Übungen zur Vorlesung
Spezifikation eingebetteter Systeme
WS 2010/2011

AG BS
Jan Peleska
Elena Vorobev

Übungsblatt 4

Revision: 1.2

In der Vorlesung wurde der in Abbildung 1 dargestellte Workflow zur Erzeugung von ausführbarem Code aus DSL-Modellen vorgestellt. Das Ziel dieses Übungsblattes ist es, den in dieser Abbildung grau markierten Teil umzusetzen.

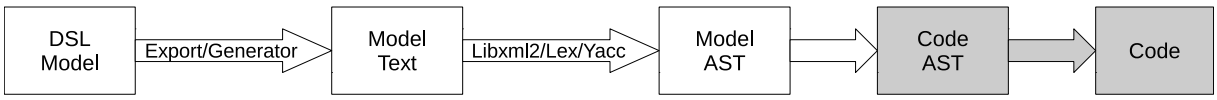

Abbildung 1: Workflow: vom DSL-Modell zum ausführbaren Code

Durch den Code-AST (im Folgenden: AST) werden Programme abstrakt - d.h. von der tatsächlichen Zielsprache unabhängig - repräsentiert. Um Code in einer bestimmen Zielsprache zu erzeugen, werden Backends implementiert, die einen gegebenen AST in die jeweilige Sprache übersetzen. Um das Hinzufügen neuer Backends bzw. anderer Transformationen ohne Anpassung der AST-Implementierung zu ermöglichen, kann das in der Vorlesung vorgestellte Visitor-Pattern verwendet werden.

Aufgabe 1: Implementieren Sie alle für die Repräsentation eines Programms als AST erforderlichen Klassen sowie ein Backend, durch welches AST-Instanzen in C-Code transformiert werden können[1]. Wenden Sie dabei das Visitor-Pattern an.

Die in der Vorlesung bereits erarbeiteten Teile dieser Aufgabe können von der SeS-Webseite heruntergeladen werden. `Ast.h` enthält dabei die Implementierung des AST, `AstVisitor.h` die der abstrakten AST-Visitor-Klasse und `CVisitor.h` die Implementierung des C-Backends. Die (unvollständige) Implementierung sollte geeignet ergänzt/verändert werden, wobei zusätzlich folgende Hinweise zu berücksichtigen sind:

- Deklarationen und Definitionen sollten getrennt werden (*.h- und *.cpp-Dateien).
- Alle Klassen sollten über geeignete Konstruktoren und Destruktoren verfügen.
- Es sollte auf die Kapselung interner Daten von Klassen geachtet werden.

Aufgabe 2: Testen Sie Ihre Implementierung, indem Sie AST-Instanzen erzeugen und diese unter Verwendung Ihres C-Backends in C-Code transformieren. Es sollen sowohl die einzelnen Sprachelemente des AST als auch geeignet gewählte Kombinationen aus diesen überprüft werden.

Abgabe: 11.01.2011 bis 16:00 Uhr

[1] Neben dem eigentlichen C-Code soll das Backend auch Zeilenumbrüche jedoch keine Tabulatoren (zur Code-Einrückung) ausgeben.