

Vergleich Java - Python

Gegenüberstellung am Beispiel Raumplaner Anfangsprojekt
Klasse Stuhl

Importe

```
import java.awt.Shape;  
import java.awt.geom.GeneralPath;  
import java.awt.geom.Rectangle2D;  
import java.awt.geom.AffineTransform;
```

Jede Anweisung
mit Semikolon
abschließen

```
from grafikfenster import *  
from math import radians
```

Zeilenende
beendet die
Anweisung

Vergleich Java - Python

Gegenüberstellung am Beispiel Raumplaner Anfangsprojekt
Klasse Stuhl

Kopf der Klassendefinition

```
/**  
 * Ein Stuhl, der ...  
 *  
 * @author Claus Albowski  
 * @version ...  
 */  
public class Stuhl  
{  
    private int xPosition;  
    private int yPosition;  
    private int orientierung;  
    private String farbe;  
    private boolean istSichtbar;  
    private int breite;  
    private int tiefe;
```

Deklaration mit
Sichtbarkeit und
Typ

```
class Stuhl():  
    """Ein Stuhl, der ..."""
```

Keine Zugriffsrechte

Keine Deklaration

Vergleich Java - Python

Gegenüberstellung am Beispiel Raumplaner Anfangsprojekt
Klasse Stuhl

Konstruktor

```
/**  
 * Erzeuge ... (Standardkonstruktor)  
 */  
public Stuhl() {  
    xPosition = 160;  
    yPosition = 80;  
    farbe = "blau";  
    orientierung = 0;  
    istSichtbar = false;  
    breite = 40;  
    tiefe = 40;  
}
```

Mehrere Konstruktoren
möglich [overload]

Geschweifte Klammern
definieren einen Block

```
def __init__(self,  
             xPos=20,  
             yPos=20,  
             breite=40,  
             tiefe=40,  
             winkel=0,  
             farbe="black",  
             sichtbar=False):  
    self.x=xPos  
    self.y=yPos  
    self.b=breite  
    self.t=tiefe  
    self.w=winkel  
    self.f=farbe  
    self.s=sichtbar  
    if sichtbar: self.Zeige()
```

Genau ein Konstruktor,
vordefinierte Parameter
möglich

Attribute kennzeichnen
durch self.

Einrückungstiefe
definiert einen Block

Vergleich Java - Python

Gegenüberstellung am Beispiel Raumplaner Anfangsprojekt
Klasse Stuhl

Methoden

```
/**  
 * Bewege dieses Objekt ...  
 */  
public void bewegeHorizontal(int entfernung) {  
    loesche();  
    xPosition += entfernung;  
    zeichne();  
}
```

```
def BewegeHorizontal(self, weite):  
    """Bewege dieses Objekt ... """  
    self.Verberge()  
    self.x += weite  
    self.Zeige()
```

[Instanz-]Methoden
erster Parameter self

Doppelpunkt
leitet Block ein

self. ist zwingend!

Vergleich Java - Python

Gegenüberstellung am Beispiel Raumplaner Anfangsprojekt
Klasse Stuhl

Zugriff auf die Klasse zum Zeichnen

```
/**
 * Zeichne dieses Objekt ...
 */
private void zeichne() {
    if (istSichtbar) {
        Shape figur = gibAktuelleFigur();
        Leinwand leinwand = Leinwand.gibLeinwand();
        leinwand.zeichne (
            this,      // leinwand kennt das Objekt
            farbe,    // definiert seine Zeichenfarbe
            figur);   // definiert seinen grafischen Aspekt
        leinwand.warte(10);
    }
}
```

```
def Zeichne(self):
    """Übergibt der Zeichenfläche das Objekt"""
    Zeichenflaeche.GibZeichenflaeche().Zeichne(self)
```

ist ersetzt in der aktuellen Version durch Zeige und Verberge:

```
def Zeige(self):
    """Übergibt der Zeichenfläche das Objekt"""
    Zeichenflaeche.GibZeichenflaeche().Zeichne(self)
```

Vergleich Java - Python

Gegenüberstellung am Beispiel Raumplaner Anfangsprojekt
Klasse Stuhl

In der Klasse zum Zeichnen

in Leinwand

```
public void zeichne(Object figur, String farbe, Shape shape) {  
    figuren.remove(figur); // entfernen, falls schon eingetragen  
    figuren.add(figur);    // am Ende hinzufügen  
    figurZuShape.put(figur, new ShapeMitFarbe(shape, farbe));  
    erneutZeichnen();  
}
```

Leinwand-Objekt
kennt zwar die Objekte,
greift aber nicht darauf zu

in erneutZeichnen()

```
{  
    loeschen();  
    for (Object figur:figuren)  
        (figurZuShape.get(figur)).draw(graphic);  
    zeichenflaeche.repaint();  
}
```

in Zeichenflaeche

```
def Zeichne(self, objekt):  
    if objekt in self.objekte:  
        self.objekte.remove(objekt)  
    self.objekte.append(objekt)  
    self.Refresh()
```

Zeichenflaeche-Objekt
greift direkt
auf Objekte zu!

in Draw(...) von Zeichenflaeche

```
for objekt in self.objekte:  
    if objekt.GibSichtbar():  
        path = gc.CreatePath()  
        gc.PushState()  
        path.AddPath(objekt.GibFigur())  
        gc.SetPen(wx.Pen(objekt.GibFarbe()))  
        gc.DrawPath(path)  
        gc.PopState()
```