

# Praktikum Compilerbau

## Sitzung 4 – Semantik und Typprüfung

Lehrstuhl für Programmierparadigmen  
Universität Karlsruhe (TH)

13. Mai 2009

- 1 Letzte Woche
- 2 Semantik & Typprüfung
- 3 Sonstiges

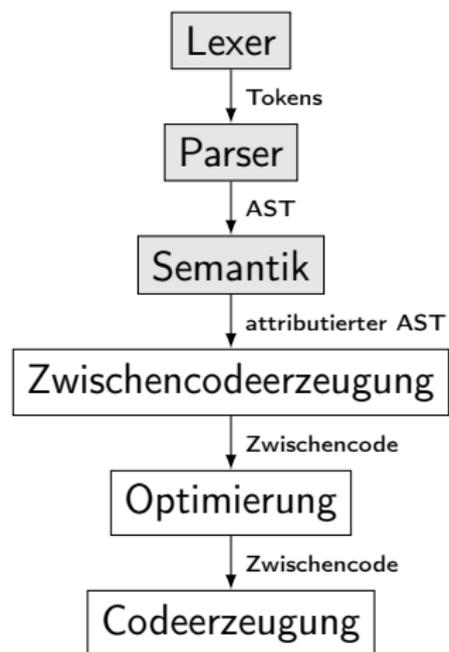
# Letzte Woche

- Beispielprogramme im Wiki:
  - Syntax: `pp.info.uni-karlsruhe.de/lehre/SS2009/comprakt/wiki/index.php/Tests_Syntax`
  - Semantik `pp.info.uni-karlsruhe.de/lehre/SS2009/comprakt/wiki/index.php/Tests_Semantik`
- Modellierung: Wie wurde Sie gewählt? (Dokumente)
- Was waren die Probleme?
- Hat soweit alles geklappt?

## Wie sieht hier der AST aus?

```
public class A {  
  
    public int x;  
    public int y;  
  
    public int calc(int x, int y) {  
        int sum;  
        if (x > y) {  
            sum = x + y;  
        } else {  
            sum = x * y;  
        }  
        return sum;  
    }  
}
```

# Compilerphasen



# Aufgaben für diese Woche

- Namensanalyse: Wofür steht der Name?
  - Definitionstabelle: Jede Definition ein Eintrag.
  - Verweise auf entsprechenden Eintrag.
- Typanalyse: Welchen Typ hat der Ausdruck?
- Semantik: Sind die Typen ok?
- Operatoridentifikation (nicht nötig)

## Namensanalyse: Attribute, Parameter, Lokale Variablen

```
class A {  
  
    public int a;  
  
    public int foo(int a) {  
        return a + this.a;  
    }  
  
    public int foo2() {  
        return a;  
    }  
  
    public int foo3(A a) {  
        return foo(this.a) + a.foo2();  
    }  
}
```

# Namensanalyse: Symboltabelle

```
public class Symbol {}  
  
public interface SymbolTable {  
    Symbol findOrInsert(String string);  
}
```

# Namensanalyse: Namenstabelle

```
public interface NameTable {  
    void enterScope();  
    void leaveScope();  
  
    void enterDefinition(Symbol symbol,  
                          Definition definition);  
    Definition getCurrentDefinition(Symbol symbol);  
    boolean definitionInCurrentScope(Symbol symbol);  
}
```

Beachte: Feldzugriff nicht über Namenstabelle

# Typanalyse

Wo kommen Typen vor? (Ausdrücke, Methoden, Felder)

Was für Typen gibt es? (Integer, Boolean, ObjektTyp, MethodenTyp?)

- $3 + 4: 3 \rightarrow int, 4 \rightarrow int, int + int \rightarrow int$
- $a$ : Auf welche Definition verweist  $a$ ? Typ ist Typ der Definition.
- $foo(), foo2(p1, p2)$ : Typ ist Returnwert.
- $3 < 4: 3 \rightarrow int, 4 \rightarrow int, int < int \rightarrow boolean$
- $new X()$ : Typ ist Objekt der Klasse  $X$ .

# Semantik

- Zuweisungen  $x = y$ :  $\text{Typ}(x) == \text{Typ}(y)$  und  $x$  ist Attribut/Variable
- Bedingungen  $\text{if } (x), \text{ while } (x)$ :  $\text{Typ}(x) == \text{Boolean}$
- Vergleiche  $x < y$ :  $\text{Typ}(x) = \text{Typ}(y) = \text{Integer}$
- Operationen  $x + y, x * y$ :  $\text{Typ}(x) = \text{Typ}(y) = \text{Integer}$
- Boolsche Operationen  $!x, x \ \&\& \ y$ :  $\text{Typ}(x) = \text{Typ}(y) = \text{Boolean}$
- Feldzugriffe  $o.f$ :  $\text{Typ}(o)$  ist ein Objekt einer Klasse die ein Attribut  $f$  besitzt
- Deklarationen `MyClass m`: Existiert eine Klasse `MyClass`?
- Methodenaufrufe  $m(p_1, \dots, p_n)$ :  $m$  hat  $n$  Parameter und jeder Ausdruck  $p_i$  hat den Typ des entsprechenden Parameters
- Return Statements `return x`:  $\text{Typ}(x)$  passt zur Methode. Return darf nicht fehlen, falls Methode nicht `void`.

# Testen!

`http://pp.info.uni-karlsruhe.de/lehre/SS2009/  
comprakt/wiki/index.php/Tests\_Semantik`

# Feedback! Fragen? Probleme?

- Laptop mitbringen (Firm wird installiert)
- Anmerkungen?
- Probleme?
- Fragen?