

## 5.2 Rekursion

### a) Was beschreiben die Begriffe Rekursion, lineare Rekursion und Endrekursion

- Rekursion beschreibt das Aufrufen einer Funktion durch sich selbst. Das gilt nicht nur für Funktionen in Programmen, sondern auch für abstrakte Mathematische Funktionen. Der Selbstaufwurf muss nicht mit gleichen Parametern erfolgen, in der reinen funktionalen Programmierung erfolgt der Selbstaufwurf immer mit veränderten Parametern, da sonst eine Endlosschleife erzeugt würde.
- Linear Rekursiv sind Methoden die pro Möglicher Funktionsverzweigung maximal einen Selbstaufwurf durchführen, d.h. folgende Funktion ist nicht linear-rekursiv:

```
public void foo(int x){  
    if(true){  
        foo(2);  
        foo(3);  
    }  
}
```

- Endrekursive Funktionen übergeben als Parameter zum Selbstaufruf nicht das Ergebnis eines anderen Selbstaufrufs, d.h. folgende Funktion ist nicht end-rekursiv:

```
public void foo(int x){  
    foo(foo(3));  
}
```

### b) Gegeben seien folgende Funktionen in Scala. Geben Sie an, ob die Funktionen jeweils rekursiv, linear rekursiv und/oder endrekursiv sind.

```
/**  
 * Rekursiv  
 */  
def func1(n: Int): Int = {  
    if (n > 100) {  
        n - 10  
    }  
    else {  
        func1(func1(n + 11))  
    }  
}  
  
/**  
 * Rekursiv, End-Rekursiv, Linear-Rekursiv  
 */  
def func2(n: Int): Int = {  
    if (n == 0) {  
        0  
    }  
}
```

```
    }
    else if (n > 0) {
        func2(n - 1) - n
    }
    else {
        func2(n - 1) + n
    }
}

/**
 * Rekursiv, End-Rekursiv, Linear-Rekursiv
 */
def func3(n: Int, m: Int): Int = {
    if (n <= 0 | m <= 0) {
        n + m
    }
    else if (n > m) {
        func3(n - 1, m)
    }
    else {
        func3(n, m - 1)
    }
}

/**
 * Nicht-rekursiv
 */
def func4A(n: Int, m: Int): Int = {
    if (n >= 0) {
        func4B(n - 1, m + 1)
    }
    else {
        m
    }
}

/**
 * Nicht-rekursiv
 */
def func4B(n: Int, m: Int): Int = {
    if (n >= 0) {
        func4A(n - 2, m + 1)
    }
    else {
        m
    }
}
```