Durchreichen

Fibonacci

$$f(n) = \begin{cases} 1 & wenn\ n < 2 \\ f(n-1) + f(n-2) & sonst \end{cases}$$

Das heißt wir wissen:

$f(0) = 1 \rightarrow$ a

$f(1) = 1 \rightarrow b$

```
fib(int a, int b, int n) {
}
```

→ Das größere Glied ist immer b

Überlegung, wie sähe der Aufruf für $f(2)$ und $f(3)$ aus?

$f(2) = f(1) + f(0) = b + a$

$f(3) = f(2) + f(1) = b + a,\ b$

Durchreichen

$f(2) = f(1) + f(0) = b + a$

$f(3) = f(2) + f(1) = b + a,\ b$

Da das **größere Folgeglied immer b** ist, ist f(2) unser neues b
f(1), also das alte b wird zum neuen a

```
public static int fibEndRek (int a, int b, int n) {
        if (n== 0 || n == 1) {
                return b;
        }
        return fibEndRek (b, b + a, n - 1);
}
```

Durchreichen

Noch einmal der Unterschied zwischen Ent- und Endrekursivierung

Eine Funktion ist **endrekursiv**, wenn der r**ekursive Funktionsaufruf die letzte Aktion** der Berechnung von f ist

Vorteil: **Kein zusätzlicher Platz** zur Verwaltung der Rekursion

Eine Funktion ist **entrekursiviert**, wenn es **keine rekursiven Aufrufe** mehr gibt. Die Funktion ist dann iterativ.

Durchreichen

Durchreichen von Zwischenergebnissen (Klausur Feb 2009)

$$f(n) = \begin{cases} n+1 & \text{für } n < 3 \\ 1 + \Big(\big(\big(f(n-1) - f(n-2)\big) * f(n-3)\big) \;\%\; 100\Big) & sonst \end{cases}$$

Die Funktion für das Durchreichen wird wie folgt aufgerufen:
lin(1, 2, 3, n); wichtig!

Was wissen wir?

$f(0) = 1$
$f(1) = 2$
$f(2) = 3$

```
lin(int a, int b, int c, int steps) {
…
}
```

Durchreichen

$$f(n) = \begin{cases} n+1 & \text{für } n < 3 \\ 1 + \Big(\big(\big(f(n-1) - f(n-2)\big) * f(n-3)\big) \,\%\, 100\Big) & sonst \end{cases}$$

$f(0) = 1$ → a
$f(1) = 2$ → b
$f(2) = 3$ → c

siehe Methodensignatur

Überlegung: wie sähe das für f(3) aus?

$$f(3) = 1 + \Big(\big(\big(f(2) - f(1)\big) * f(0)\big) \,\%\, 100\Big)$$

Überlegung: wie sähe das für f(4) aus?

$$f(4) = 1 + \Big(\big(\big(f(3) - f(2)\big) * f(1)\big) \,\%\, 100\Big)$$

Allgemein:

$$f(n) = 1 + \Big(\big(\big(c - b\big) * a\big) \,\%\, 100\Big)$$

Durchreichen

$f(0) = 1$ → a

$f(1) = 2$ → b

$f(2) = 3$ → c

Überlegung: wie sähe das für f(3) aus?

$$f(3) = 1 + \Big(\big((f(2) - f(1)) * f(0)\big)\ \%\ 100\Big)$$

Überlegung: wie sähe das für f(4) aus?

$$f(4) = 1 + \Big(\big((f(3) - f(2)) * f(1)\big)\ \%\ 100\Big)$$

Man sieht:
Das berechnete f(3) ist das neue c in f(4)
Das alte c f(2) wird zum neuen b
Das alte b f(1) wird zum neuen a

Durchreichen

Somit sieht der Code so aus:

```
lin (int a, int b, int c, int steps) {

        if (steps == 0) {
                return a;
        }

        return lin (b, c, lin(b, c, 1+(((c-b)*a)%100), steps-1);
}
```

Das alte b wird zum neuen a

Das alte c wird zum neuen b

Das berechnete f(n) ist das neue c in f(n+1)