Übung 2 – Aufgabenstellung

Aufgabe 1: Schreiben Sie eine Funktion, die n über k rekursiv über das Pascalsche Dreieck ermittelt. Benutzen Sie dafür die rekursive Definition.

def pascal(c: Int, r: Int): Int = ???

Aufgabe 2: Die Fibonacci-Folge ist eine unendliche Folge von Zahlen (den Fibonacci-Zahlen), bei der sich die jeweils folgende Zahl durch Addition ihrer beiden vorherigen Zahlen ergibt: 0, 1, 1, 2, 3, 5, 8, 13, … Benannt ist sie nach Leonardo Fibonacci, der damit 1202 das Wachstum einer Kaninchenpopulation beschrieb.

Die Fibonacci-Folge $f_0,\ f_1,\ f_2,\ \cdots$ ist durch das rekursive Bildungsgesetz

$$f_n = f_{n-1} + f_{n-2} \text{ für } n \geq 2$$

mit den Anfangswerten

$$f_0 = 0 \text{ und } f_1 = 1$$

definiert. Das bedeutet in Worten:

- Für die beiden ersten Zahlen werden die Werte *null* und *eins* vorgegeben.
- Jede weitere Zahl ist die Summe ihrer beiden Vorgänger.

Schreiben Sie eine Funktion fibo (X), die für eine beliebige Zahl X, die Fibonacci-Zahl berechnet.

Aufgabe 3: Wandeln Sie die Funktion aus Aufgabe 2 so um, dass der Aufruf der Funktion fibo(100) zu einem richtigen Ergebnis kommt.

Aufgabe 4: Schreiben Sie eine Funktion, die überprüft, ob innerhalb eines Ausdrucks (eine Liste von Character) eine valide Klammerung existiert. So soll bspw.:

- Dies ist ein (kleiner) ((()Test)) - true zurückgeben und
- Dies)(() ist falsch - false.

Sie können die Funktionen isEmpty, head und tail der Klasse List verwenden.

def balance(chars: List[Char]): Boolean = ???

Verwenden Sie für die Lösungen nur Elemente aus der Funktionalen Programmierung, d.h. hier nur unveränderliche Variablen und Rekursionen.