



## **1. Beschreibe das Wesen Objektorientierter Modellierung in JAVA**

Modellieren bedeutet, die Wirklichkeit vereinfacht darzustellen. In der Wirklichkeit werden Objekte erkannt, dies können reale Gegenstände oder auch nur Kategorien sein. Von jedem Objekt lassen sich Attribute und deren Attributwerte benennen.

Objekte mit gleichen Attributen lassen sich zu Klassen zusammenfassen.

JAVA besteht nur aus Klassen.

## **2. Was versteht man unter einer Instanz?**

Klassen können als Bauanleitungen verstanden werden. Von jeder Klasse wird ein Objekt als Instanz dieser Klasse erzeugt, indem der new-Befehl ausgeführt wird.

z. B. `Kreis mykreis = new Kreis(10,10,10);`

## **3. Was versteht man unter Methoden?**

Methoden bilden eine Möglichkeit auf Attributwerte eines Objektes zuzugreifen. Entweder in Form einer „sondierenden Methode“, die lediglich den Attributwert liefert oder in Form einer „verändernden Methode“, mit der man Attributwerte verändern kann. Eine Methode wird mit ihrem Namen und den abschließenden runden Klammern „()“ aufgerufen, in der „Formale Parameter“ enthalten sein können.

Methoden können ein Ergebnis haben, dies wird im Quelltext mit der Anweisung return ausgegeben, durch den die Ausführung der Methode beendet wird. Das Schlüsselwort „void“ bedeutet, dass die Methode kein Ergebnis hat.

## **4. Was versteht man unter einer Signatur?**

Den Kopf einer Methode, dieser genügt um die Methode aufrufen zu können. Die Bedeutung ist aus der Signatur meist nicht eindeutig zu schließen und Bedarf daher einer Erklärung.

## **5. Was versteht man in Java unter einem Datentyp?**

Die Art eines Parameters, man unterscheidet primitive Datentypen (int, boolean, char, double, long) von Objekt-Datentypen, d.h. dass als Parameter eine Referenz auf ein Objekt erwartet wird.

**6. Benenne die einzelnen Teile der folgenden Klasse!**

```
class wuerfel{  
  
    int wert;  
  
    public wuerfel(){  
        wert=0;  
    }  
  
}
```

Die Klasse besteht aus einem Namen und einer äußeren Klammer, im Innenteil befinden sich Datenfelder (Attribute) und ein Konstruktor. Der Konstruktor hat den gleichen Namen wie die Klasse und als einzige Methode keine Angabe über einen Rückgabewert. Im Konstruktor werden für die Datenfelder die Startwerte festgelegt.

**7. Beschreibe, ob eine Klasse mehrere Konstruktoren haben kann!**

Ja, mehrere Konstruktoren sind möglich. Man bezeichnet dies als Überladen, die Konstruktoren müssen sich aber unterscheiden. Dies ist nur über unterschiedliche Parameter möglich.

**8. Wie lange sind Variablen gültig?**

Datenfelder (Attributwerte) existieren solange es das Objekt gibt. Formale Parameter und lokale Variablen sind nur gültig, solange die Methode abgearbeitet wird.

**9. Schreibe zu oben angegebener Klasse „wuerfel“ eine verändernde Methode, die den Wert der Variable wert auf einen beliebigen Wert zwischen 1 und 6 verändert!**

```
public void aendern(int wert){  
    if ((wert<=7) && (wert>=1)) this.wert=wert;  
    else System.out.println(„ungueltiger Wert“);  
}
```

**10. Schreibe zur Klasse „wuerfel“ eine Sondierende Methode zur Ausgabe des Wertes!**

```
public int getWert(){  
    return wert;  
}
```

**11. Beschreibe was folgende Anweisungen bedeuten!**

- a) `$wert=$wert++;`
- b) `$wert+=10;`
- c) `$wert*=10;`

- a) Der Wert der Variable wird um 1 erhöht.
- b) Der Wert der Variable wird um 10 erhöht.
- c) Der Wert der Variable wird mit 10 multipliziert.

**12. Beschreibe, welche Ergebnisse sich bei der Auswertung einer if-Anweisung ergeben können!**

In der Klammer muss sich ein boolescher Datentyp ergeben, also true oder false.

**13. Welche der folgenden Zeilen kann nicht sinnvoll sein? Gib eine Begründung an!**

```
if ((wert>5) || (wert<10)) Anweisung1...  
if ((wert>5) && (wert<10)) Anweisung1...
```

**13. Was versteht man unter einer Konkatenation?**

Konkatenation ist die Verkettung von Zeichenketten (Strings). Dies erfolgt in Java mit dem Operator „+“

**14. Begründe, ob folgender Programmcode funktioniert?**

```
public int getUhrzeit(int stunde, int minute){  
  
    return stunde+":"+minute;  
  
}
```

Nein, der Ergebnistyp muss String sein, da durch die Konkatenation mit einem String („:“) alle anderen Datentypen in String umgewandelt werden.

**15. Beschreibe die Vor- und Nachteile von ArrayList und Array!**

Eine ArrayList hat den Vorteil, dass sie unbegrenzt wachsen kann, bei der Instanzierung muss die Größe nicht bekannt sein wie beim Array. Sie ist im Vergleich zum Array aber sehr langsam.

**16. Gib zwei Möglichkeiten an, eine ArrayList zu durchlaufen!**

a) mit einer While-Schleife

```
int index=0;
while (index < mylist.size()){
    tu_was(mylist.get(index));
    index++;
}
```

b) mit einem Iterator

```
Iterator it=mylist.iterator();
while (it.hasNext()){
    tu_was(it.next());
}
```

**16. Wozu ist der CAST-Operator notwendig, wenn man ein Objekt aus einer Sammlung holt.**

Mit dem CAST-Operator wird der Datentyp festgelegt. In einer Sammlung lassen sich alle Arten von Objekten speichern, ohne dass die Sammlung weiß, um welchen Typ es sich handelt. Dies muss beim Herausholen explizit angegeben werden.