Übung 3 – Aufgabenblatt

Aufgabe 1: Schreiben Sie eine Funktion, quersumme mit der folgenden Signatur: def quersumme(zahl:Int):Int. Sie soll die Quersumme der Zahl berechnen, die an die Funktion übergeben wurde.

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: 2520 ist die kleinste Zahl, die durch jede Zahl von 1-10 ohne Rest geteilt werden kann. Was ist die kleinste positive Zahl, die durch alle Zahlen von 1-20 ohne Rest teilbar ist? (Projekt Euler Aufgabe 5)

Schreiben Sie eine Funktion, die in Abhängigkeit von einer Zahl X berechnet, welches die kleinste Zahl ist, die durch alle Zahlen von 1..X ohne Rest teilbar ist.

(Als kleiner Tipp: Schreiben Sie erst eine Funktion, die testet, ob eine Zahl durch eine Menge von Zahl teilbar ist oder nicht. Dann lassen rufen Sie die Funktion solange auf, bis sie einen entsprechenden Wert gefunden haben.)

Aufgabe 5: Die Summe aller Primzahlen der Zahlen bis 10 ist: 2+3+5+7 =17. Schreiben Sie eine Funktion, die die Summe aller Primzahlen unter 2 Millionen bildet.

Aufgabe 6: Implementieren Sie die Methode calculatePi, die auf Basis Zufall die Zahl Pi ermittelt (Monte Carlo Algorithmus – Nachlesbar in Wikipedia) Verwenden Sie dabei keine Variablen sondern nur Rekursionen. Zufallszahlen erzeugen Sie mit der Klasse Random, die Funktion nextDouble enthält:

```
import scala.util.Random
 val x= new Random
 val y= new Random
```

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