Variablen

Unveränderliche Werte speichern mit dem Schlüsselwort **`val`**

Beispiel mit einfachem Wert

```
val verbrauch100km = 4.8

fun benzinkostenBerechnen(kilometer: double, literpreis: double) =
    kilometer / 100 * verbrauch100km * literpreis

benzinkostenBerechnen(376.7, 1.239)
```

Der durchschnittliche Benzinverbrauch eines Fahrzeugs ist bekannt. Auf dieser Grundlage wird aus den jeweils gegebenen Entfernungskilometern und dem jeweils gegebenen Benzinpreis berechnet, was der Sprit für eine bestimmte Fahrt kostet.

Beispiel mit Objekten

```
val meineWelt = Welt(10, 10, 5)
val robi = Roboter(meineWelt)

robi.Schritt()
```

Je ein Objekt der Klasse `Roboter` und `Welt` wird unter einem Bezeichner gespeichert und mithilfe dieses Bezeichners verwendet.

Veränderliche Werte speichern mit dem Schlüsselwort **`var`**

Es soll abwechselnd nach links und rechts abgebogen werden:

```
...
var naechsteWenderichtung = "links"
...
while (...) {
    ...
    if (naechsteWenderichtung == "links") {
        linksUmkehren()
        naechsteWenderichtung = "rechts"
    }
    else {
        rechtsUmkehren()
        naechsteWenderichtung = "links"
    }
}
```

Unter dem Bezeichner `naechsteWenderichtung` wird abgespeichert, wohin als Nächstes abgebogen werden soll. Die bedingte Anweisung ‚prüft' den gespeicherten Wert mithilfe einer Aussagefunktion (siehe Arbeitsblatt „Kotlin Basics" unter Punkt 4) und reagiert entsprechend. Dabei wird auch der Wert der Variablen neu gesetzt.

Was steckt dahinter?

Offenbar ermöglichen es sowohl (formale) Parameter als auch **val**ues als auch **var**iables, einen Wert beziehungsweise ein Objekt im Arbeitsspeicher des Computers abzulegen, ohne sich darum zu kümmern, wie das genau funktioniert – man wählt dafür einen Bezeichner und kann mithilfe des Bezeichners wieder auf den Wert beziehungsweise das Objekt zugreifen. Nur im Fall von **var**iables ist es möglich, den Wert später zu verändern – also einen neuen Wert an denselben Speicherplatz zu schreiben.

Vokabeln: Das Erzeugen von Variablen nennt man **Deklarieren**; das Speichern eines Wertes „in" einer Variablen nennt man **Zuweisen** eines Wertes (deshalb heißt das =-Zeichen auch **Zuweisungsoperator**); das erste Zuweisen eines Wertes an eine zuvor deklarierte Variable nennt man **Initialisieren**.

Interessant: **val**ues und **var**iables müssen beim Deklarieren auch gleich initialisiert werden. Parameter werden erst initialisiert, wenn die betreffende Funktion aufgerufen wird – deswegen muss ihr Datentyp angegeben werden: Kotlin kann ihn nicht einfach aus der Angabe rechts vom Zuweisungsoperator erschließen. In allen Fällen wird der Datentyp bei der Deklaration festgelegt und kann dann nicht mehr geändert werden.

Wichtig: Variablenbezeichner können nur innerhalb des Blocks verwendet werden, in dem sie deklariert worden sind. Parameter können nur in der Definition der Funktion verwendet werden, in deren Parameterliste sie deklariert worden sind.