

Introduktion til Programmering og Problemløsning (PoP)

Jon Sparring
Department of Computer Science
2021/09/23

UNIVERSITY OF COPENHAGEN



Hvor langt er I kommet med materialet?

<https://tinyurl.com/yajt7smf>

Regneregler

<https://tinyurl.com/nn3rybf6>

Operator	Associativity	Description
<code>+<expr></code> , <code>-<expr></code> , <code>~~~<expr></code>	Left	Unary identity, negation, and bitwise negation operator
<code>f <expr></code>	Left	Function application
<code><expr> ** <expr></code>	Right	Exponent
<code><expr> * <expr></code> , <code><expr> / <expr></code> , <code><expr> % <expr></code>	Left	Multiplication, division and remainder
<code><expr> + <expr></code> , <code><expr> - <expr></code>	Left	Addition and subtraction binary operators
<code><expr> ^^^ <expr></code>	Right	bitwise exclusive or
<code><expr> < <expr></code> , <code><expr> <= <expr></code> , <code><expr> > <expr></code> , <code><expr> >= <expr></code> , <code><expr> = <expr></code> , <code><expr> <> <expr></code> , <code><expr> <<< <expr></code> , <code><expr> >>> <expr></code> , <code><expr> &&& <expr></code> , <code><expr> <expr></code> ,	Left	Comparison operators, bitwise shift, and bitwise 'and' and 'or'.
<code><expr> && <expr></code>	Left	Boolean and
<code><expr> <expr></code>	Left	Boolean or

$2.0 ** 3.0 ** 4.0 = 2.0 ** (3.0 ** 4.0)$
Ens præcedens, højre association

$2.0 / 3.0 / 4.0 = (2.0 / 3.0) / 4.0$
Ens præcedens, venstre association

$2.0 ** 3.0 + 4.0 = (2.0 ** 3.0) + 4.0$
'**' præcedens over '+'

For-løkker

<https://tinyurl.com/3ttrm6dc>

<https://tinyurl.com/5ymk75hb>

<https://tinyurl.com/28hszu9s>

forExample.fsx

```
let mutable sum = 1
for i = 1 to 3 do
    sum <- sum + i
printfn "sum er %d" sum
```

forExample2.fsx

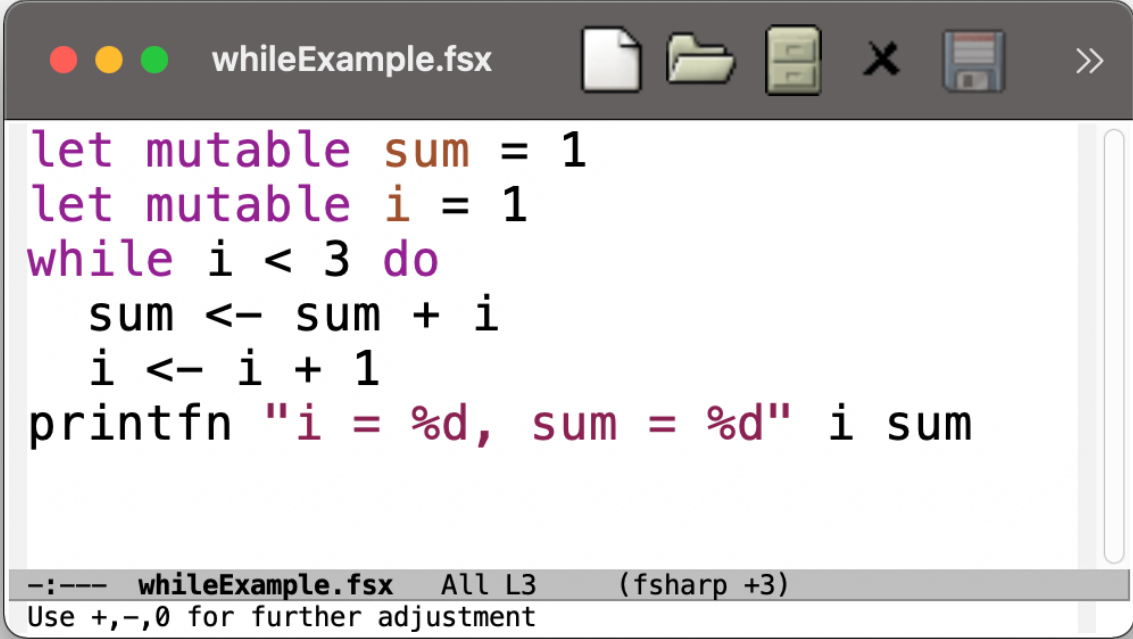
```
for i = 1 to 3 do
    let mutable sum = 1
    sum <- sum + i
printfn "sum er %d" sum
```

forExample3.fsx

```
let mutable sum = 1
for i = 1 to 3 do
    let mutable sum = 1
    sum <- sum + i
printfn "sum er %d" sum
```

While-løkker

<https://tinyurl.com/nu67k773>



```
let mutable sum = 1
let mutable i = 1
while i < 3 do
    sum <- sum + i
    i <- i + 1
printfn "i = %d, sum = %d" i sum
```

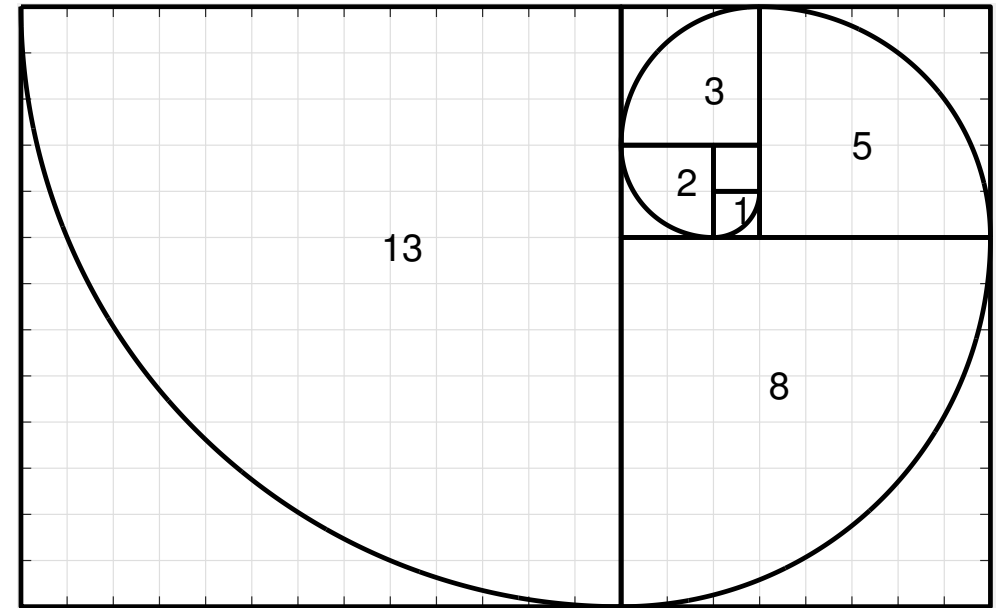
--- whileExample.fsx All L3 (fsharp +3)
Use +,-,0 for further adjustment

Fibonacci

En talsekvens efter programmet:

```
f: int -> int
f(0)=0
f(1)=1
f(i) = f(i-1) + f(i-2), i > 1
```

0, 1, 1, 2, 3, 5, 8, 13, ...

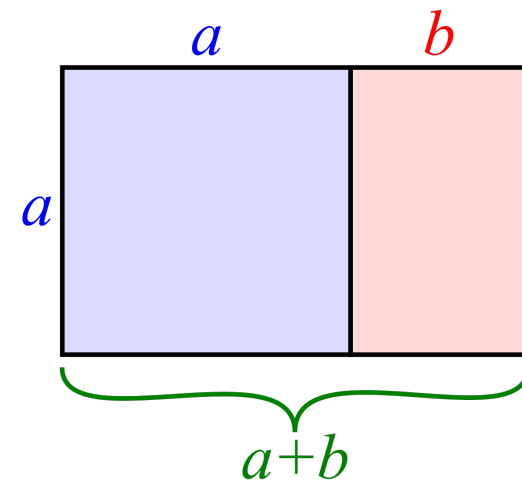


Relationer:

- Talteori, kodning, løsning af polynomier, kompleksitetsanalyse
- Gyldne snit:

$$\frac{a+b}{a} = \frac{a}{b} = s$$

$$\lim_{i \rightarrow \infty} \frac{f(i)}{f(i-1)} = s$$



Fibonacci

$f: \text{int} \rightarrow \text{int}$
 $f(0)=0$
 $f(1)=1$
 $f(i) = f(i-1) + f(i-2), i > 1$

0, 1, 1, 2, 3, 5, 8, 13, ...

For-løkke

```
let fib n =  
  if n < 1 then 0  
  else  
    let mutable prevPrev = 0  
    let mutable prev = 1  
    for i = 2 to n do  
      let curr = prev + prevPrev  
      prevPrev <- prev  
      prev <- curr  
    prev
```

While-løkke

```
let fib n =  
  if n < 1 then 0  
  else  
    let mutable prevPrev = 0  
    let mutable prev = 1  
    let mutable i = 2  
    while i <= n do  
      let curr = prev + prevPrev  
      prevPrev <- prev  
      prev <- curr  
      i <- i + 1  
    prev
```

Tupple + for-løkke

```
let fib n =  
  if n < 1 then 0  
  else  
    let mutable prev = (0, 1)  
    for i = 2 to n do  
      prev <- (snd prev, (fst prev) + (snd prev))  
    snd prev
```

Spørgetime

<https://tinyurl.com/4vuh96b7>

DIKU Bits
TUESDAY LECTURES

28 SEPTEMBER 2021

3D printing the best video game controller

Valkyrie Arline Savage, Ph.D
Assistant Professor, Human-Centred Computing, DIKU



12.15 - 13.00
diku.dk/diku-bits