

Kotlin Basics

1. Funktionen definieren

- a) Funktionsschreibweise: siehe „Hefteintrag“
- b) Blockschreibweise: siehe „Hefteintrag“

2. Werte/Objekte mit einem Bezeichner versehen und unter diesem speichern

- a) Wert unveränderlich:

```
val meinWert = 35
```

Die Zahl 35 wird unter dem Bezeichner `meinWert` abgespeichert. Danach kann `meinWert` als ‚Platzhalter‘ für die Zahl verwendet werden. Der Term `meinWert + 5` hat dann also den Wert 40.

```
val meinObjekt = KlasseXY ()
```

Ein Objekt der Klasse `KlasseXY` wird erzeugt und unter dem Bezeichner `meinObjekt` abgespeichert (hier ist die Parameterliste nach dem Klassenbezeichner leer, das könnte aber auch anders sein). Danach können mit der Punktnotation Methoden des Objekts aufgerufen werden – `meinObjekt.methode1 ()` – oder es kann auf Attribute zugegriffen werden – `meinObjekt.attribut1 = „Heinzelmann“ ...`

- b) Wert veränderlich:

```
var meinWert = 35
```

Zunächst geschieht dasselbe wie oben. Später kann der unter dem Bezeichner `meinWert` abgespeicherte Wert aber verändert werden: `meinWert = meinWert + 5` bewirkt jetzt, dass der Wert 40 unter dem Bezeichner `meinWert` abgespeichert wird.

3. Grundbausteine von Algorithmen

- a) Sequenzen, also Folgen von Anweisungen, werden mit geschweiften Klammern zu sogenannten Blöcken zusammengefasst:

```
... {  
    anweisung1  
    anweisung2  
}
```

- b) Bedingte Anweisungen:

```
if (bedingung) {  
    anweisung1  
    anweisung2  
}
```

- c) Bedingte Wiederholungen:

```
while (bedingung) {  
    anweisung1  
    anweisung2  
}
```

Sonderfall: Wiederholung mit fester Anzahl – im Beispiel zehnmal

```
for (wert in 1..10) {  
    anweisung1  
    anweisung2  
}
```

4. Operatoren für gängige Aussagefunktionen und logische Funktionen (für Bedingungen)

`==` ist-gleich, `!=` ist-nicht-gleich, `>` ist-größer-als, `>=` ist-größer-oder-gleich ...

`&&` oder, `||` oder, `!` nicht