

Die Programmiersprache Smalltalk (Beispiele: IBM Smalltalk und Visual Age Version 2.0)

Dr. Monika Meiler
SS 1996

Universität Leipzig
Fakultät für Mathematik und Informatik
Institut für Informatik

Telefon: (0341) 97 32224
Email: meiler@informatik.uni-leipzig.de
URL: <http://www.informatik.uni-leipzig.de/~meiler/>

Inhalt

| | |
|--|------|
| 1 Einführung..... | 1-1 |
| 1.1 Warum objektorientierte Programmierung (OOP) ?..... | 1-1 |
| 1.2 Grundbegriffe der objektorientierten Programmierung..... | 1-2 |
| 1.2.1 Objekte | 1-2 |
| 1.2.2 Klassen | 1-3 |
| 1.2.3 Vererbung..... | 1-6 |
| 1.2.4 Polymorphismus..... | 1-9 |
| 2 ST-Sprachzusammenfassung..... | 2-1 |
| 2.1 Konventionelle Sprachelemente | 2-1 |
| 2.1.1 Das Alphabet..... | 2-1 |
| 2.1.2 Der Kommentar (<i>comment</i>)..... | 2-1 |
| 2.2 Objekte, Konstanten und Variablen..... | 2-1 |
| 2.2.1 Objekte | 2-1 |
| 2.2.2 Namen (<i>identifier</i>)..... | 2-2 |
| 2.2.3 Konstanten (<i>literal</i>)..... | 2-2 |
| 2.2.4 Variablen (<i>variable</i>)..... | 2-4 |
| 2.3 Ausdrücke | 2-10 |
| 2.3.1 Einfacher Ausdruck (<i>primary</i>) | 2-10 |
| 2.3.2 Wertzuweisung..... | 2-12 |
| 2.3.3 Nachrichtenausdruck (<i>messageExpression</i>)..... | 2-12 |
| 2.3.4 Nachrichtenkaskade (<i>cascadeMessage</i>)..... | 2-16 |
| 2.3.5 Strukturierte Nachrichtenausdrücke..... | 2-17 |

| | | |
|-------|---|------|
| 3 | ST-Klassen und Methoden | 3-1 |
| 3.1 | Das Klassenkonzept | 3-1 |
| 3.1.1 | Klassendefinition | 3-2 |
| 3.1.2 | Methodendefinition | 3-3 |
| 3.1.3 | System Browser | 3-3 |
| 3.2 | Eigene Klassen | 3-4 |
| 3.2.1 | Applications Browser | 3-4 |
| 3.2.2 | Beispiel | 3-6 |
| 3.3 | Unterklassen und Vererbung | 3-8 |
| 3.4 | Klassenhierarchie und abstrakte Klassen | 3-11 |
| 3.5 | Klassenvariablen und Klassenmethoden | 3-12 |
| 3.6 | Abhängigkeiten | 3-15 |
| 3.6.1 | Partnerbeziehung | 3-15 |
| 3.6.2 | Der Dependency-Mechanismus | 3-17 |
| 4 | Komplexbeispiel - Verwaltung eines Warenlagers | 4-1 |
| 4.1 | Objektorientierter Entwurf | 4-1 |
| 4.2 | Bestimmen des Analyse-Kontextes (Phase 0) | 4-2 |
| 4.2.1 | Vorgehensweise | 4-2 |
| 4.2.2 | Komplexbeispiel | 4-2 |
| 4.3 | Detaillierte Beschreibung des Problems (Phase 1) | 4-4 |
| 4.3.1 | Vorgehensweise | 4-4 |
| 4.3.2 | Komplexbeispiel | 4-4 |
| 4.4 | Definition von Objekten (Phase 2) | 4-11 |
| 4.4.1 | Vorgehensweise | 4-11 |
| 4.4.2 | Komplexbeispiel | 4-11 |
| 4.5 | Objektklassifikation und Bestimmung von Beziehungen (Phase 3) | 4-14 |
| 4.5.1 | Vorgehensweise | 4-14 |
| 4.5.2 | Komplexbeispiel | 4-14 |
| 4.6 | Objektlebenszyklen und dynamische Aspekte des Programms (Phase 4) | 4-15 |
| 4.7 | Implementierung des Programms | 4-17 |
| 5 | Komplexbeispiel-Quellen | 5-1 |
| 6 | Was es sonst noch gibt | 6-1 |
| 6.1 | Die Klasse Collection | 6-1 |
| 6.1.1 | Set | 6-3 |
| 6.1.2 | SequenceableCollection | 6-3 |
| 6.1.3 | Dictionary | 6-6 |
| 6.2 | Die Klasse Stream | 6-7 |
| 6.2.1 | PositionableStream | 6-9 |
| 6.2.2 | EsRandom | 6-11 |
| 6.3 | Die Klasse CfsFileStream | 6-12 |
| 6.3.1 | CfsReadFileStream | 6-13 |
| 6.3.2 | CfsWriteFileStream | 6-15 |