

## Übungen zu Einführung in die Informatik I

### Aufgabe 31      **Nichtstrikte Funktionen**

Betrachten Sie folgende boolesche Ausdrücke und die zugehörigen Resultate:

```
# true || (1=1/0) ;;  
- : bool = true  
# false || (1=1/0) ;;  
Exception: Division_by_zero.  
# false && (1/0=2/0) ;;  
- : bool = false
```

Der bedingte Ausdruck `if ... then ... else ...` kann als dreistellige Funktion betrachtet werden. Betrachten Sie unter diesem Gesichtspunkt folgende Ausdrücke:

```
# if true then 1 else 2/0 ;;  
- : int = 1  
# if false then 1/0 else 2 ;;  
- : int = 2
```

Was folgt aus diesen Ausdrücken für die Auswertung von Konjunktion, Disjunktion und bedingten Ausdrücken in Ocaml?

### Aufgabe 32      **Primfaktorzerlegung**

Jede natürliche Zahl  $n \geq 2$  ist entweder selbst eine Primzahl oder lässt sich als Produkt von Primzahlen schreiben; d. h. in Primfaktoren zerlegen, z. B.  $120 = 2 * 2 * 2 * 3 * 5$ . In dieser Aufgabe ist ein Java-Programm zu entwickeln, das eine natürliche Zahl  $n \geq 2$  in Primfaktoren zerlegt und das Ergebnis auf dem Bildschirm ausgibt; z. B. `primfak(120)` soll die Ausgabe `2 2 2 3 5` liefern.

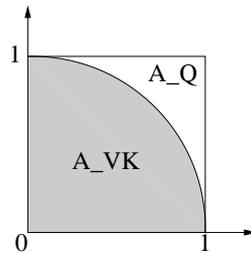
- Implementieren Sie eine Java-Methode `kleinsterTeiler`, die für eine natürliche Zahl  $n$  die kleinste natürliche Zahl  $d > 1$  ermittelt, die  $n$  ohne Rest teilt.
- Implementieren Sie eine Java-Methode `primfak`, die die einzelnen Primfaktoren am Bildschirm ausgibt.

### Aufgabe 33      **Berechnung von $\pi$ nach der Monte-Carlo-Methode**

Mit der Monte-Carlo-Methode kann der Wert von  $\pi$  angenähert werden. Dieses Verfahren setzt zufällige Punkte  $P(x,y)$  in das Quadrat und stellt anschließend fest, ob der gesetzte Punkt im Viertelkreis (Treffer) liegt.  $P(x,y)$  ist ein Treffer, falls  $d^2 = x^2 + y^2 \leq r^2 = 1$ .

Das Verfahren benutzt folgende Annahme: Die Flächen von Quadrat und Viertelkreis verhalten sich wie die jeweilige Anzahl der Zufallspunkte darin:

$$\frac{\text{treffer}}{\text{anzahl}} = \frac{A_{VK}}{A_Q} = \frac{\frac{1}{4} \cdot \pi}{1} = \frac{\pi}{4}$$



Implementieren Sie das Verfahren in Java und geben Sie den Wert von  $\pi$  nach 1000 Versuchen aus. (Hinweis: Zum Erstellen von Zufallszahlen können Sie die Methode `nextDouble()` aus der Klasse `java.util.Random` verwenden).

- a) Schreiben Sie das Java Programm Rekursiv, d. h. der für die Berechnung relevante Code steht hinter dem `return` Statement.
- b) Schreiben Sie das Java Programm Iterativ, d. h. die Berechnung erfolgt innerhalb einer `for` oder `while` Schleife.

### Aufgabe 34      **Potenz und allgemeine Wurzel**

- a) Schreiben Sie eine iterative Java-Methode, die zu einer Fließkommazahl  $x$  und einer ganzen Zahl  $m$ , den Wert  $x^m$  berechnet.
- b) Schreiben Sie eine iterative Java-Methode, die für eine positive Fließkommazahl  $x$  die  $n$ -te Wurzel  $\sqrt[n]{x}$ ,  $n \in \mathbb{N}$  mittels Intervallschachtelung (Halbierungsverfahren, siehe Aufgabe 7d/e) berechnet. Die Genauigkeit soll dabei angegeben werden. (Die  $n$ -te Wurzel  $y = \sqrt[n]{x}$  ergibt sich dabei als Lösung der Gleichung  $y^n = x$ . Bei der Lösung sollen keinerlei Funktionen aus der Bibliothek mathematischer Funktionen verwendet werden!)

### Aufgabe 35 (H) **Binomialkoeffizienten und Pascalsches Dreieck** (10 Punkte)

Die Binomialkoeffizienten sind positive ganze Zahlen, die im binomischen Lehrsatz auftreten, und die Koeffizienten der einzelnen Potenzprodukte bei der Berechnung von  $(a + b)^n = \sum_{k=0}^n \binom{n}{k} a^k \cdot b^{n-k}$  darstellen. Die rekursive Formel für ihre Berechnung ist:  $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$ , wobei  $\binom{n}{n} = \binom{n}{0} = 1$ ,  $n \geq k$ . Das auf *Blaise Pascal* zurückgehende **Pascalsche Dreieck** enthält die Binomialkoeffizienten, wobei sie im Dreieck derart angeordnet sind, dass ein Eintrag die Summe der zwei darüberstehenden Einträge ist.

$$\begin{array}{ccccccc} & & & & 1 & & & & \\ & & & & 1 & 1 & & & \\ & & & 1 & 2 & 1 & & & \\ & & 1 & 3 & 3 & 1 & & & \\ & 1 & 4 & 6 & 4 & 1 & & & \\ 1 & 5 & 10 & 10 & 5 & 1 & & & \\ & & \dots & & & & & & \end{array}$$

#### **Aufgabenstellung:**

- a) Implementieren Sie eine Java Methode, die die Binomialkoeffizienten  $\binom{n}{k}$  rekursiv berechnet und auf dem Bildschirm ausgibt. Kümmern Sie sich nicht um die Formatierung der Ausgabe.

- b) Entwerfen Sie eine weitere Methode, die in Anlehnung an das Pascalsche Dreieck die Zeilen beginnend mit der obersten (1) iterativ berechnet und ausgibt. Tipp: Legen Sie ein rechteckiges Feld doppelter Breite an, und füllen Sie die Felder ausserhalb des Dreiecks und zwischen den eigentlichen Einträgen mit Nullen auf.
- c) Betten Sie ihre Methoden in eine Klasse ein und vergleichen Sie die Ausführungsgeschwindigkeiten beider Methoden für  $n = 30$ .