Язык bI

© Dmitry Ponyatov <dponyatov@gmail.com>

17 августа 2015 г.

Оглавление

Язык *bI*

- динамический
  в *LISP*-смысле: программа сама является структурой данных, и может модифицировать другие программы или сама себя, создавать и удалять программы в процессе выполнения.
- DSL-ориентированный
  так как очень часто приходиться работать с данными в текстовых форматах, в ядре языка предусмотрен функционал создания парсеров, оптимизаторов, трансляторов и компиляторов любых других языков.
- объектный
  в стиле *SmallTalk*.

- параллельный
  при разработке языка большое внимание уделяется обеспечению параллелизма в самом широком смысле: микропотоки, использование потоков runtime-системы и/или ОС, управление выполнением программ на гетерогенных вычислительных кластерах, облачных севисах и p2p распределенных сетях, средства платформенно-независимой сериализаци, поддержка персистентности и резервирования, синтаксическая поддержка параллельного программирования.

- run-time спецификация
  объектов в процессе выполнения программы не требует обязательное предварительное определение объектов, при попытке использования несуществующего объекта открывается интерактивная отладочная сессия.

- интерактивная отладка
  в стиле SmallTalk позволяет программисту создавать программу в диалоговом режиме в процессе отладки и прогона на тестовом стенде или наборе юнит-тестов.

- компилируемый через трансляцию
  результирующая система может быть оттранслирована[1] в исходный код на C_{+}^{+}[2] для обеспечения переносимости программ для систем, для которых не реализована bI-машина, недостаточно аппаратных ресурсов[3], или предъявляются жесткие требования по надежности[4].

- литературное программирование[5] и автоматическая генерация документации

---

[1] ограниченно
[2] и $Python$
[3] встраиваемые системы
[4] ↑
[5] literate programming

Часть I

Синтаксис

Глава 1

Комментарии

строчный

```
// line comment //
```

блочный

```
\\ block
        comment \\
```

разметка для документации

Глава 2

Литералы

числа

строки

Глава 3

Контейнеры

пары

списки

словари

Часть II

Реализация

Глава 4

Ядро на C_+^+

4.1 Файлы

Makefile файл проекта

4.2 lexer.lpp: лексер

4.3 parser.ypp: парсер

4.4 core.cpp: runtime ядро

4.5 bl.h: декларации типов

Часть III

Система типов

Глава 5

Базовые

5.1 object: объект

5.2 int: целое число

5.3 float: число с плавающей точкой

5.4 string: строка

Глава 6

Контейнеры

6.1 pair: пара

6.2 list: список

6.3 dict: словарь

6.4 class: класс

Глава 7

Файлы

7.1 file: бинарный файл /binary/

7.2 text: текстовый файл /plain text/

7.3 lexer: лексер

7.4 parser: парсер синтаксиса

7.5 data: синтаксически размеченный файл с данными

Глава 8

Многозадачность

Глава 9

GUI

Глава 10

Сетевой функционал

Глава 11

Мультимедиа

Глава 12

Математические

Глава 13

Геометрия и САПР