

Refal Plus Development Tools



Юрий Климов

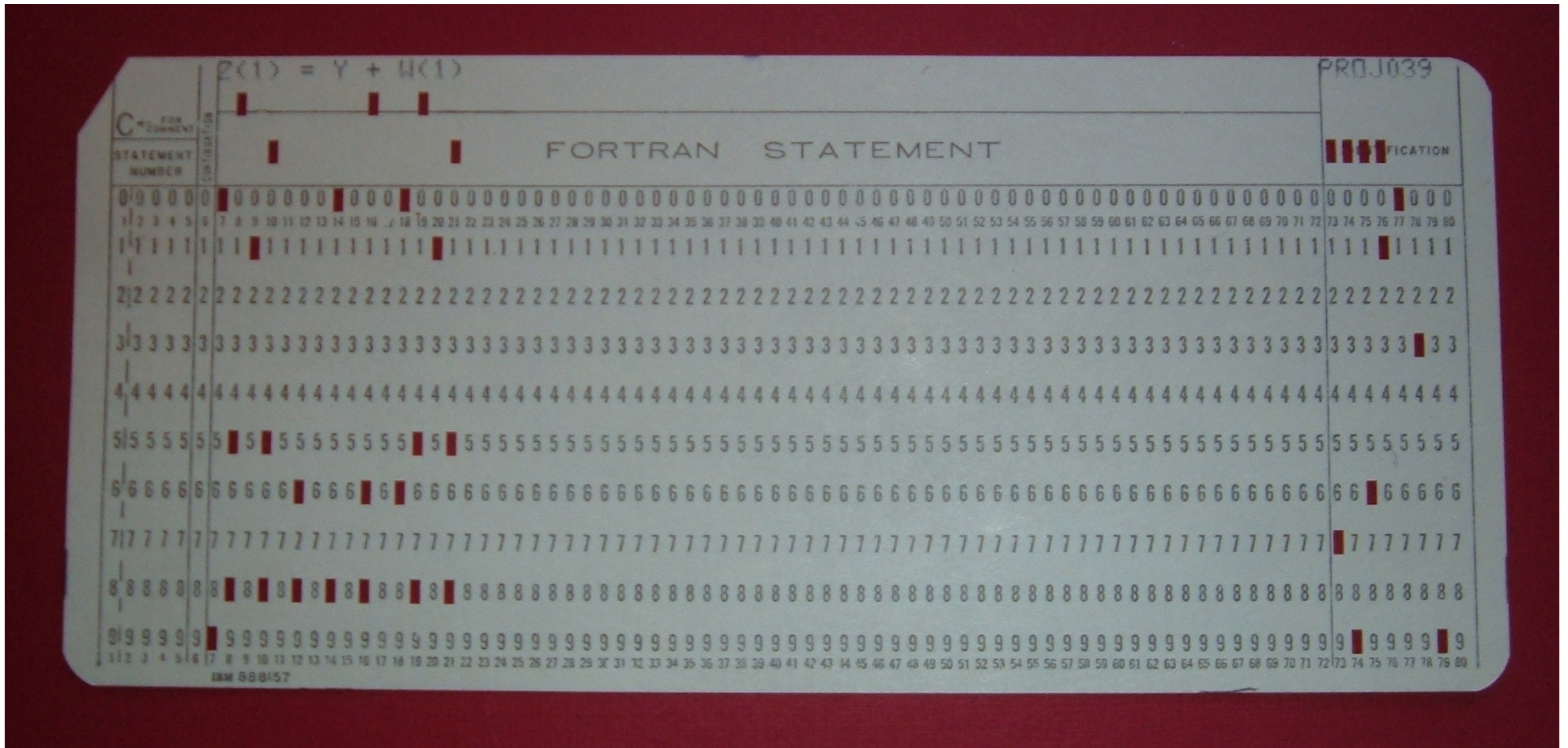
<http://pat.keldysh.ru/~yura/>



Антон Орлов

<http://pat.keldysh.ru/~orlov/>

Real Programmers Don't Use IDEs!





*Я вспомнил нашу БЭСМ. Панель управления
цвета заварного крема...*

«Понедельник начинается в субботу»
А. и Б. Стругацкие

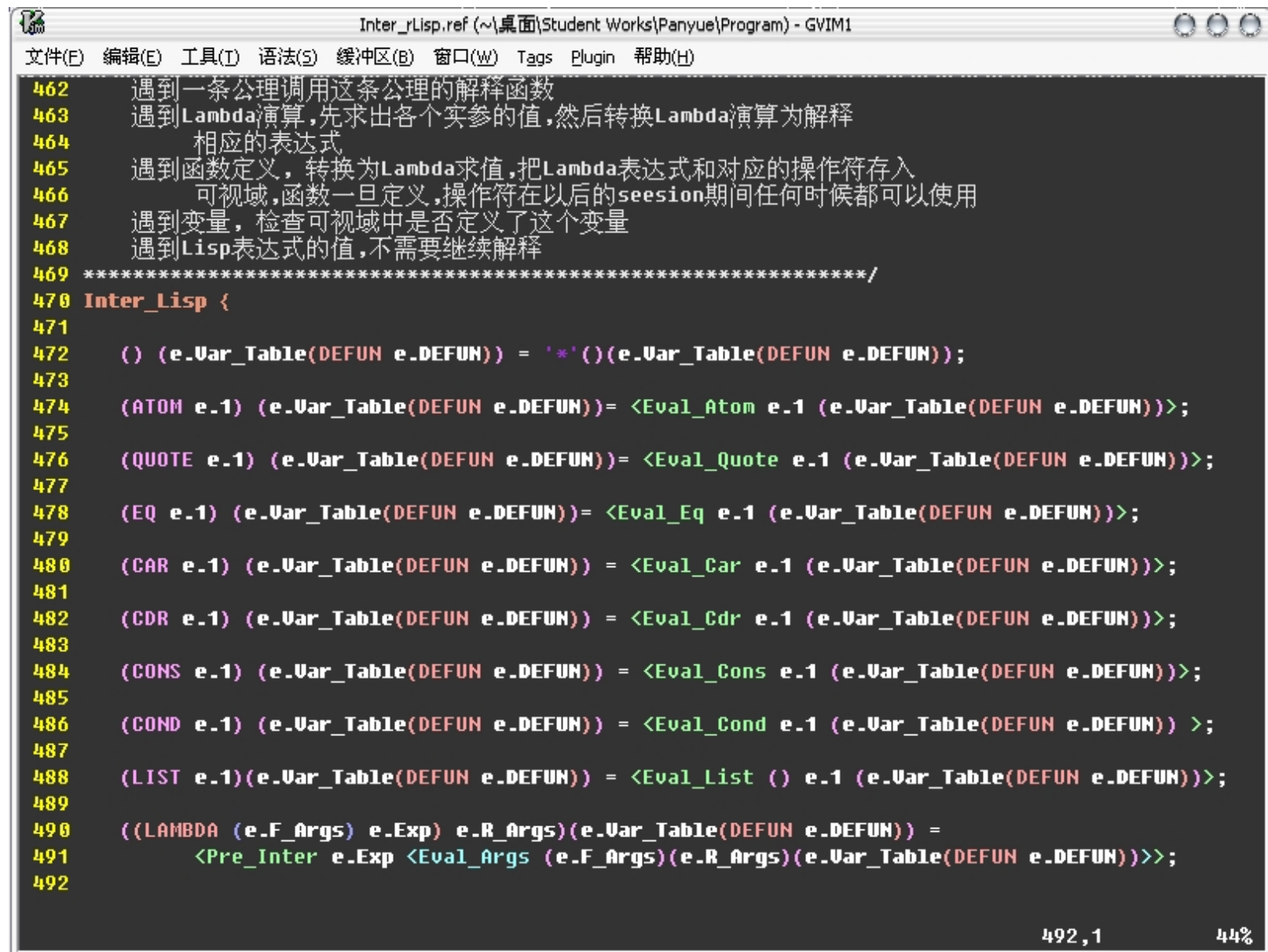
Real Programmers Use Command Prompt Only!

Not even an editor!

```
> echo Репка | sed "s/???$/`echo Дедка Бабка |  
sed 's/^???//'`/" | sed "s/пк/`echo Внучка |  
sed 's/B.*/ф/'`/" | sed "s/$/`echo Жучка Кошка  
Мышка | sed s/.*/л/'`/"
```

Рефал

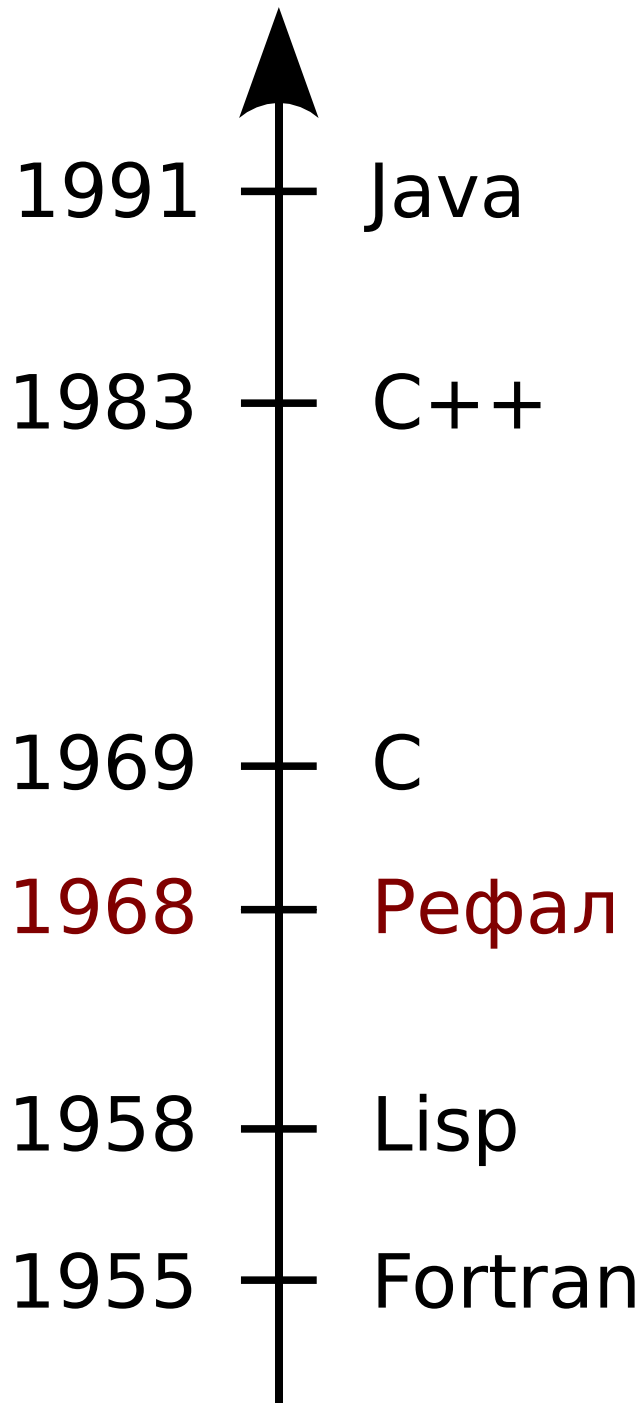
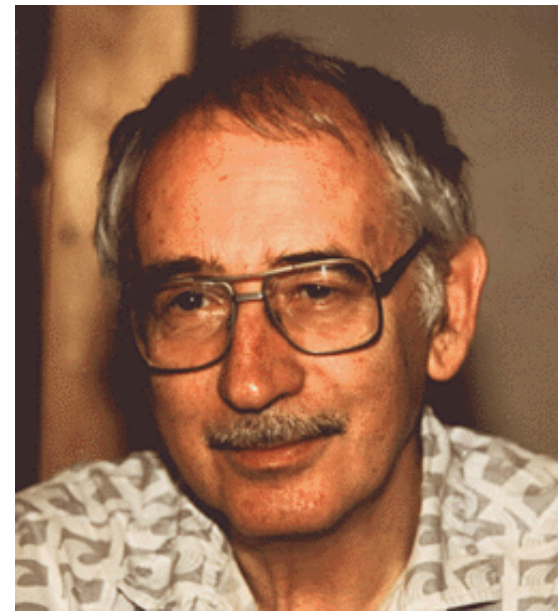
Refal Programmers Don't Use IDEs?



```
462 遇到一条公理调用这条公理的解释函数
463 遇到Lambda演算,先求出各个实参的值,然后转换Lambda演算为解释
464 相应的表达式
465 遇到函数定义,转换为Lambda求值,把Lambda表达式和对应的操作符存入
466 可视域,函数一旦定义,操作符在以后的session期间任何时候都可以使用
467 遇到变量,检查可视域中是否定义了这个变量
468 遇到Lisp表达式的值,不需要继续解释
469 *****/
470 Inter_Lisp {
471
472  () (e.Var_Table(DEFUN e.DEFUN)) = '*'()(e.Var_Table(DEFUN e.DEFUN));
473
474  (ATOM e.1) (e.Var_Table(DEFUN e.DEFUN))= <Eval_Atom e.1 (e.Var_Table(DEFUN e.DEFUN))>;
475
476  (QUOTE e.1) (e.Var_Table(DEFUN e.DEFUN))= <Eval_Quote e.1 (e.Var_Table(DEFUN e.DEFUN))>;
477
478  (EQ e.1) (e.Var_Table(DEFUN e.DEFUN))= <Eval_Eq e.1 (e.Var_Table(DEFUN e.DEFUN))>;
479
480  (CAR e.1) (e.Var_Table(DEFUN e.DEFUN)) = <Eval_Car e.1 (e.Var_Table(DEFUN e.DEFUN))>;
481
482  (CDR e.1) (e.Var_Table(DEFUN e.DEFUN)) = <Eval_Cdr e.1 (e.Var_Table(DEFUN e.DEFUN))>;
483
484  (CONS e.1) (e.Var_Table(DEFUN e.DEFUN)) = <Eval_Cons e.1 (e.Var_Table(DEFUN e.DEFUN))>;
485
486  (COND e.1) (e.Var_Table(DEFUN e.DEFUN)) = <Eval_Cond e.1 (e.Var_Table(DEFUN e.DEFUN)) >;
487
488  (LIST e.1)(e.Var_Table(DEFUN e.DEFUN)) = <Eval_List () e.1 (e.Var_Table(DEFUN e.DEFUN))>;
489
490  ((LAMBDA (e.F_Args) e.Exp) e.R_Args)(e.Var_Table(DEFUN e.DEFUN)) =
491  <Pre_Inter e.Exp <Eval_Args (e.F_Args)(e.R_Args)(e.Var_Table(DEFUN e.DEFUN))>>;
492
```

Рефал —

РЕкурсивный
Функциональный
Алгоритмический
язык



Автор: В. Ф. Турчин

Диалекты:

- Рефал-2 – 1986
- Рефал-5 – 1989, 2000
- Рефал-6 – 1992, 2004
- Рефал+ – 1991, 2006

Рефал

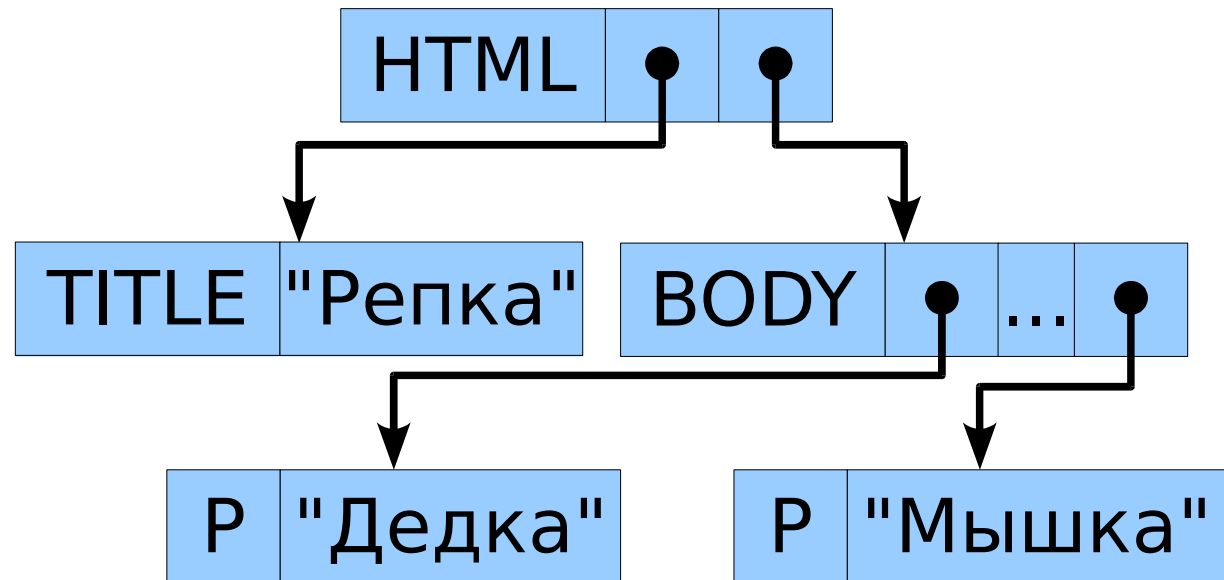
- Функциональный язык («чистые» функции)
- Уникальная основная структура данных — рефал-выражение
- Удобен для задач обработки символьной информации
- Хорошая интеграция с языками C/C++/Java
- Современная среда разработки (IDE)

Рефал-выражение

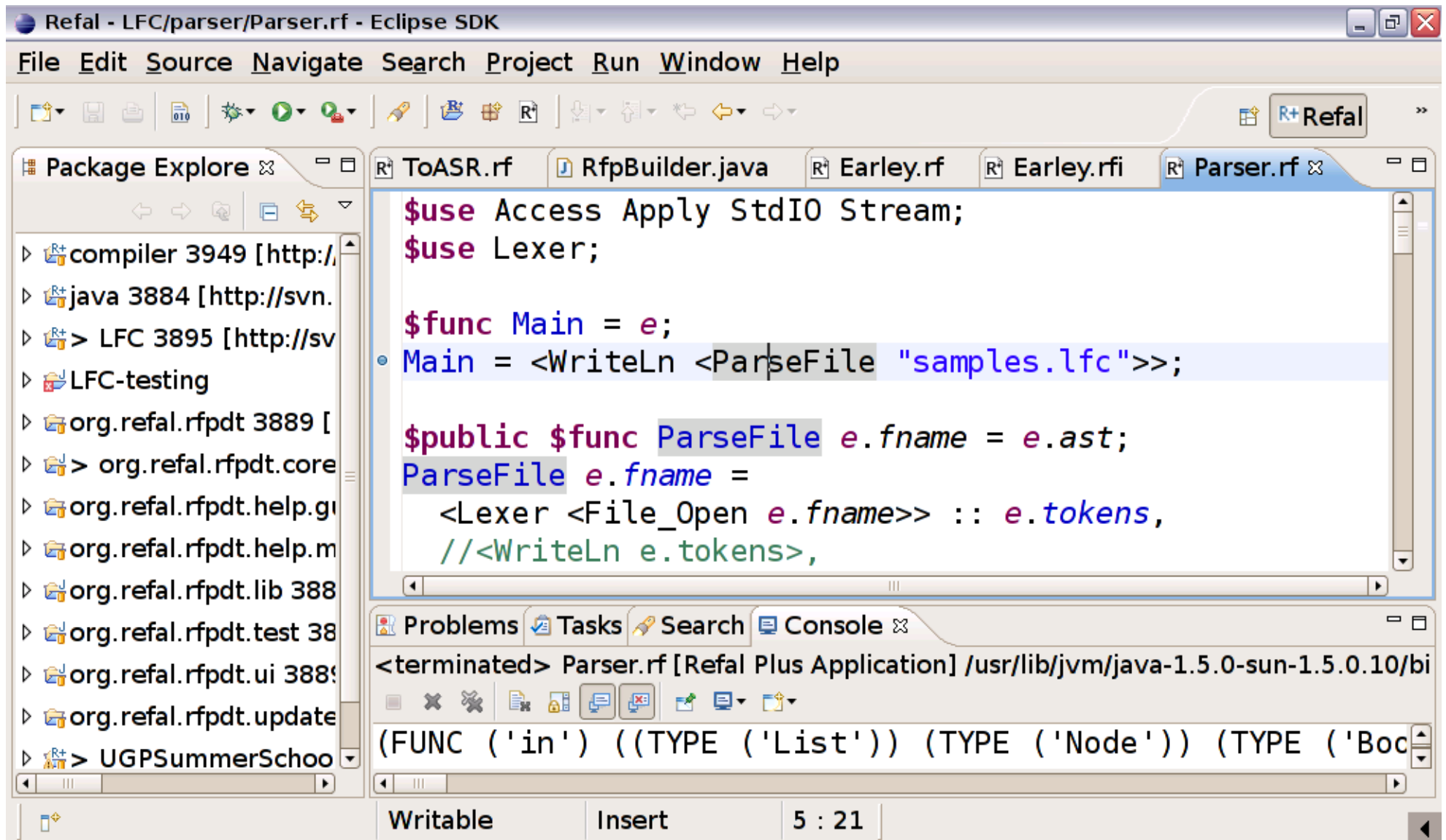
- Последовательность символов с правильно расставленными скобками
 - Выражение — последовательность термов
 - Терм — символ или выражение в скобках

- Пример:

```
(HTML  
  (TITLE "Репка")  
  (BODY  
    (P "Дедка")  
    ...  
    (P "Мышка")  
  )  
)
```



Среда разработки



Пример рефальской задачи

Даны две ASCII-строки:

- одна состоит только из латинских букв – это **слово**
 - в другой могут встречаться большие латинские буквы и еще два спецсимвола: * и ? – это **шаблон**
- ? и * – wildcards, как в именах файлов в DOS или Unix-shell:
- ? заменяет собой ровно один произвольный символ
 - * заменяет собой любое количество произвольных символов

Программа должна выдать ответ **YES**, если слово удовлетворяет шаблону, либо **NO** в противном случае.

Кодометрия

Решение задачи потребовало:

- Pascal – 1.5 часа, 116 строк
- Java – 3 часа, 270 строк
- Рефал – 20 минут, 25 строк

<http://wiki.botik.ru/Refaldevel/ForJavaProgrammer>

```
$func? Match e.word (e.pattern) = ;
```

```
Match \{  
  t1 e2      (t1 e3)  = <Match e2 (e3)>;  
  t1 e2      ('?' e3) = <Match e2 (e3)>;  
  e1         ('*' e3) = { e1 : e11 e12, <Match e12 (e3)>; };  
  /*empty*/  ()      = /*yes*/  
};
```

Match принимает на вход **слово** и **шаблон**, в котором '?' заменяет собой ровно один произвольный символ, а '*' заменяет собой любое количество произвольных символов - 0 или более.

Завершает работу успешно, если слово удовлетворяет шаблону.

Задача о цепочках

Найдите цепочку из трех единичек, трех двоек, ..., трех девяток такую, что для каждой цифры **n** между последовательными вхождениями **n** в эту цепочку находится ровно **n** чисел.

Пример такой последовательности:

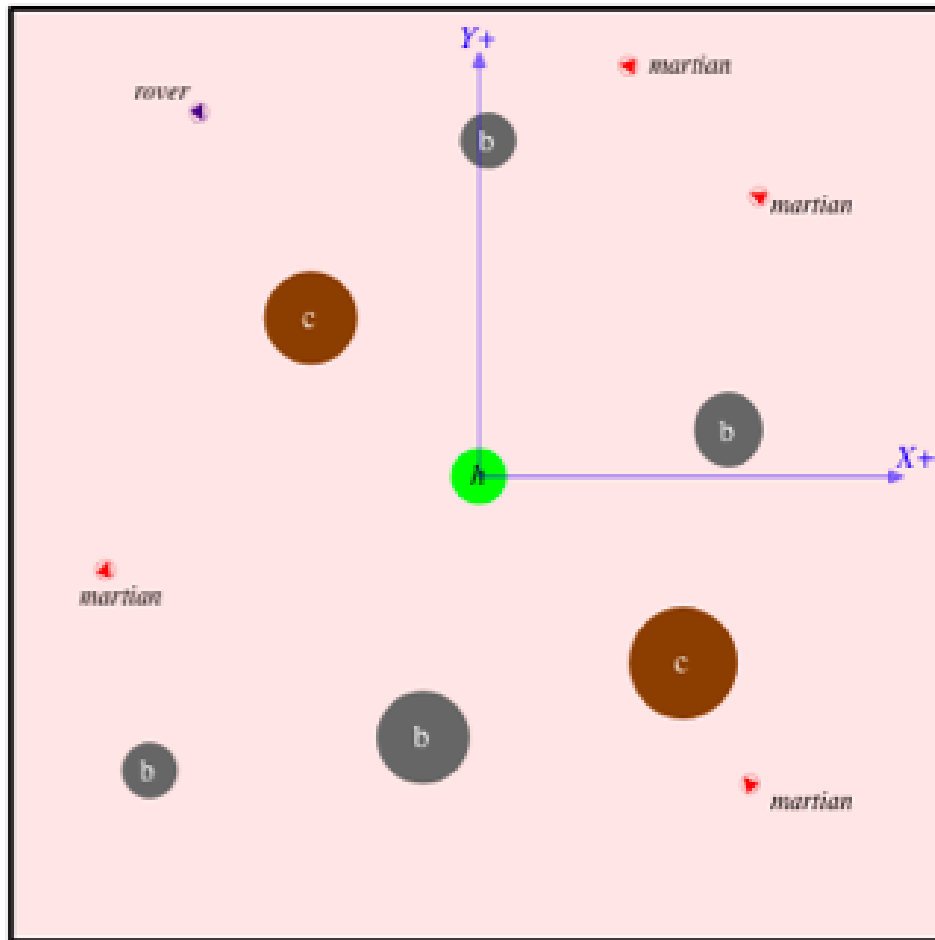
1, 9, 1, 2, 1, 8, 2, 4, 6, 2, 7, 9, 4, 5, 8, 6, 3,
4, 7, 5, 3, 9, 6, 8, 3, 5, 7

<http://icfpcontest.org>

Language Frequency 2008

- Java 56
- Python 54
- C++ 54
- Haskell 36
- ML-family 31
- Ruby 16
- C 13
- Erlang 10
- C# 8
- Lisp 7
- Perl 6
- Scheme 4
- Scala 4
- Pascal 2
- ATS/C 1
- D 1
- F# 1
- JavaScript 1
- Slag 1
- Rhope 1
- Clojure 1
- Mathematica 1
- Dylan 1
- Disciple 1
- LaTeX 1

The World



Bounded 2D plane

Static obstacles:

- boulders: bounce off if you hit one

- craters: die if you fall in

Dynamic obstacles:

- Martians: move, intelligence varies

Boundaries act like boulders

Top 20 Teams (Main Contest)

1	Team Smartass	Java
2	shinh	C++
3	Error 404	Java
4	LazyBear	Haskell
5	SzM	Haskell
6	UBE	C
7	fireinthedisco	C++
8	souris aveugle	C++
9	Giantslalom	C++
10	frank324	C++

11	OmNomNomNom	Java
12	kstm.org	C++
13	Mosquito	Java
14	epriestley	C++
15	intelliarts	Java
16	The B-Team	C
17	Spore...popcap.com	C++
18	cw	Java
19	Blue Iris	Python
20	Side Effects	Python

Java: halfway to Refal

... we were not out to win over the Lisp programmers; we were after the C++ programmers. We managed to drag a lot of them about halfway to Lisp. Aren't you happy?



-- Guy Steele

Garbage Collection

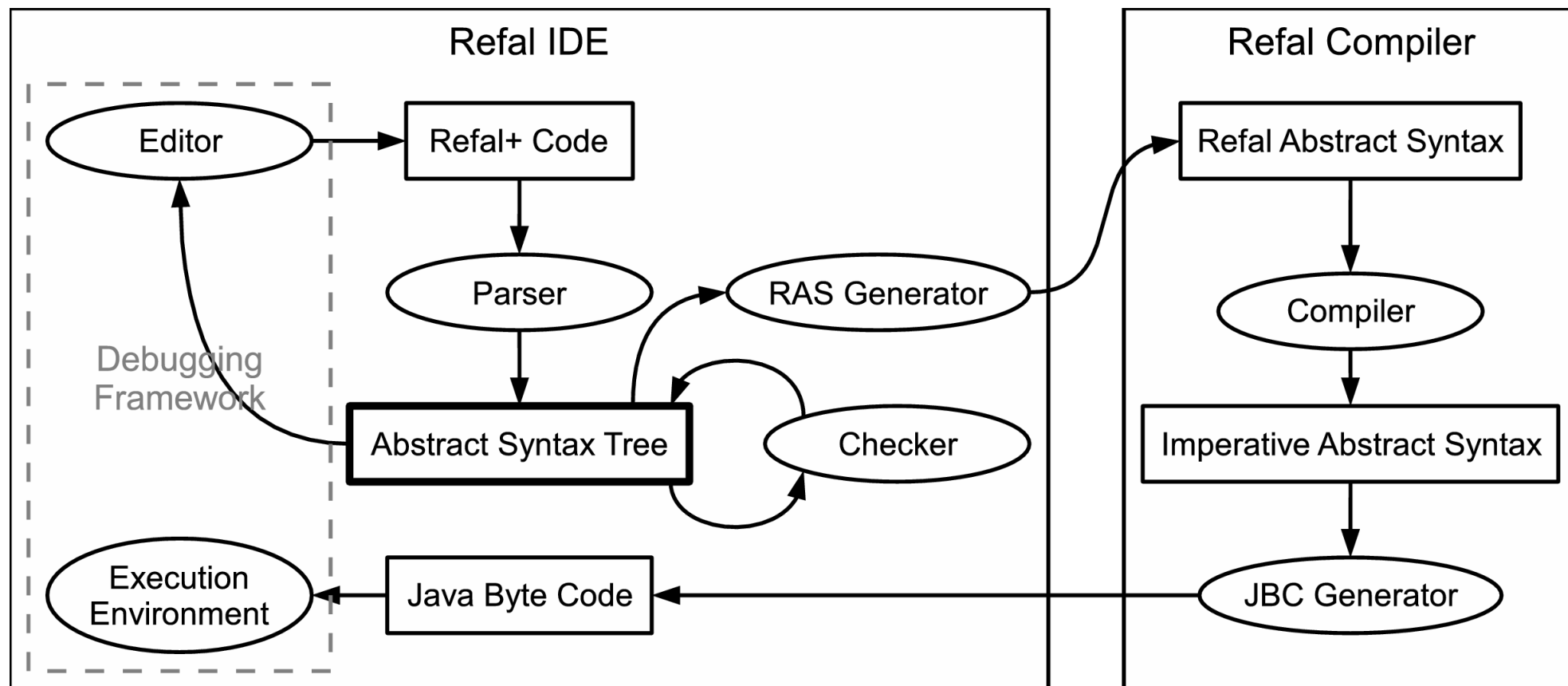




Eclipse как платформа

- Много языков
- Привычная среда
- Огромное активное сообщество
- Java

Схема работы RfpDT



Интерфейс Рефал-Ява

Зачем он нужен? Пример

```
$func Nub e.list = e.list;  
Nub e.list = <NubAux () e.list>;  
  
$func NubAux (e.nubbed) e.list = e.nubbed;  
NubAux {  
    (e.nubbed) /*empty*/ = e.nubbed;  
    (e.nubbed) t1 e2 = {  
        e.nubbed : e t1 e = <NubAux (e.nubbed) e2>;  
        <NubAux (e.nubbed t1) e2>;  
    };  
};
```

Параллельный NUB на Java

```
public static Object[] nub (Object[] array, int n)
{
    List[] bufs = new ArrayList[n];
    ...
    for (Object obj : array)
        bufs[Math.abs(obj.hashCode()) % n].add(obj);
    ...
    for (final List buf : bufs)
        new Thread() {
            public void run() {
                try {
                    ... Nub.Nub(buf.toArray()) ...
                } catch (RefalException _) {}
                ...
            }
        }.start();
    ...
}
```

There and back again

```
$func JNub e.list s.n = e.list;  
$native JNub = "static java.lang.Object[]  
                jnub(java.lang.Object[], int)";  
  
$func Main = e;  
Main = ... <JNub e.list s.n> ... ;  
  
public static Object[] jnub (Object[] array, int n) {  
    ...  
    Nub(buf);  
    ...  
}  
  
$native "java.lang.Object[] Nub(java.lang.Object[])" = Nub;  
  
$func Nub e.list = e.list;  
Nub e.list = ...;
```

Ошибка!

```
public static Object[] nub (Object[] array, int n)
{
    List[] bufs = new ArrayList[n];
    ...
    for (Object obj : array)
        bufs[Math.abs(obj.hashCode()) % n].add(obj);
    ...
    for (final List buf : bufs)
        new Thread() {
            public void run() {
                try {
                    ... Nub.Nub(buf.toArray()) ...
                } catch (RefalException _) {}
                ...
            }
        }.start();
    ...
}
```

О чем было рассказано

- IDE **бывают** полезны
- Каждой задаче – подходящий инструмент!
- Ява хороша как универсальный язык, Рефал – как язык для «интеллектуальных» подзадач
- Рефал и Ява взаимодействуют легко
- Особенно в Eclipse

<http://rfp.botik.ru>

<http://wiki.botik.ru/Refaldevel/EclipseDemoCamp2008>