

# Praktikum Compilerbau

## Sitzung 12 – Abschlußveranstaltung

Lehrstuhl für Programmierparadigmen  
Universität Karlsruhe (TH)

22. Juli 2009

Alle Gruppen sind bereits fertig und haben einen Übersetzer abgegeben:

- Einlesen des Quellprogramms (Parsing)
- Semantische Tests (Typprüfung)
- Optimierungen:
  - Dead-Code Elimination
  - Constant Propagation
  - Constant Folding
  - Arithmetic Identities
  - Common Subexpression Elimination
  - Scheduling
- Bytecodeerzeugung

# Optimierwettbewerb

## Ziele

- Praxistest für die Compiler!
- Gespür für Korrektheit und Benchmarks.
- Spaß

## Ergebnisse

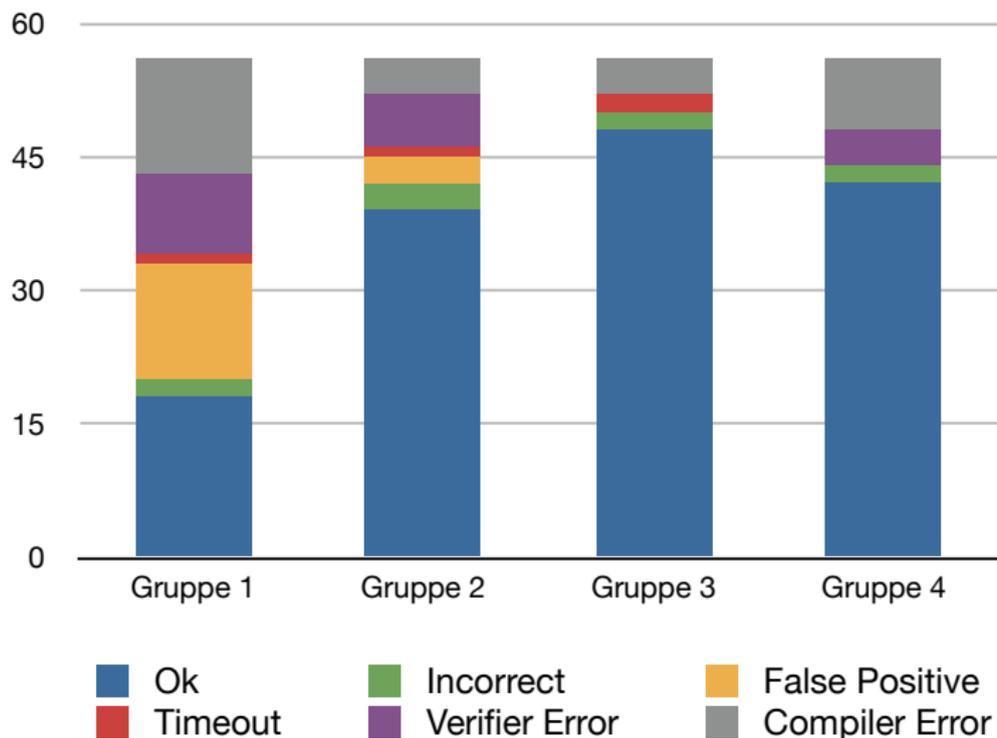
- Alle Gruppen haben einen funktionierenden Compiler abgegeben.
- Testen auf Korrektheit und erzeugten Code mit verschiedenen Benchmarks.

# Korrektheit

Testen von:

- syntaktischen Problemfälle
- Vollständigkeit der semantische Tests
- Typische Problemfälle bei der Codeerzeugung:
  - „Swap-Problem“ beim SSA-Abbau
  - Endlosschleifen
  - Return innerhalb einer Schleife
  - Jasmin Syntax

## Korrektheit Ergebnisse



# Compiler Geschwindigkeit, erzeugter Code

Testen von:

- Übersetzungsgeschwindigkeit
- Korrektheit der Resultate
- Anzahl ausgeführter Bytecodeinstruktionen (mit Hilfe eines JVM-Agents)

Testprogramme:

Empty, Hello1, opt1–opt10, Ackermann, Fibonacci, Factorial, GCD, SortedTree, TreeWalk, eingereichte Benchmarks der Gruppen (RSATest, Phonebook/Brainfuck interpreter, WhileTest, ...)

Punktevergabe:

- Je Programm 10 Punkte
- Inkorrekt → 0 Punkte
- Schnellste Compilezeit → 3 Punkte (2. Platz → 2 Punkte, ...)
- Beste Laufzeit → 7 Punkte (2. Platz → 5 Punkte, ...)

## Punkttestand

Endergebnis	Gruppe 1	Gruppe 2	Gruppe 3	Gruppe 4
Laufzeit	47	60	142	82
Compilezeit	8	26	49	49
Gesamt	55	86	191	131

# Endergebnis

## Endergebnis: Punktestand

