

# Язык *bI*

чтобы никто не догадался

© Dmitry Ponyatov <[dponyatov@gmail.com](mailto:dponyatov@gmail.com)>

27 августа 2015 г.

# Оглавление

Язык <i>bI</i>	2
Литература	3
Файлы проекта	4
<b>I Синтаксис языкового ядра</b>	<b>5</b>
Комментарии	6
Точечные команды	7
<b>II Система типов</b>	<b>8</b>
<b>1 Нативные</b>	<b>9</b>
1.1 c: одиночный символ <i>/[c]har/</i>	10
1.2 s: строка <i>/[s]tring/</i>	10
1.3 i: целое <i>/[i]nt/</i>	10
1.4 f: число с плавающей точкой <i>/[f]loat/</i>	10

III    **Расширение**    ☉ *ECLIPSE*

Манифест . . . . . 12

Перспектива *bI* . . . . . 13

Язык *bI*

- DSL-ориентированный
  - динамический
  - объектный
  - параллельный
- язык программирования и проектирования программно-аппаратных систем.

Язык *bI* был создан в экспериментах с синтаксическим анализом, символьными вычислениями, и чтении книг типа [1, 2, 3]. Движущей силой было желание создать **инструмент для метапрограммирования: описания прикладных программ и программно-аппаратных систем в свободном синтаксисе**, с применением смешанных методов программирования, и без необходимости полностью описывать все объекты программы во всех тонкостях и с соблюдением всех заморочек языка реализации, библиотек, платформ и ОС.

Идея о свободном синтаксисе предполагала, что программист имеет полный контроль над языком, так как сам написал его реализацию, и по мере необходимости меняет работу существующего функционала языковой системы, и дописывает дополнительные фичи. В процессе работы над различными проектами в итоге формируется некий клон базового языка, заточенный под решаемые задачи, под используемые платформы, и с определенным набором средств автоматизации рутинных вопросов, типа генерации кода на Си, автоматического отслеживания зависимостей между файлами и т.п.

# Литература

- [1] SICP:  
*Харольд Абельсон, Джеральд Джей Сассман, Джули Сассман*  
**Структура и интерпретация компьютерных программ**
- [2] Книга Дракона  
*А.Ахо, Р.Сети, Д.Ульман*  
**Компиляторы: принципы, технологии, инструменты**
- [3] *Ю.А.Кирютенко, В.А.Савельев*  
**Объектно-ориентированное программирование Язык SmallTalk**
- [4] *Келли М., Спайс Н.*  
**Язык программирования Форт**  
/ Пер. с англ. — М.: Радио и связь, 1993. — 320 с. — ISBN 5-256-00438-7
- [5] *Баранов С. Н., Ноздрунов Н. Р.*  
**Язык Форт и его реализации**  
— Л.: Машиностроение, 1988. — 157 с. — (ЭВМ в производстве). — 100 000 экз. — ISBN 5-217-00324-3.

# Файлы проекта

Makefile	файл зависимостей
lexer.lpp	лексический анализатор $bI_0$ на языке flex
parser.ypp	синтаксический парсер $bI_0$ на языке bison
core.cpp	runtime-ядро $bI_0$ на $C_+^+$
bI.hpp	определения встроенных типов $bI_0$
bI.bI	описание модели языка $bI$ следующей версии
doc/	документация в формате $\text{\LaTeX}$
eclipse/ META-INF/ plugin.xml build.properties	расширение для $\text{\textcircled{E}}ECLIPSE$

# Часть I

## Синтаксис языкового ядра

*bI* построен на bypass-принципе: все, что языковая система получила на вход, и не распознала как элемент языка, отправляется на выход без изменений.

Также используется литературное программирование<sup>1</sup>: решение задачи описывается на удобном языке разметки ( $\text{\LaTeX}$ , DocBook,...), а *bI* **вычленяет и выполняет блоки кода**. Поэтому комментарии в программах на *bI* **не используются**.

Для упрощения отладки и разработки "ядерная" реализация языка *bI*<sub>0</sub> выдает на stdout html-заголовок, позволяет использовать в исходном тексте html-тэги, и при вычислении выражений результат также обрмляется в стилевые тэги в соответствии с типом полученного значения.

Использование на начальном этапе разработки языка такой "обрезанной" html-разметки вместо полноценного документирования в формате  $\text{\LaTeX}$  позволяет исключить использование относительно медленного  $\text{\TeX}$ -процессора для просмотра результата: используется быстрый минибраузер, встроенный в IDE  $\ominus ECLIPSE$ . Кроме того, этот подход оказывается удобным и для встраиваемых систем с веб-интерфейсом.

## Комментарии

Тем не менее из условий запуска скриптов в UNIX требуется, чтобы первая строка имела формат `#!/bin/bI`, поэтому в язык все же вводятся лексемы комментариев:

```
#!/bin/bI
```

---

<sup>1</sup> literate programming

# строчный комментарий  
#| блочный  
комментарий |#

## Точечные команды

Точечные dot-команды используются для документирования и управления форматами вывода  $bI$ -системы<sup>2</sup>.

.html	режим html-форматирования, для $bI_0$ единственный поддерживаемый режим
.tex	$\text{\LaTeX}$ -разметка
.sec	секция документации
.sec- .sec+	изменение текущего уровня секционирования (глава, раздел,...)

---

<sup>2</sup> точка была взята из языка *Forth*, подразумевает конструкции языка, относящиеся к выводу



Часть II

Система типов

# Глава 1

## Нативные

Нативные типы реализуются на уровне  $bI_{VM}^1$ , являются типами самого низкого уровня, и не покрываются в полной мере средствами отладки, трассировки, и контроля. При кросс-трансляции преобразуются в базовые типы целевого языка, и элементы данных с аппаратной поддержкой<sup>2</sup>.

Особенно аккуратно их нужно использовать при трансляции для 8/16-битных платформ, так как низкая разрядность нативных типов и облегченный контроль со стороны  $bI$ -машины дает весь спектр побочных эффектов: переполнения, знаковые ошибки, потерю точности и т.п.

---

<sup>1</sup> виртуальная машина, рантайм-движок языковой системы  $bI$

<sup>2</sup> машинное слово, блок памяти, элемент стека мат.сопроцессора и т.п.

1.1 **c**: одиночный символ /**[c]**har/

1.2 **s**: строка /**[s]**tring/

1.3 **i**: целое /**[i]**nt/

1.4 **f**: число с плавающей точкой /**[f]**loat/

# Часть III

Расширение  $\oplus$  *ECLIPSE*

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <?eclipse version="3.4"?>
3 <plugin>
4 <extension point="org.eclipse.ui.perspectives">
5 <extension point="org.eclipse.ui.newWizards">
6 <extension point="org.eclipse.ui.commands">
7 <extension point="org.eclipse.ui.menus">
8 </plugin>
```

## Манифест

```
1 Manifest-Version: 1.0
2 Bundle-ManifestVersion: 2
3 Bundle-Name: bl
4 Bundle-SymbolicName: org.eclipse.bl; singleton:=true
5 Bundle-Version: 0.0.1.alpha
6 Bundle-Activator: org.eclipse.bl.Activator
7 Bundle-Vendor: Dmitry Ponyatov <dponyatov@gmail.com>
8 Require-Bundle: org.eclipse.core.runtime,
9   org.eclipse.core.resources,
10  org.eclipse.ui,
11  org.eclipse.ui.console
12 Bundle-RequiredExecutionEnvironment: JavaSE-1.7,
13   JavaSE-1.8
```

## Перспектива *bl*

org.eclipse.ui.perspectives

```
1 <extension point="org.eclipse.ui.perspectives">
2     <perspective
3         class="org.eclipse.bl.Perspective"
4         id="org.eclipse.bl.perspective"
5         icon="META-INF/icon_16x16.png"
6         name="bl">
7     </perspective>
8 </extension>
```

org.eclipse.bl.Perspective

```
1 package org.eclipse.bl;
2
3 import org.eclipse.ui.IFolderLayout;
4 import org.eclipse.ui.IPageLayout;
5 import org.eclipse.ui.IPerspectiveFactory;
6 import org.eclipse.ui.console.IConsoleConstants;
7
8 public class Perspective implements IPerspectiveFactory {
9
10     public static final String ID = "org.eclipse.bl.perspective";
11 }
```

```
12 @Override
13 public void createInitialLayout(IPageLayout layout) {
14     defineActions(layout);
15     defineLayout(layout);
16 }
17
18 private void defineActions(IPageLayout layout) {
19     // new menu items
20     layout.addNewWizardShortcut("org.eclipse.ui.wizards.new.file");
21     layout.addNewWizardShortcut("org.eclipse.ui.wizards.new.folder");
22     layout.addNewWizardShortcut("org.eclipse.bl.newproject");
23     layout.addNewWizardShortcut("org.eclipse.bl.newtarget");
24     // views
25     layout.addShowViewShortcut(IPageLayout.ID_PROJECT_EXPLORER);
26     layout.addShowViewShortcut(IConsoleConstants.ID_CONSOLE_VIEW);
27 }
28
29 private void defineLayout(IPageLayout layout) {
30     String editorArea = layout.getEditorArea();
31     IFolderLayout left = layout.createFolder("left", IPageLayout.LEFT, (float) 0.22);
32     IFolderLayout bottom = layout.createFolder("bottom", IPageLayout.BOTTOM, (float) 0.78);
33     left.addView(IPageLayout.ID_PROJECT_EXPLORER);
34     bottom.addView(IConsoleConstants.ID_CONSOLE_VIEW);
35 }
36
37 }
```