

методическое пособие

ОБРАБОТКА ТЕКСТОВЫХ ФОРМАТОВ ДАННЫХ И РЕАЛИЗАЦИЯ КОМПЬЮТЕРНЫХ ЯЗЫКОВ на Flex/Bison/C++/LLVM Java/C#/ANTLR

GitHub: <https://github.com/ponyatov/lexman>

© <dponyatov@gmail.com>

2 января 2016 г.

Оглавление

1	Java, Android, C#, ANTLR	2
1.1	Инсталяция	3
1.1.1	Java JDK	3
1.1.2	ANTLR	4
1.1.3	Android Studio	5
1.2	Необходимый инструментарий	5
1.3	Разработка грамматики	5
	Литература	8
	Оснoвы компиляторов	8
	LLVM	9
	Java/ANTLR	9
	Утилиты	9
	Л ^A T _E X: система верстки для научных публикаций	9

Глава 1

Java, Android, C#, ANTLR

В современных IT-приложениях не всегда возможно использовать язык C^{++} и динамическую компиляцию. Для случаев, когда вы вынуждены использовать среды, заточенные под Java или .NET/C#, можно использовать генератор парсеров ANTLR. В этом разделе будет рассмотрен простая языковая среда-калькулятор, реализованная в нескольких вариантах, в т.ч. как приложение Android.

¹

Когда встает вопрос о том, как в программе на Java или C# разобрать и вычислить арифметическое выражение, заданное в виде привычного инфиксного выражения, можно составить грамматику входного языка, и построить по ней лексический и синтаксический анализатор при помощи ANTLR.

[AN]other [T]ool for [L]anguage [R]ecognition) — мощный генератор парсеров для чтения, обработки, выполнения и трансляции структурированных текстовых или бинарных файлов. ANTLR широко использу-

ется для построения языковых утилит или фреймворков. На основе грамматики ANTLR генерирует парсер, который может строить и обходить деревья разбора.

ANTLR — генератор парсеров, позволяющий автоматически создавать программу-парсер (как и лексический анализатор) на одном из целевых языков программирования (C^{++} , Java, $C\sharp$, Python, Ruby) по описанию LL(*)-грамматики на языке, близком к РБНФ.

1.1 Инсталляция

1.1.1 Java JDK

Для самостоятельного использования платформы JavaSE достаточно 32-битной версии, для разработки под Android современная среда разработки Android Studio 1.1.3 требует исключительно 64-битную версию JDK.

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Скачайте и установите Java JDK x32/x64:

Java Platform, Standard Edition >> JDK >> Download

Java SE Development Kit 8u65 >> Accept License agreement

Windows x86 >> jdk-8u65-windows-i586.exe

Windows x64 >> **jdk-8u65-windows-x64.exe**

После установки JDK нужно прописать несколько переменных окружения для пользователя:

Компьютер >> Свойства >> Доп.параметры системы >> Переменные среды

```
JAVA_HOME = C:\Java\jdk64
PATH = %JAVA_HOME%\bin;%PATH%
```

Переменная JAVA_HOME необходима для правильной установки Android Studio 1.1.3.

1.1.2 ANTLR

▣Windows

```
cd \Java ; mkdir ANTLR ; cd ANTLR
wget -c http://www.antlr.org/download/antlr-4.5.1-complete.jar
wget -c
```

Установите ANTLRWorks, распаковав .zip в C:\Java\ANTLR\.

Если вы работаете в ☹ECLIPSE, плагин Edgar Espina доступен для установки через ☹ECLIPSE Marketplace под именем ANTLR 4 IDE в комплекте с самой библиотекой ANTLR, устанавливаемой как плагин.

Установите переменные окружения для пользователя CLASSPATH² и добавьте в PATH:

```
CLASSPATH = .;C:\Java\ANTLR\antlr-4.5.1-complete.jar;%CLASSPATH%
PATH = C:\Java\ANTLR;%PATH%
```

Создайте в \Java\jdk\bin\ пару батников для запуска утилит:

antlr4.bat

```
1 java org.antlr.v4.Tool %*
```

² не забудьте про точку .;

```
1 java org.antlr.v4.gui.TestRig %*
```

1.1.3 Android Studio

<http://developer.android.com/intl/ru/sdk/index.html>

All Android Studio Packages >> Windows >> android-studio-bundle-xxx.xxx-windows.exe

Bundle включает не только IDE на базе IntelliJJava, но и Android SDK.

1.2 Необходимый инструментарий

- **ANTLRWorks** — среда для разработки и отладки грамматик
- **ANTLR** — собственно сам ANTLR
- **Antlr4.Runtime.dll** C# runtime distribution – .NET библиотека для работы со сгенерированными анализаторами, входит в пакет **antlr-csharp-runtime-4.x.x.zip**

1.3 Разработка грамматики

Настало время запустить ANTLRWorks и создать проект с грамматикой калькулятора:

File >> New File >> ANTLR >> ANTLR 4 Combined Grammar >> Next

Folder >> D:/w/lexman/Jcalc

File Name >> ANTLRgrammar

Finish

ANTLRgrammar.g4

```
1 grammar ANTLRgrammar;
```

Тип грамматики Combined Grammar означает, что в одном файле у нас будут находиться одновременно и правила лексера, и синтаксического анализатора.

Давайте разберем, как записываются правила для лексера на примере правила: целого числа INT.

Правило начинается с имени, **для правил лексера** обязаны начинаться **с заглавной буквы**. После имени идет символ `:`, за которым следуют **альтернативы**. Альтернативы разделяются символом `|` и должны заканчиваться символом `;`.

Посмотрим на единственную альтернативу для правила INT. Запись `'0'..'9'+` в ANTLR означает, что INT — непустая последовательность цифр. Символ `*` допускает пустую последовательность.

ANTLRgrammar.g4

```
1 INT : ('0'..'9')+ ;
```

Правило ID несколько сложнее и означает, что ID — это последовательность символов, состоящая из строчных и заглавных латинских букв, цифр, символов подчеркивания и начинающаяся с буквы или символа подчеркивания.

ANTLRgrammar.g4

```
1 ID : [a-zA-Z_][a-zA-Z_0-9]* ;
```

Для работы смешанной грамматики необходимо минимум одно правило-нетерминал: первое правило для парсера или по-научному **аксиому**, то есть правило, с которого парсер начнет проверять поток лексем:

ANTLRgrammar.g4

```
1 calc : expr+ ;  
2 expr : scalar ;  
3 scalar : INT | ID ;
```

В данном случае мы говорим парсеру, что входной поток лексем представляет из себя непустую (+) последовательность `expr`ов, в одном из вариантов включающий **скалярные типы** `INT` и `ID`.

Литература

Основы компиляторов

- [1] [Книга Дракона \(Dragon Book\)](#):
Альфред Ахо, Моника С. Лам, Рави Сети, Джеффри Ульман
Компиляторы: принципы, технологии и инструментарий
- [2] Habrahabr: [Компиляция. 1: лексер](#)
- [3] Habrahabr: [Компиляция. 2: грамматики](#)
- [4] Википедия [Регулярные выражения](#)
- [5] OpenNet.ru: [Генератор лексических анализаторов lex](#)
- [6] [Converting from old lex & yacc to flex & bison](#)

LLVM

- [7] [LLVM tutorial](#)

Java/ANTLR

- [8] [Грамматика арифметики или пишем калькулятор на ANTLR](#)

Утилиты

- [9] Emden Gansner and Eleftherios Koutsofios and Stephen North
[Drawing graphs with dot](#)

Л^AT_EX: система верстки для научных публикаций

- [10] Википедия: [система верстки Л^AT_EX](#)

- [11] Котельников И. А., Чеботаев П. З.

Л^AT_EXпо-русски.

— СПб.: «Корона-Век», 2011. — 496 с. — 2000 экз. — ISBN 978-5-7931-0878-2.

[12] Львовский С. М.

Набор и верстка в системе LaTeX.

— М.: МЦНМО, 2006. — С. 448. — ISBN 5-94057-091-7.

[13] Балдин Е. М.

Компьютерная типография LaTeX.

— «БХВ-Петербург», 2008. — 304 с. — 2000 экз. — ISBN 978-5-9775-0230-6. Книга доступна в электронном виде на сайте CTAN под лицензией CC-BY-SA.

[14] Столяров А. В.

Сверстай диплом красиво: LaTeX за три дня.

— Москва: МАКС Пресс, 2010. — 100 с. — 200 экз. — ISBN 978-5-317-03440-5.