

# Fragen zum F# BootCamp

6. Mai 2009

# Theorie

1. Was versteht man unter dem Begriff “Funktionale Programmierung”? Versuchen Sie den Begriff von „Imperativer Programmierung“ abzugrenzen.
2. Erklären Sie die Bedeutung des “**let**” Keywords. Warum spricht man bei F# nicht von Variablen sondern von „(let-)Bindings“?
3. Was ist eine Rekursion? Erklären Sie warum es oft sinnvoll ist Funktionen “tail-recursive” zu formulieren? Hinweis: Was passiert bei folgendem C# Programm (probieren Sie es aus) und wie könnte man es umformulieren um dem Problem zu entgehen?

```
public static Int64 Factorial(Int64 x) {  
    if (x == 0) return 1;  
    return x*Factorial(x - 1);  
}
```

..

```
Factorial(100000);
```

4. Versuchen Sie die Begriffe “Currying” und “Partial Application” an einem einfachen Beispiel zu erläutern. Hinweis: Verwenden Sie den Pipe-Operator `|>`.
5. Was versteht man unter “Lazy evaluation“ bzw. was unter “Eager evaluation“? Geben Sie Beispiele an, wo die Benutzung jeweils sinnvoll ist.
6. Erklären Sie die Begriffe “Structural Comparison”, “Structural Equality” und “Structural Hashing”.
7. Worin unterscheiden sich „Pure Functions“ von „Impure Functions“? Welche Vorteile können sich aus der Nutzung von „Pure functions“ ergeben?
8. Was versteht man unter Typinferenz? Erklären Sie warum man in F# (in den meisten Fällen) die Typdeklaration weglassen kann und die Sprache trotzdem typsicher ist. Worin liegt der Unterschied bzgl. der Variablendeklaration in Scriptsprachen wie z.B. PHP?
9. Was versteht man unter “Funktionen höherer Ordnung” (“higher-order functions“)? Versuchen Sie einfache Beispiele solcher Funktionen zu finden.

10. Erklären Sie das Prinzip der vier eingebauten F#-Funktionen “List.iter”, “List.filter”, “List.map” und “List.fold\_left”.
11. Wo befindet sich der Einsprungspunkt in ein F# Programm? Hinweis: Alle F# Module werden als statische Klassen kompiliert und “freier Quellcode” in den Konstruktor dieser Klassen.
12. Versuchen Sie Gemeinsamkeiten und Unterschiede der beiden Klassen **Dictionary** (System.Collections.Generic) und **Map** (Microsoft.FSharp.Collections) zu finden. Obwohl eine Map für die wichtigsten Operationen (Einfügen, Lookup und Löschen)  $O(\log n)$ -Laufzeit benötigt, während ein Dictionary immer konstante Laufzeit ( $O(1)$ ) bietet, gibt es Szenarien wo eine Map von Vorteil sein könnte. Versuchen Sie ein solches Beispiel zu beschreiben.
13. Was versteht man unter „Memoization“? Wie könnte man Memoization in .NET nutzen?

# Programmieraufgaben

1. Schreiben Sie ein Programm (z.B. in C# oder F#), das für die ersten 100 natürlichen Zahlen „foo“ ausgibt wenn die Zahl durch drei teilbar ist und „bar“ wenn sie durch fünf teilbar ist.
2. Schreiben Sie ein Programm (z.B. in C# oder F#), das alle Fibonacci-Zahlen bis 40 auflistet. Versuchen Sie die Laufzeit Ihres Programms zu messen. Die Fibonacci-Zahlen sind folgendermaßen rekursiv definiert:

$$\begin{aligned}fib(0) &= 1 \\fib(1) &= 1 \\fib(x) &= fib(x - 1) + fib(x - 2)\end{aligned}$$

3. Schreiben Sie ein Programm (z.B. in C# oder F#), das die “Summe der Quadrate” und das “Quadrat der Summe” der ersten 1000 Zahlen ausgibt. Hinweise:
  - a)  $1^2 + 2^2 + \dots + 10^2 = 385$  (Summe der ersten 10 Quadrate)
  - b)  $(1 + 2 + \dots + 10)^2 = 55^2 = 3025$  (Quadrat der ersten 10 Zahlen)
4. Schreiben Sie ein Programm (z.B. in C# oder F#), das das größte Palindrom findet, das sich als Produkt zweier dreistelliger Zahlen darstellen lässt. Hinweise:
  - a) Ein Palindrom ist ein Wort oder eine Zahl die sich vorwärts und rückwärts gleich liest. Beispiele: Anna, 1054501, lagerregal
  - b) Das größte Palindrom, das sich als Produkt von zwei zweistelligen Zahlen darstellen lässt ist  $9009 = 91 \cdot 99$
5. Schreiben Sie ein Programm (z.B. in C# oder F#), das berechnet wie viele Sonntage im 20. Jahrhundert (01.01.1901 bis 31.12.2000) auf einen Monatsanfang gefallen sind.
6.  $2^{15} = 32768$  und hat die Quersumme  $3 + 2 + 7 + 6 + 8 = 26$ . Was ist die Quersumme von  $2^{1000}$ ?
7. Die Summe der Primzahlen unter 10 ist  $2 + 3 + 5 + 7 = 17$ . Bestimmen Sie die Summe aller Primzahlen unter 2 Millionen.

8. Die Fakultät ist folgendermaßen rekursiv definiert:

$$\begin{aligned}0! &= 1 \\ n! &= (n - 1)! \cdot n\end{aligned}$$

Bestimmen sie die Quersumme (Summe aller Ziffern) von 100!

9. Ein pythagoreisches Tripel ist eine Menge von drei natürlichen Zahlen  $(a, b, c)$  für die gilt:  $a < b < c$  und  $a^2 + b^2 = c^2$ . Beispielsweise ist  $\{3, 4, 5\}$  ein bereits sehr lange bekanntes pythagoreisches Tripel, das benutzt wurde um rechte Winkel abzumessen. Es gibt genau ein pythagoreisches Tripel, für das gilt:  $a + b + c = 1000$ . Finden Sie dieses.
10. Finden Sie das größte Produkt von fünf aufeinanderfolgender Ziffern in der folgenden 1000-stelligen Zahl:

```
73167176531330624919225119674426574742355349194934
96983520312774506326239578318016984801869478851843
85861560789112949495459501737958331952853208805511
12540698747158523863050715693290963295227443043557
66896648950445244523161731856403098711121722383113
62229893423380308135336276614282806444486645238749
30358907296290491560440772390713810515859307960866
70172427121883998797908792274921901699720888093776
65727333001053367881220235421809751254540594752243
52584907711670556013604839586446706324415722155397
53697817977846174064955149290862569321978468622482
83972241375657056057490261407972968652414535100474
82166370484403199890008895243450658541227588666881
16427171479924442928230863465674813919123162824586
17866458359124566529476545682848912883142607690042
24219022671055626321111109370544217506941658960408
07198403850962455444362981230987879927244284909188
84580156166097919133875499200524063689912560717606
05886116467109405077541002256983155200055935729725
71636269561882670428252483600823257530420752963450
```

- a) In der E-Mail befindet sich die Zahl als C#-String zum Kopieren.