(18, 4, 1969);
```

Die runden Klammern sind Parameter-Klammern
(wie bei Methodenaufrufen)



F R A G E N



photography: woodleywonderworks
<http://www.flickr.com/photos/wwworks/2350106729>
art work: Peter Kaiser