Inhaltsverzeichnis

1 Einführung / Ziele **3**

2 DDTDSL **4**
2.1 Objektbeschreibung . . . 4
2.1.1 Node . . . 4
2.1.2 Next . . . 4
2.1.3 Attribute . . . 4
2.1.4 Maybe . . . 4
2.1.5 Many . . . 4
2.2 Stringbeschreibung . . . 4
2.2.1 Key . . . 4
2.2.2 Value . . . 4
2.2.3 Variablen . . . 4
2.2.4 Maybe . . . 4
2.2.5 Many . . . 4
2.2.6 Or . . . 4

3 Verwendete Technologien **4**
3.1 Xtext . . . 4
3.2 Xtend . . . 4
3.3 Java Reflection . . . 4

4 Programming - Compilergenerator **4**
4.1 Was ist ein Compilergenerator? . . . 4
4.2 Fehlerbehandlung . . . 4
4.3 SimpleScanner . . . 4
4.4 Objekte . . . 4
4.4.1 Node . . . 4
4.4.2 Next . . . 4
4.4.3 Attribute . . . 4
4.4.4 Maybe . . . 4
4.4.5 Many . . . 4
4.4.6 Parserkreise . . . 4
4.5 Strings . . . 4
4.5.1 Key . . . 4
4.5.2 Value . . . 4
4.5.3 Variablen . . . 4
4.5.4 Maybe . . . 4
4.5.5 Many . . . 4

4.5.6 Or 4

5 Testing **4**
5.1 Validation - Generator Syntax 4
5.2 JUnit - Generator Model 4
5.3 JUnit - Compiler 4

6 Anwedungsbeispiel **4**

7 Fazit **4**

8 Anhang **4**

1 Einführung / Ziele

Listing 1: Erster Test

2 DDTDSL

2.1 Objektbeschreibung

2.1.1 Node

2.1.2 Next

2.1.3 Attribute

2.1.4 Maybe

2.1.5 Many

2.2 Stringbeschreibung

2.2.1 Key

2.2.2 Value

2.2.3 Variablen

2.2.4 Maybe

2.2.5 Many

2.2.6 Or

3 Verwendete Technologien

3.1 Xtext

3.2 Xtend

3.3 Java Reflection

4 Programming - Compilergenerator

4.1 Was ist ein Compilergenerator?

4.2 Fehlerbehandlung

4.3 SimpleScanner

4.4 Objekte

4.4.1 Node

4.4.2 Next

4.4.3 Attribute

4.4.4 Maybe

4.4.5 Many

4.4.6 Parserkreise

4.5 Strings