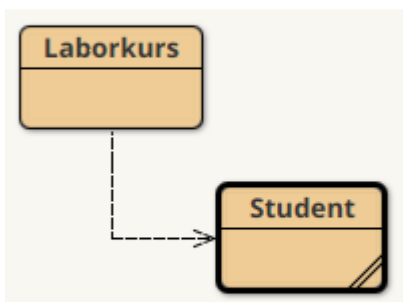


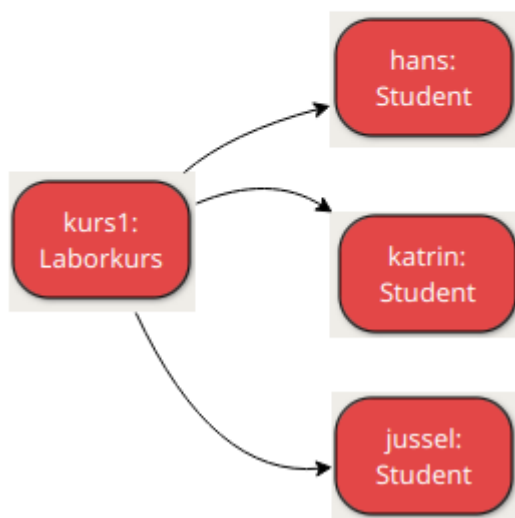
# Lösungsvorschläge zum Kapitel 3

## Übung 3.1

**Klassendiagramm:**  
Die Klasse "Laborkurs" verwaltet  
Objekte der Klasse "Student".



**Objektdiagramm:**  
Das Objekt "kurs1" vom Typ  
"Laborkurs" zeigt auf 3 Objekte des  
Typs "Student", und zwar "hans",  
"katrin" und "jussel"



## Übung 3.2

Ein **Klassendiagramm** ändert sich, wenn Sie den Quelltext modifizieren. Dies geschieht, indem die Beziehungen zwischen den Klassen verändert werden oder Klassen neu erstellt bzw. gelöscht werden.

## Übung 3.3

Ein **Objektdiagramm** ändert sich, während das Programm läuft. Es kann sich durch das Erstellen neuer Objekte oder den Aufruf von Methoden verändern.

## Übung 3.4

Jede Klassendefinition erzeugt in Java einen Variablentyp für Objektvariablen - die Klasse "Lehrender" kann also bei der Deklaration von Variablen verwendet werden wie beispielsweise `int`, die so deklarierte Variable kann dann auf Objekte des Typs "Lehrender" verweisen:

```
private Lehrender tutor;
```

## Übung 3.5

The screenshot shows the BlueJ IDE interface. On the left, the console window displays the following code:

```
Nummernanzeige stunden = new Nummernanzeige();
stunden.erhoeihen();
stunden.erhoeihen();
stunden.erhoeihen();
stunden.erhoeihen();
```

On the right, the class browser shows the 'Nummernanzeige' class. A red dialog box is overlaid on the console, titled 'stunden : Nummernanzeige', and contains two input fields: 'private int limit' with the value '24' and 'private int wert' with the value '4'. Below these fields is a button labeled 'Zeige statische Variablen'. The background shows a blurred code editor with some text visible, including 'Übung 3.5' and 'Wählen Sie im Menü'.

## Übung 3.6

Wenn der wert für die Anzeige der Minuten auf 0 zurückspringt, ist eine Stunde vergangen und die `erhoeihen()`-Methode der Stunden muss aufgerufen werden. man muss also jedes Mal prüfen, ob der Wert der Minuten auf Null springt, und wenn ja die Stunden erhöhen.

## Übung 3.8

```
Nummernanzeige.gibWert();
Error: Methode gibWert() ist nicht statisch und kann nicht aus einem statischen Kontext referenziert werden
```

Bei normalen Klassen und Methode muss man zuerst ein Objekt der Klasse erzeugen und kann erst anschließend die Methode dieses Objekts aufrufen. Der Versuch, die Methode "generisch" auf der Klasse aufzurufen schlägt fehl. wenn man Methoden benötigt, die man verwenden kann, ohne zuvor ein Objekt zu erstellen muss man "statische" Klassen und Methoden verwenden.

From: <https://info-bw.de/> -

Permanent link: <https://info-bw.de/faecher:informatik:oberstufe:bluej:kap03:lsgk3:start?rev=1728376939>

Last update: **08.10.2024 08:42**

