

Methodisches Programmierung

Übungen zur Vorlesung

Übungsblatt 1 (zu besprechen in Vorlesungswoche 2)

Beachten Sie diese Anweisungen bei allen folgenden Programmierübungen:

- *Achten Sie bei allen Funktionen, Werten und Variablen auf eine möglichst gute Benennung.*
- *Schreiben Sie für alle Methoden Tests. Schreiben Sie diese bevor Sie die Methoden implementieren.*
- *Dokumentation: Ist der Name einer Methode nicht selbsterklärend, sollten Sie zunächst überlegen, ob Sie nicht einen besseren Namen vergeben können (siehe Punkt 1). Wenn das nicht möglich ist, erklären Sie möglichst präzise in einem Code-Kommentar, was die Methode tut.*
- *Fragen Sie in der A-Übung nach, wenn die Aufgabenstellung missverständlich ist!*

1 Einführung in die Programmierung

1. Definieren Sie einen Int-Wert `constant` mit einem von Ihnen gewählten Wert und schreiben Sie eine Funktion, die prüft, ob eine Eingabe über dieser Konstanten liegt.
2. Schreiben Sie eine Funktion `square`, die das Quadrat der Eingabe berechnet.
3. Schreiben Sie eine Funktion `max`, die das Maximum zweier ganzer Zahlen zurückgibt.
4. Schreiben Sie die Funktion `abs`, die jeder ganzen Zahl ihren Absolutbetrag zuordnet: Z.B. `abs(10) = 10`, `abs(-10) = 10`.
5. Schreiben Sie eine Funktion `modulo`, die den Rest bei der Division in den ganzen Zahlen berechnet. Beispiele (für positive Parameterwerte): `modulo(7,4) = 3`, `modulo(12,6) = 0`, `modulo(7,10) = 7`. (Es gibt natürlich in Kotlin einen Modulo-Operator (%) – diesen sollten Sie jedoch hier nicht verwenden.)
 - a. Schreiben Sie die Funktion zunächst nur für den Fall $x \geq 0$ und $y > 0$.

- b. Überlegen Sie, ob die Implementierung auch für negative Eingaben verwendet werden kann, wenn man verlangt, dass zunächst der Rest bei Division der Absolutbeträge von x und y ermittelt wird (s.o.) und dann das Vorzeichen des Zählers dem Ergebnis vorangestellt wird.
- 6. Schreiben Sie die Funktion `factorial(n : Int) : Int`, die jeder natürlichen Zahl $n > 0$ das Produkt $1 * 2 * 3 * \dots * n$ ($n!$) zuordnet.
 - a. Schreiben Sie einige Tests z.B. für Eingabewerte 1, 2, 3, 4.
 - b. Schreiben Sie einen Test für `factorial(20)`. Das Ergebnis von $20!$ ist 2432902008176640000. Verändern Sie die Funktion, sodass Sie einen Wert eines Datentyps zurückliefert, der dem erwarteten Wert entspricht.

(Hinweis: Diese Aufgabe können Sie entweder mit einer Schleife lösen oder durch einen rekursiven Aufruf der Funktion (ein Aufruf von `factorial` innerhalb von `factorial`). Beides haben wir in der Vorlesung noch nicht besprochen – es ist also in Ordnung, wenn Sie hier noch nicht selbst auf die Lösung kommen ... aber vielleicht tun Sie es ja mit etwas Nachdenken doch?!)

- 7. Schreiben Sie eine Funktion, die den Schnittpunkt zweier Geraden berechnet. Nehmen wir an, eine Gerade $y = a * x + b$ ist definiert durch zwei Parameter a und b . Die Funktion sollte Double-Parameter $a1$, $b1$, $a2$ und $b2$ für zwei Geraden akzeptieren und den Schnittpunkt als String der Form „ $\langle a \rangle$, $\langle b \rangle$ “ zurückgeben.

(Diese Aufgabe ist etwas knifflig und erfordert ein bisschen mathematisches Denken. Kleiner Hinweis: Sie können herleiten, an welchem x -Wert der Schnittpunkt liegt – Achtung: wenn es einen gibt! – und dann leicht den zugehörigen y -Wert ermitteln...)

- a. Schreiben Sie eine Variante der Funktion, die als Eingabe vier Punkte akzeptiert, in Form von je zwei Double-Parametern für die X- und Y-Koordinaten dieser Punkte. Die ersten beiden dieser Punkten sind zwei Punkte auf der ersten Gerade, welche sie dadurch definieren. Die letzten zwei Punkte definieren die zweite Gerade.