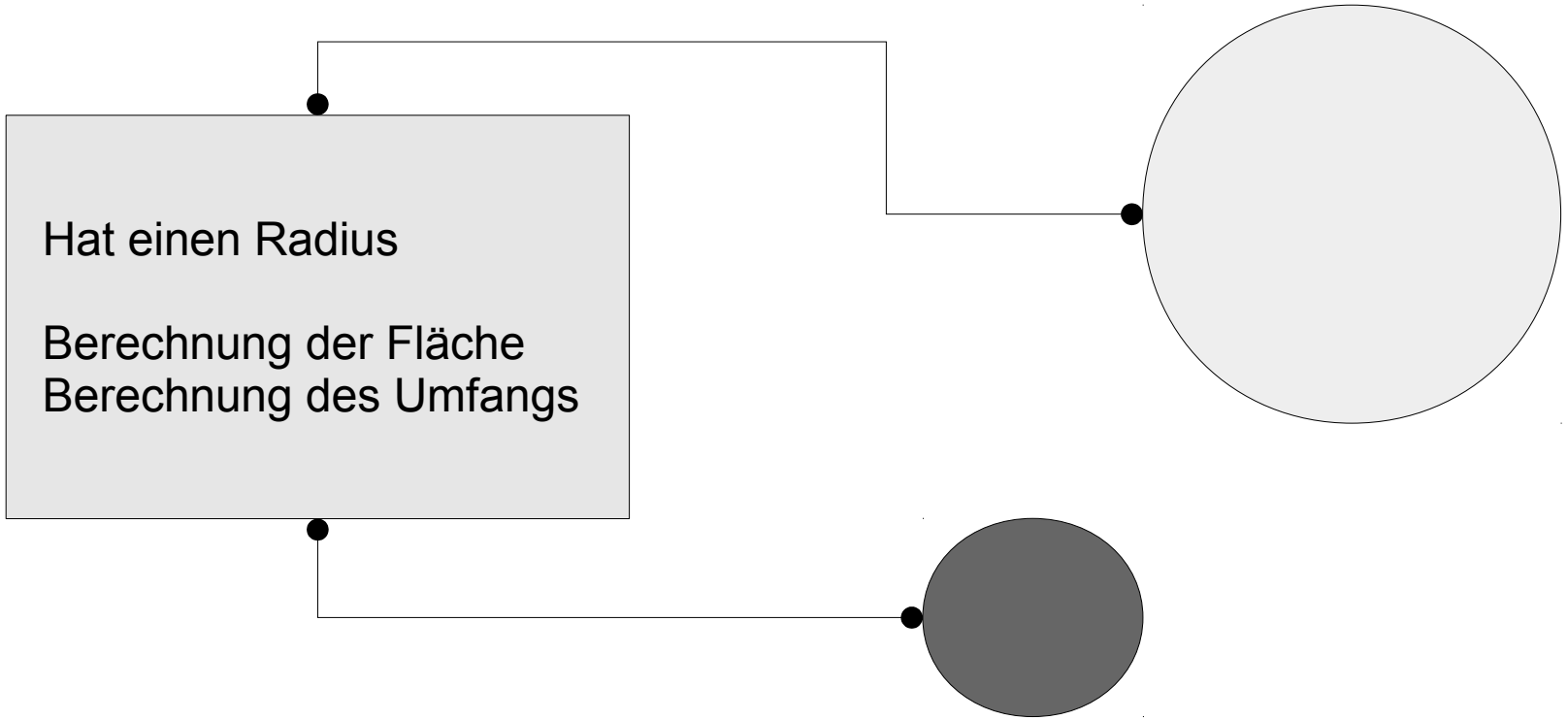
Java - Klassen, Objekte und Arrays



Klasse

- Abstraktion von Dingen aus der realen Welt.
- Definition eines konkreten Objekts. Welche Daten werden zur Beschreibung des Objekts benötigt? Wie kann das Verhalten des Objekts allgemeingültig beschrieben werden?
- Formale Beschreibung einer bestimmten Objektart.
- Vorlage für die Erzeugung eines Objektes.

Objekt

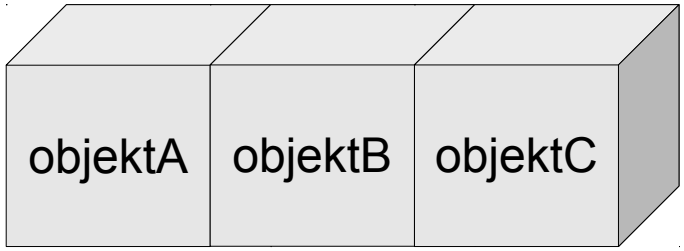
- Ein Ding (Exemplar, Instanz) aus der realen Welt.
- Substantive in einer Textbeschreibung.
- Erzeugung zur Laufzeit eines Programms über Klassen.
- Instanz einer Klasse.
- Alle Elemente einer Kategorie von Dingen haben die gleichen Eigenschaften, aber in unterschiedlichen Ausprägungen. Sie nutzen die gleichen Methoden zum Ändern ihrer Ausprägungen.

Array (Felder, Vektoren)

- Zusammenfassung von vielen Werten gleichen Datentyps.
- Gruppieren von Variablen zu einem Thema (zum Beispiel Temperaturwerte eines Monats).
- Speicherung einer festen Anzahl von Werten gleichen Datentyps.
- Ein- oder mehrdimensional.

Eindimensionale Arrays

- Folge von Werten gleichen Datentyps.
- In einem Container befinden sich mehrere kleinere Boxen gleicher Größe, aber unterschiedlichen Inhalts. Die Größe und die Art des Inhalts ist abhängig von der gewählten Klasse.
- Sammlung von Objekten der gleichen Klasse.



… deklarieren

```
int[] monat;

clsKreis[] vieleKreise;
```

- Die eckigen Klammern kennzeichnen einen Container, der viele kleine Boxen enthält.
- Durch die Angabe des Datentyps oder der Klasse wird die Größe der einzelnen Boxen im Container festgelegt.
- Der Speicherplatz für den Container ist noch nicht reserviert. Die Gesamtzahl der Boxen im Container ist noch nicht bekannt.
- Der Container hat einen eindeutigen Namen.

Deklaration von eindimensionalen Arrays

`int`	`[`	`]`		`monat`	`;`
`Datentyp`	`[`	`]`		`feldname`	`;`
`clsKreis`	`[`	`]`		`vieleKreise`	`;`
`Klasse`	`[`	`]`		`feldname`	`;`

- Die Variable verweist auf ein Feld von x Werten vom Datentyp oder der Klasse ...
- Der Feldname ist ein Verweis auf das erste Element in dem Feld.

Feldnamen

- Frei wählbares Label für den Speicherort der ersten Box in einem Container.
- Jeder Name kommt nur einmal in einem Block vor. Der Block von Anweisungen beginnt und endet mit den geschweiften Klammern.
- Häufig wird die Pluralform von Dingen für benutzerdefinierte Namen genutzt. Zum Beispiel der Name `kreis` speichert ein Objekt von der Klasse `clsKreis`. Der Name `kreise` speichert x Objekte von der Klasse `clsKreis`.

… initialisieren

```
monat = new int[11];
vieleKreise = new clsKreis[4]
```

- Die Anweisung new erzeugt ein Feld von einer bestimmten Größe von einem Datentyp oder Klasse.
- Die Gesamtgröße des Containers und die Anzahl der Boxen im Container werden festgelegt.
- Die Anzahl der Objekte von einer Klasse wird festgelegt.

Initialisierung von eindimensionalen Arrays

monat	=	new	int	[	11	]	;
feldname	=	new	Datentyp	[	anzahl	]	;
vieleKreise	=	new	clsKreis	[	4	]	;
feldname	=	new	Klasse	[	anzahl	]	;

Erläuterung

- Mit Hilfe von `new` wird Speicherplatz für x Felder reserviert.
- Die Anzahl der Felder wird in den eckigen Klammern angegeben. Die Anzahl kann nachträglich nicht verändert werden.
- Jedes Feld ist so groß, dass es einen Wert entsprechend des Datentyps oder Klasse speichern kann.

… deklarieren und initialisieren

```
clsKreis[] vieleKreise = new clsKreis[4];
```

- Die Variable `vieleKreise` wird als Feld vom Typ `clsKreis` deklariert.
- Die Anweisung `new` erzeugt ein Feld von einer bestimmten Größe von der Klasse `clsKreis`, welches x Objekte von der Klasse speichern kann.
- Die Adresse des ersten Feldes im Speicher wird der Variablen `vieleKreise` übergeben.

Speicherung von Objekten in einem Array

```
vieleKreise[0] =
   new clsKreis(4, "schwarz", new int[]{0,255,255});

index = 1;
vieleKreise[index] = new clsKreis(3, "blau");

clsKreis kreisRot;
kreisRot = new clsKreis(5, "grau");
vieleKreise[2] = kreisRot;
```

Index eines Feldes

`vieleKreise`	`[`	`0`	`]`
`feldname`	`[`	`index`	`]`

- Dem Namen des Feldes folgen eckige Klammern.
- In den eckigen Klammern wird eine Ganzzahl als Index angegeben.
- Der Index identifiziert eindeutig ein Element in einem Feld.
- Literale oder Variablen können als Index genutzt werden.

Erzeugung eines Objekts

```
new clsKreis(4, "schwarz", new int[]{0,255,255});

new clsKreis();
```

- Mit Hilfe des Schlüsselwortes `new` wird ein Objekt von einer Klasse erzeugt.
- Dem Schlüsselwort folgt der Name der Klasse, die als Basis für die Instanz dient.
- Dem Namen folgen die leeren, runden Klammern. Der parameterlose Konstruktor wird aufgerufen.
- In den runden Klammern werden Parameter angegeben. Attribute werden mit den Parametern im Konstruktor initialisiert.

Zuweisung zu einem Element in einem Array

```
vieleKreise[0] =
   new clsKreis(4, "schwarz", new int[]{0,255,255});

vieleKreise[index] = new clsKreis(3, "blau");
```

- Rechts vom Gleichheitszeichen wird ein Objekt mit Hilfe von new erzeugt.
- Dem Element des Arrays wird ein Verweis auf das neu erzeugte Objekt übergeben.

Weitere Möglichkeit

```
clsKreis kreisRot;
kreisRot = new clsKreis(5, "grau");

vieleKreise[2] = kreisRot;
```

- Rechts vom Gleichheitszeichen befindet sich eine Objekt-Variable.
- Dem Element des Arrays wird der, in der Objekt-Variablen gespeicherte Verweis übergeben.
- In diesem Beispiel weisen die Objekt-Variable `kreisRot` und das Element `vieleKreise[2]` auf das gleiche Objekt.

Attribute (Instanzvariablen) einer Klasse

```
public class clsKreis {
    private final double pi = 3.14159;
    private final String linienfarbe;
    private double radius;
    private int[] innenfarbe;
```

- Felder, die als Attribute genutzt werden, werden wie Feld-Variablen in den Methoden deklariert.
- Attribute sind immer privat.

Initialisierung in einem Konstruktor

```
public clsKreis()
{
    radius = 0;
    linienfarbe = "schwarz";
    innenfarbe = new int[]{0,0,0};
```

- Felder, die als Attribute genutzt werden, werden wie Felder in Methoden im Konstruktor initialisiert.

Weitere Möglichkeit

```
public clsKreis(double dblRadius, String farbe, int[] innen){
      int index = 0;

      innenfarbe = new int[]{0,0,0};

      for(int element : innen){
        innenfarbe[index] = element;
        index++;
      }
```

Erläuterung

- Dem Konstruktor wird ein Array als Parameter übergeben.
- Das Array `int[] innen` wird dem Konstruktor übergeben.
- In dem Beispiel werden die Elemente des Attributs `innenfarbe` mit Hilfe der foreach-Schleife gesetzt.
- Das Attribut ist als Array `innenfarbe` in der Klasse deklariert.

Methoden in Klassen

- Abfolge von Anweisungen, die der Computer versteht.
- Zusammenfassung von Aktionen.
- Schnittstellen zum Benutzer.
- Lesen und Schreiben von Werten der gekapselten Instanzvariablen.

Parameter in Methoden

```
private int[] innenfarbe;

public void setzeFarbeRGB(int[] farbe)
{
   this.innenfarbe = farbe;
}
```

- Dem Attribut wird ein Verweis auf ein Array übergeben.
- Die lokale Variable `farbe` und das Attribut `innenfarbe` verweisen auf das gleiche Feld.

Rückgabewerte als Array

```
private int[] innenfarbe;

public int[] leseFarbeRGB(){
  return innenfarbe;
}
```

- Die Methode hat als Datentyp ein Array.
- In diesem Beispiel ist die Methode ein Feld vom Typ `int`.
- Der Rückgabewert muss ein Wert entsprechend des Typs sein.

Aufruf der Methode

```
int[] rgb = new int[3];

rgb = kreisRot.leseFarbeRGB();
```

- Die Methode wird mit Hilfe des Namens aufgerufen.
- Die Parameterliste entspricht der Parameterliste des Kopfes der Methode.
- Mit Hilfe des Gleichheitszeichens wird der Rückgabewert der Methode einer Variablen vom gleichen Typ wie die Funktion zugewiesen.