

Generische Klassenerzeugung

Michael Jacobs

17. September 2009

Inhalt

Motivation

- Erstellung umfangreicher Klassenhierarchien
- Schablonenprinzip

Technologien

- XML
- XSLT

Werkzeuge

- XML-Editor
- XSLT-Prozessor

Informationsquellen

Klassenhierarchien: Ausgangssituation

- ▶ Hohe Anzahl an Klassen
- ▶ Aufbau der Unterklassen meist ähnlich
- ▶ Unterschied lediglich in verschiedenen Datenfeldern begründet

Klassenhierarchien: Probleme

- ▶ Bestimmte Teile aller Unterklassen sind immer gleich
- ▶ Hoher Tippaufwand bei Erstellung
- ▶ Einheitliche Änderungen an allen Klassen zeitintensiv
- ▶ Eigentlicher logischer Inhalt reiner Datenklassen oft minimal

Lösungsansatz

- ▶ Verlagerung gemeinsamer Methoden weitestmöglich in Oberklassen
- ▶ Beschreibung der Kind-Klassen als "logische Klassen"
- ▶ Dabei Beschränkung auf:
 - ▶ Logische Parameter
 - ▶ Typen der Parameter
- ▶ Einheitliche Schablone zur Generierung realer C++-Klassen aus den logischen Klassen

Abspeicherung logischer Daten in XML

- ▶ XML = eXtensible Markup Language
- ▶ Abspeicherung beliebiger Daten in definierten Markup-Formaten
- ▶ XML ist wegen seiner Verschachtelung besonders geeignet für Daten, die in Baumstrukturen abliegen
- ▶ Klassenhierarchien sind ebenfalls als Baumdiagramme darstellbar
- ▶ Logische Parameter von Klassenhierarchien lassen sich daher sehr leicht in einem XML-Baum darstellen

Generierung von Quelltext per XSLT

- ▶ XSLT = XML Stylesheet Language Transformations
- ▶ Einsatz zur definierten Umformung von XML-Dokumenten
- ▶ Kann außer XML wahlweise auch klassisches HTML oder Plaintext generieren
- ▶ Aus einem XML-Eingabedokument können viele Ausgabedokumente generiert werden
- ▶ XSLT-Stylesheet dient als Schablone zur Code-Generierung
- ▶ XSLT-Stylesheets werden selbst wieder in einem XML-Format verfasst

Verfassen von XML-Daten und XSLT-Stylesheets

- ▶ XML-Daten grundsätzlich mit beliebigen Plaintext-Editoren verfassbar
- ▶ Viele Editoren bieten farbliche Hervorhebung von XML-Dateien an:
 - ▶ Notepad++
 - ▶ Kate
- ▶ Einige IDEs bieten Unterstützung beim Erstellen von XSLT-Stylesheets an:
 - ▶ Microsoft Visual Studio
 - ▶ Stylus Studio

Transformation von XML per XSLT-Prozessor

- ▶ Eingabedaten:
 - ▶ XML-Datendokument
 - ▶ XSLT-Stylesheet
- ▶ Ausgabe der transformierten XML-Daten in Datei(en)
- ▶ Einige Implementierungen:
 - ▶ msxsl (XSLT 1.0)
 - ▶ xsltproc (XSLT 1.0, EXSLT)
 - ▶ Saxon (Java-basiert, XSLT 2.0)
- ▶ XSLT-1.0-Transformationen direkt ausführbar unter:
 - ▶ Java
 - ▶ .NET

Weitere Informationen zu den besprochenen Themen

- ▶ W3C
 - ▶ XML¹
 - ▶ XSLT 1.0²
 - ▶ XSLT 2.0³
 - ▶ XPath⁴

¹<http://www.w3.org/XML/>

²<http://www.w3.org/TR/xslt>

³<http://www.w3.org/TR/xslt20/>

⁴<http://www.w3.org/TR/xpath>

Weitere Informationen zu den besprochenen Themen

- ▶ Community
 - ▶ Libxml⁵
 - ▶ EXSLT⁶
 - ▶ Libxslt und xsltproc⁷
 - ▶ Saxon⁸

⁵<http://xmlsoft.org/>

⁶<http://www.exslt.org/>

⁷<http://xmlsoft.org/XSLT/>

⁸<http://saxon.sourceforge.net/>