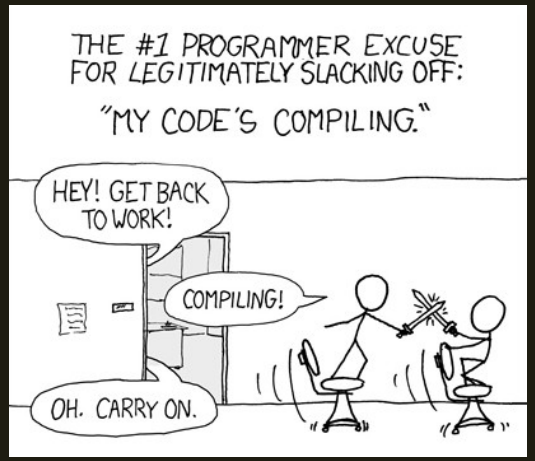
TECHNISCHE UNIVERSITÄT
BERGAKADEMIE FREIBERG
Die Ressourcenuniversität. Seit 1765.

Advanced Programming SS 2020: LLVM



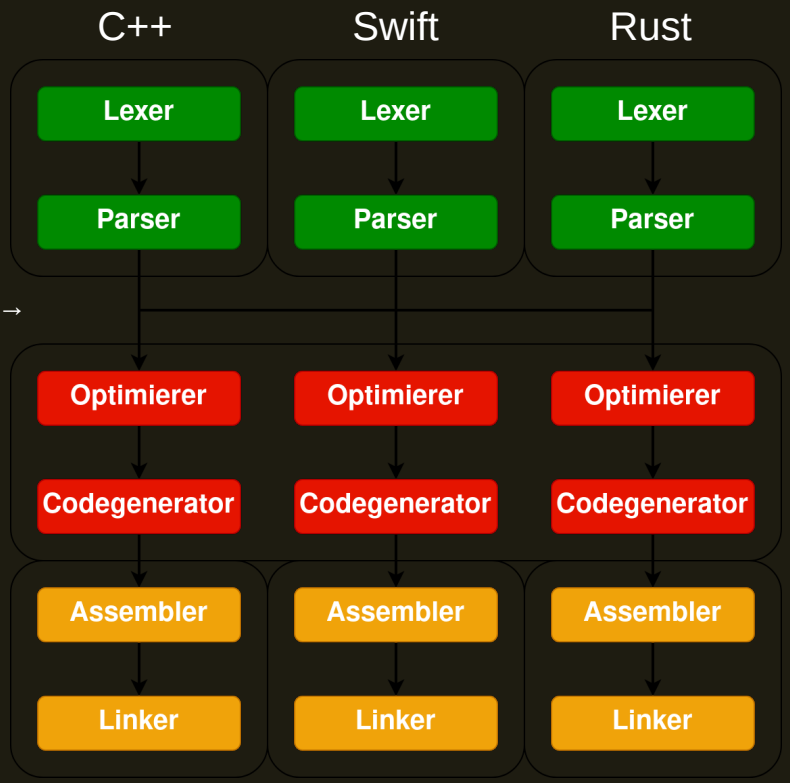
https://xkcd.com/303/

```
# Compute Ackermann's function
# for two parameters a and b.
def ack(a, b)
     if a == 0 then
          b + 1
     else if b == 0 then
          ack(a - 1, 1)
     else
          ack(a - 1, ack(a, b - 1))

def run(a, b)
     ack(a, b)
```

Ben Lorenz / Jonas Treumer

Compilerbau



Compiler Frontends

Compiler Backend

Platform Toolchains

Compilerbau

- **Parser**
  - Token vom Lexer pollen
  - Einen abstrakten Syntaxbaum (AST) aufbauen
  - Zerlegung unserer Programmiersprache in „höhersprachliche Konstrukte“:

Kommentare: ignorieren →

```
# Computes the n'th Fibonacci number
dec fib(n)

def fib(n)
    if n < 2 then
        n
    else
        fib(n - 2) + fib(n - 1)
```

Compilerbau

- **Parser**
  - Token vom Lexer pollen
  - Einen abstrakten Syntaxbaum (AST) aufbauen
  - Zerlegung unserer Programmiersprache in „höhersprachliche Konstrukte“:

```
# Computes the n'th Fibonacci number
dec fib(n)

def fib(n)
    if n < 2 then
        n
    else
        fib(n - 2) + fib(n - 1)
```

Deklaration: Top Level

Compilerbau

- **Parser**
  - Token vom Lexer pollen
  - Einen abstrakten Syntaxbaum (AST) aufbauen
  - Zerlegung unserer Programmiersprache in „höhersprachliche Konstrukte“:

```
# Computes the n'th Fibonacci number
dec fib(n)

def fib(n)
    if n < 2 then
        n
    else
        fib(n - 2) + fib(n - 1)
```

Definition:
Top Level

Compilerbau

- **Parser**
  - Token vom Lexer pollen
  - Einen abstrakten Syntaxbaum (AST) aufbauen
  - Zerlegung unserer Programmiersprache in „höhersprachliche Konstrukte“:

```
# Computes the n'th Fibonacci number
dec fib(n)
```

Prototyp

```
def fib(n)
    if n < 2 then
        n
    else
        fib(n - 2) + fib(n - 1)
```

Compilerbau

- **Parser**
  - Token vom Lexer pollen
  - Einen abstrakten Syntaxbaum (AST) aufbauen
  - Zerlegung unserer Programmiersprache in „höhersprachliche Konstrukte“:

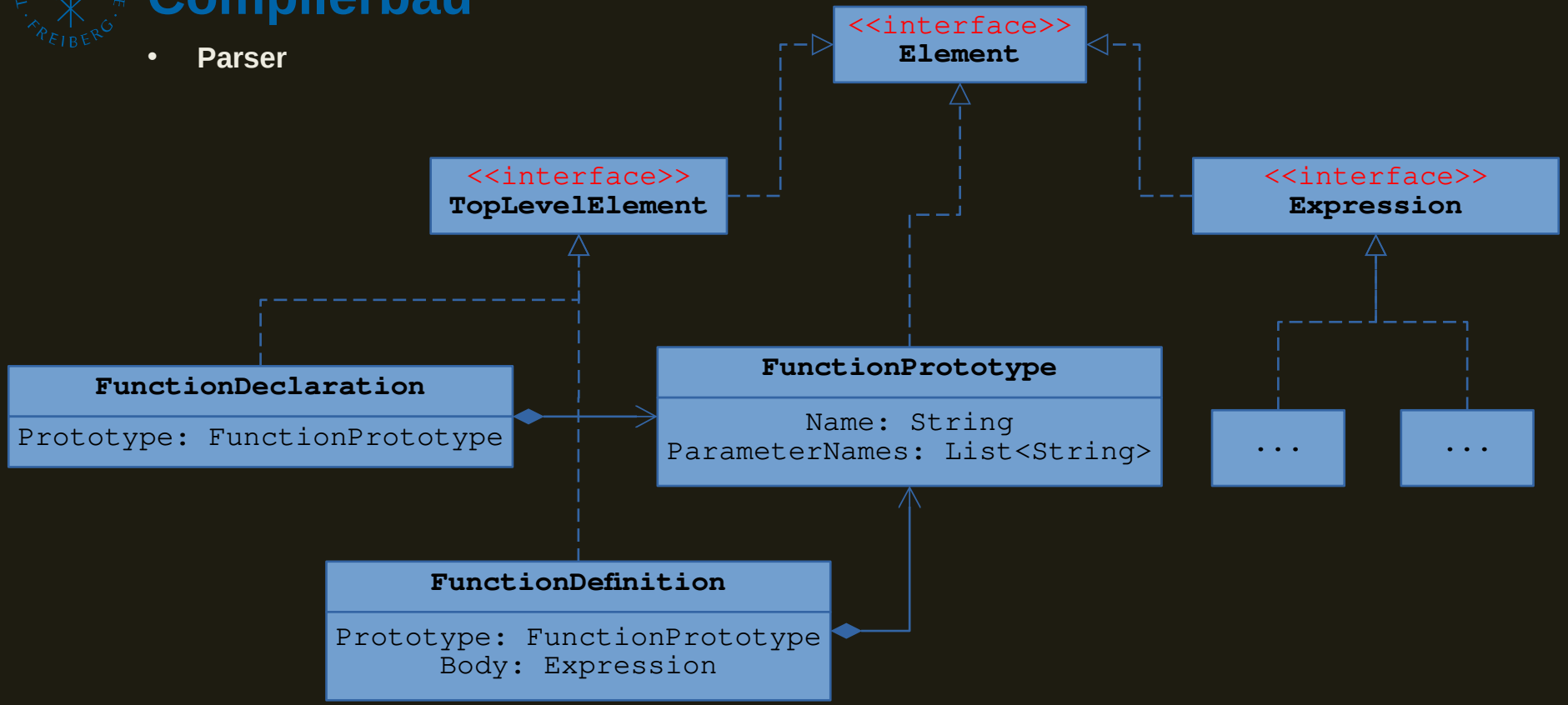
```
# Computes the n'th Fibonacci number
dec fib(n)

def fib(n)
    if n < 2 then
        n
    else
        fib(n - 2) + fib(n - 1)
```

Expression: auf Wert reduzierbar

Compilerbau

- **Parser**



Compilerbau

- **Parser**