


Ein System für die Konstruktion objektorientierter Übersetzer

Dipl.-Inform. Peter Knauber

FB Informatik, AG Programmiersprachen und Compiler

Inhalt

-  **Motivation, Anforderungen**
- Ein Modell für den objektorientierten Übersetzerbau**
- Eine objektorientierte Bibliothek**
- Das OCC-System**
- Zusammenfassung, Ausblick**

Motivation

- ❑ **Die Komplexität eines Übersetzers korreliert mit der Komplexität der zu übersetzenden Sprache**

- ❑ **Viele Sprachen enthalten ähnliche Konzepte**

- **Wiederverwenden von Übersetzerkomponenten ist sinnvoll, um die Komplexität zu reduzieren**

Ziel der Arbeit

Ein System zur Unterstützung beim Entwickeln neuer Sprachen:

- **Sprachdefinition**
- **Schnelle Verfügbarkeit**
- **Referenz-Implementierung**

Lösungsansatz: Wiederverwenden von Komponenten

Anforderungen

... an ein System zur Unterstützung bei der Sprachentwicklung:

- ❑ **Unterstützen beim Wiederverwenden bereits realisierter Konzepte**
 - ▶ **Ziel: Komponenten zur Verfügung stellen**

- ❑ **Freies Kombinieren von Sprachkonzepten**

- ❑ **Einfaches Integrieren neuer Sprachkonzepte**

Inhalt

- ❑ **Motivation, Anforderungen**

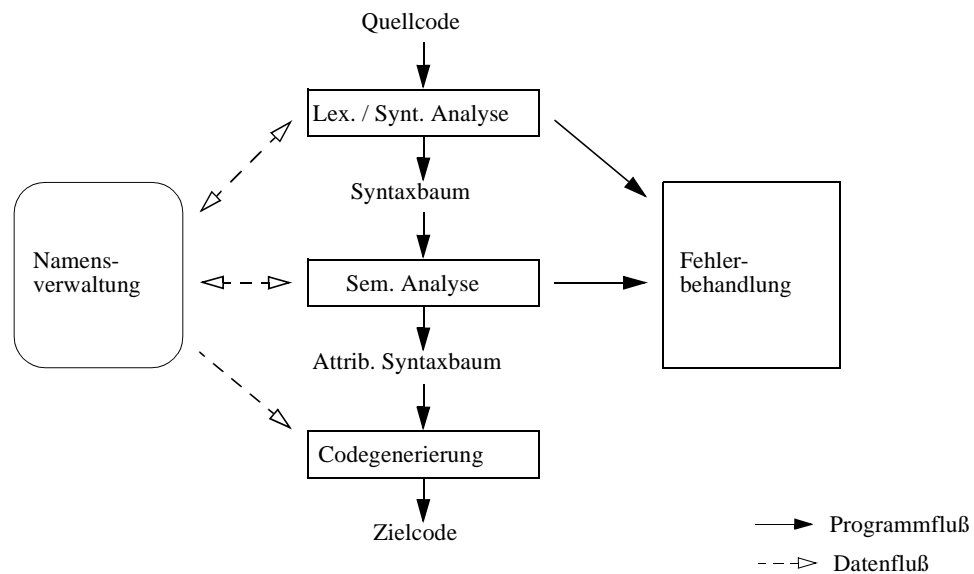
- ☞ ❑ **Ein Modell für den objektorientierten Übersetzerbau**

- ❑ **Eine objektorientierte Bibliothek**

- ❑ **Das OCC-System**

- ❑ **Zusammenfassung, Ausblick**

Traditionelle phasenorientierte Übersetzer



Phasenorientierte Übersetzer — objektorientiert

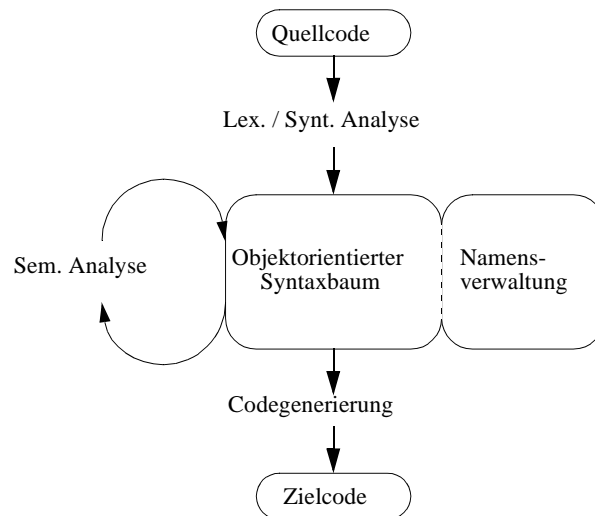
Klassen:

- Scanner
- Parser
- Sem. Analysierer
- Codegenerierer

Objektorientierte Daten:

- Einträge in die Namensverwaltung
- Elemente des Syntaxbaums

Objektorientierte Übersetzer



→ Datenfluß

Inhalt

- Motivation, Anforderungen
- Ein Modell für den objektorientierten Übersetzerbau
- ☞ Eine objektorientierte Bibliothek
- Das OCC-System
- Zusammenfassung, Ausblick

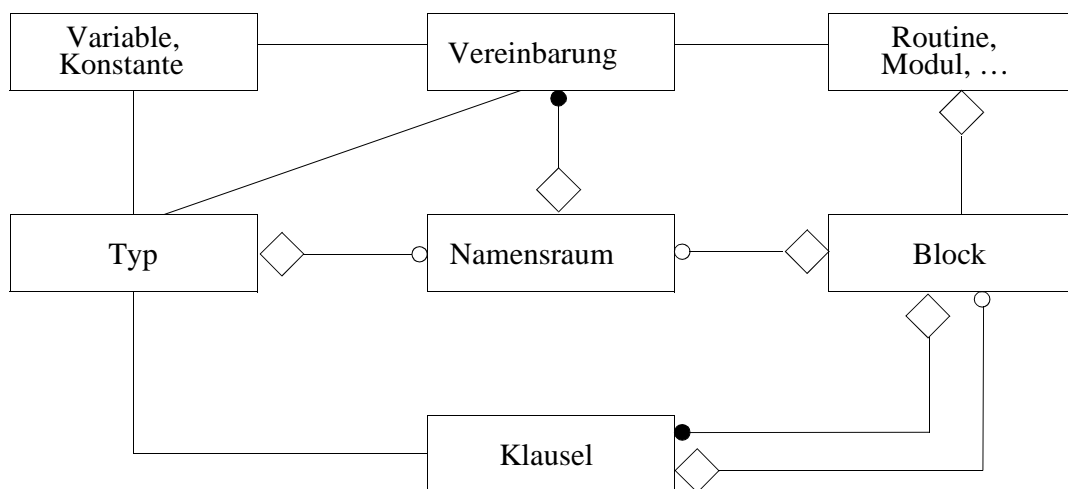
Systematik von Sprachkonzepten (1)

Untersuchte Sprachen:

- ❑ FORTRAN
- ❑ ALGOL 60, 68
- ❑ Common Lisp
- ❑ PL/I
- ❑ Simula 67, Smalltalk-80
- ❑ C, ANSI C, C++, Java
- ❑ Pascal, Modula-2, Oberon-2

Systematik von Sprachkonzepten (2)

Elementare Konzepte imperativer / objektorientierter Sprachen



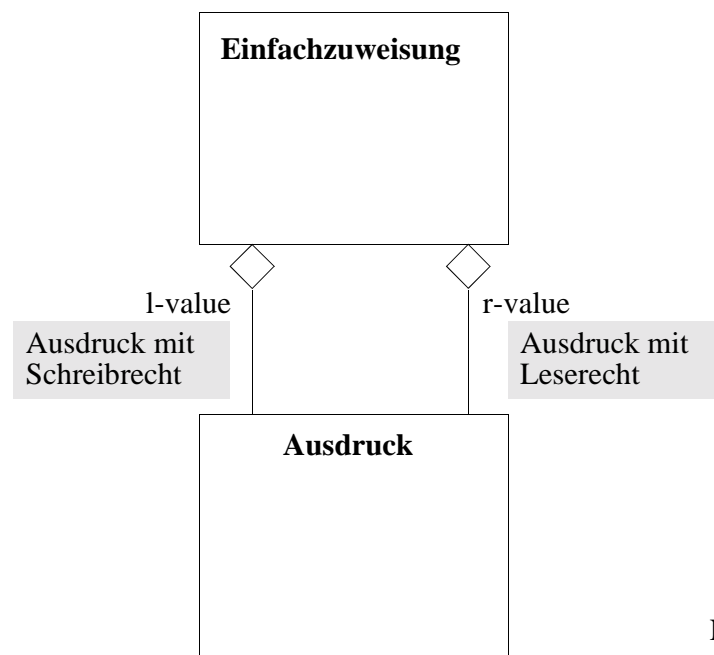
Notation: OMT

Systematik von Sprachkonzepten (3)

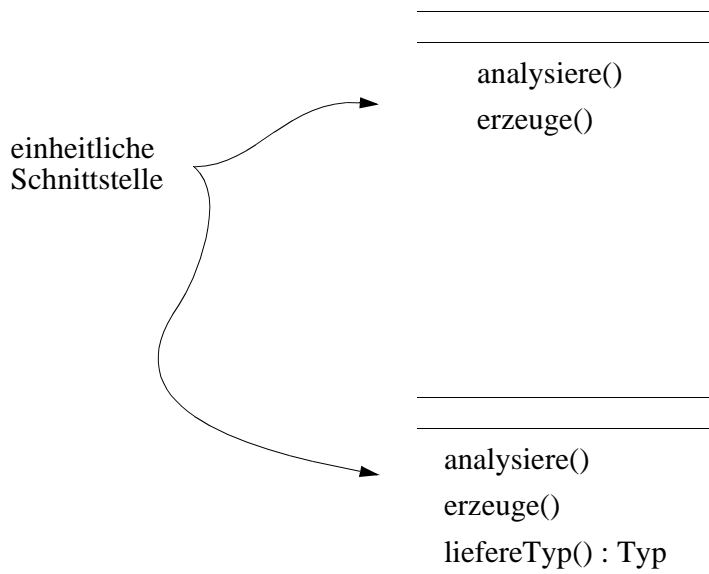
Kategorien von Sprachkonzepten:

- ❑ **Vereinbarungen:**
Typen, Variablen, Routinen, ...
- ❑ **Klauseln:**
Anweisungen, Ausdrücke
- ❑ **Blöcke:**
integrierendes Konstrukt
- ❑ **Namensräume:**
an Blöcke gebunden

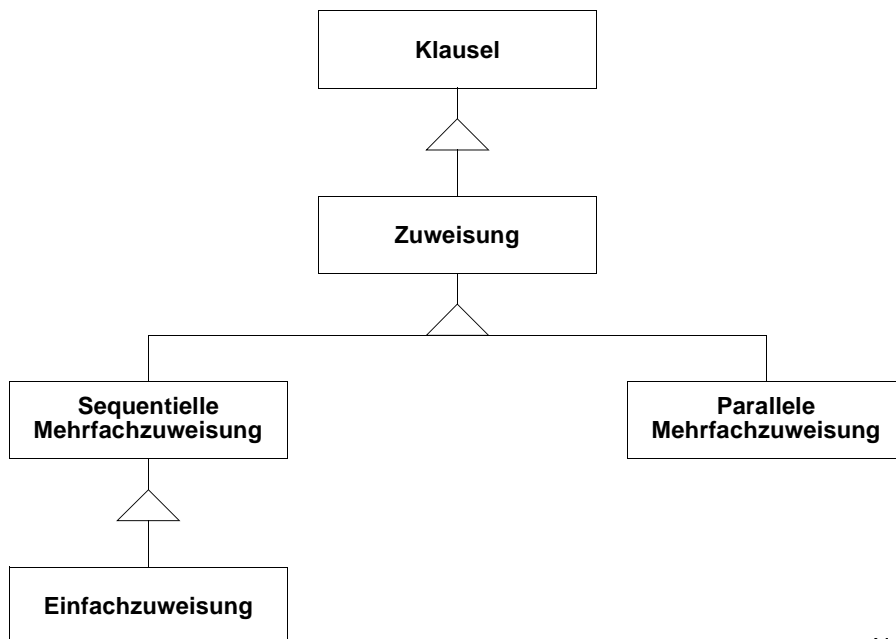
Beispiel „Einfachzuweisung“: objektorientierte Übersetzung



Notation: OMT



Beispiel „Einfachzuweisung“: Beerbung = Spezialisierung

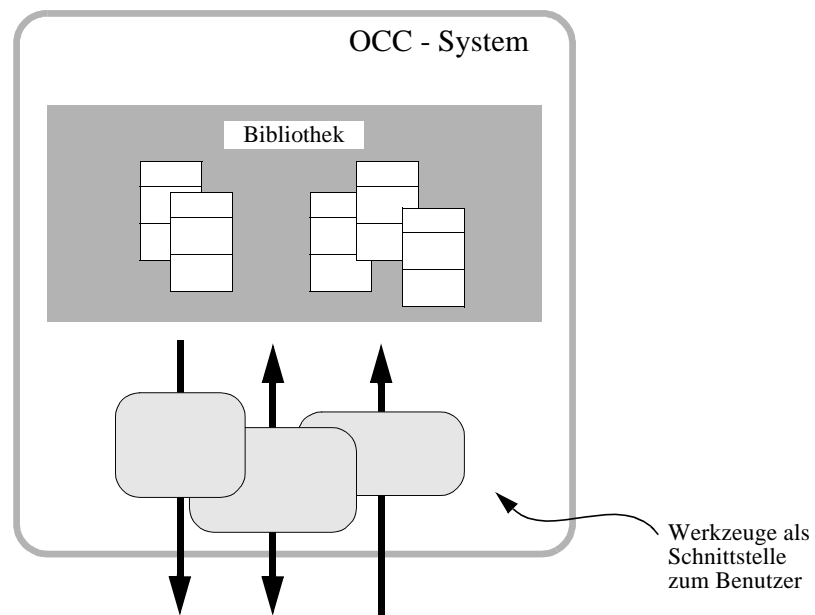


Notation: OMT

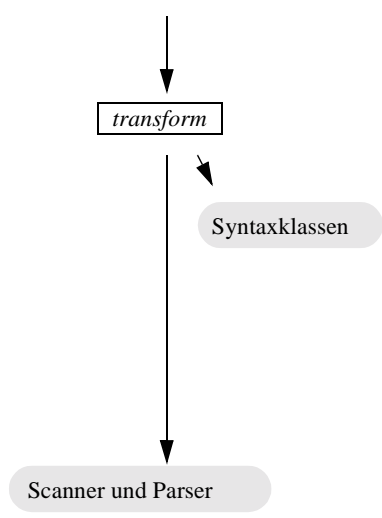
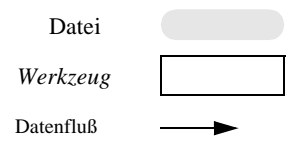
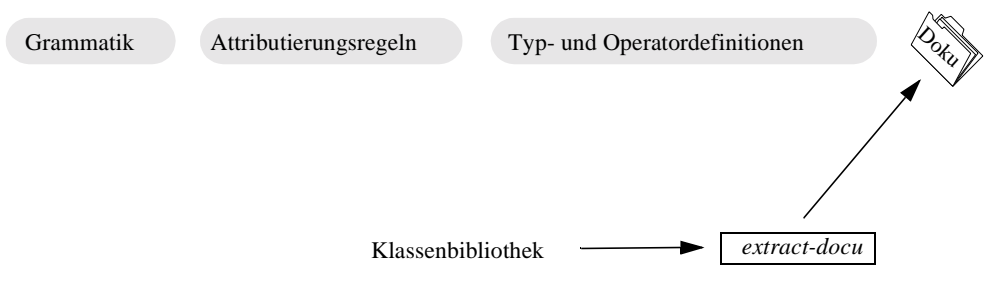
Inhalt

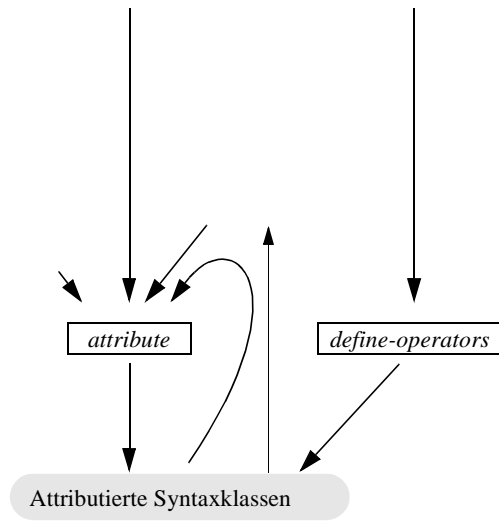
- ❑ Motivation, Anforderungen
- ❑ Ein Modell für den objektorientierten Übersetzerbau
- ❑ Eine objektorientierte Bibliothek
- ☞ ❑ Das OCC-System
- ❑ Zusammenfassung, Ausblick

Objectoriented Compiler Construction



Der Einsatz des OCC-Systems





Inhalt

- ❑ **Motivation, Anforderungen**
- ❑ **Ein Modell für den objektorientierten Übersetzerbau**
- ❑ **Eine objektorientierte Bibliothek**
- ❑ **Das OCC-System**
- ☞ ❑ **Zusammenfassung, Ausblick**

Vergleich existierender Systeme

Anforderungen		<i>Cocktail</i>	<i>Eli</i>	<i>LDL</i>	<i>OCS</i>	<i>OCC</i>
Verwenden existierender Konstrukte	möglich	(✓) ^a	✓	✓	✓	✓
	unterstützt	—	✓	(✓) ^b	—	
Freies Kombinieren von Konstrukten	möglich	(✓) ^a	✓	✓	✓	✓
	unterstützt	—	—	✓	—	
Integrieren neuer Konstrukte	möglich	✓	✓	(✓) ^c	✓	✓
	unterstützt	—	—	—	—	

a. nur funktionale Bibliotheken, z. B. für die Namensverwaltung.

b. geplante Unterstützung durch Wissensbasis

c. sehr aufwendig.

Ohne Semantik-Unterstützung: lex, yacc, SIC, ACK

Ausblick

- Bibliotheksausbau
- Testprogramme als Referenz für andere Übersetzer
- (teil-) automatisch Dokumentation und Report generieren

Erfahrungen bei der Prototyp-Implementierung

- Nachteile:**
 - Erzeugte Übersetzer sind ineffizient**
 - Benutzungsschnittstelle**

- Vorteile:**
 - Schnelle Entwicklung**
 - Konzentration auf die neuen Konzepte**
 - Konsistenzprüfung des Zwischencodes**

Daten der prototypischen OCC-Implementierung

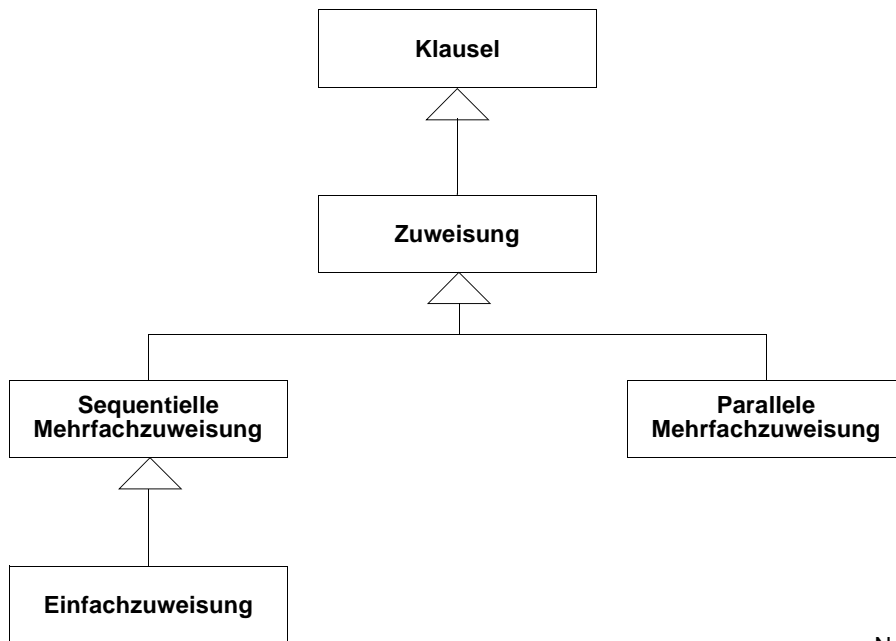
- Anzahl der Klassen: 360**

- Anzahl der Nachrichten (versch. Methoden): 130**

- Implementierungssprache: W-Lisp**

- Dokumentation als Hypertext aufgebaut**

Beispiel „Zuweisung“: Beerbung = Spezialisierung



Notation: UML