Создание языка программирования для системы тарификации

Волков Илья
Малышев Александр

СБЕРБАНК ТЕХНОЛОГИИ

JPoint 2017

Тарификация – основа основ

СБЕРБАНК ТЕХНОЛОГИИ

- вклады
- переводы
- платежи
- кредиты
- страхование
- …

Хранение тарифов

СБЕРБАНК ТЕХНОЛОГИИ

Перевод в рублях в другой банк - ..%

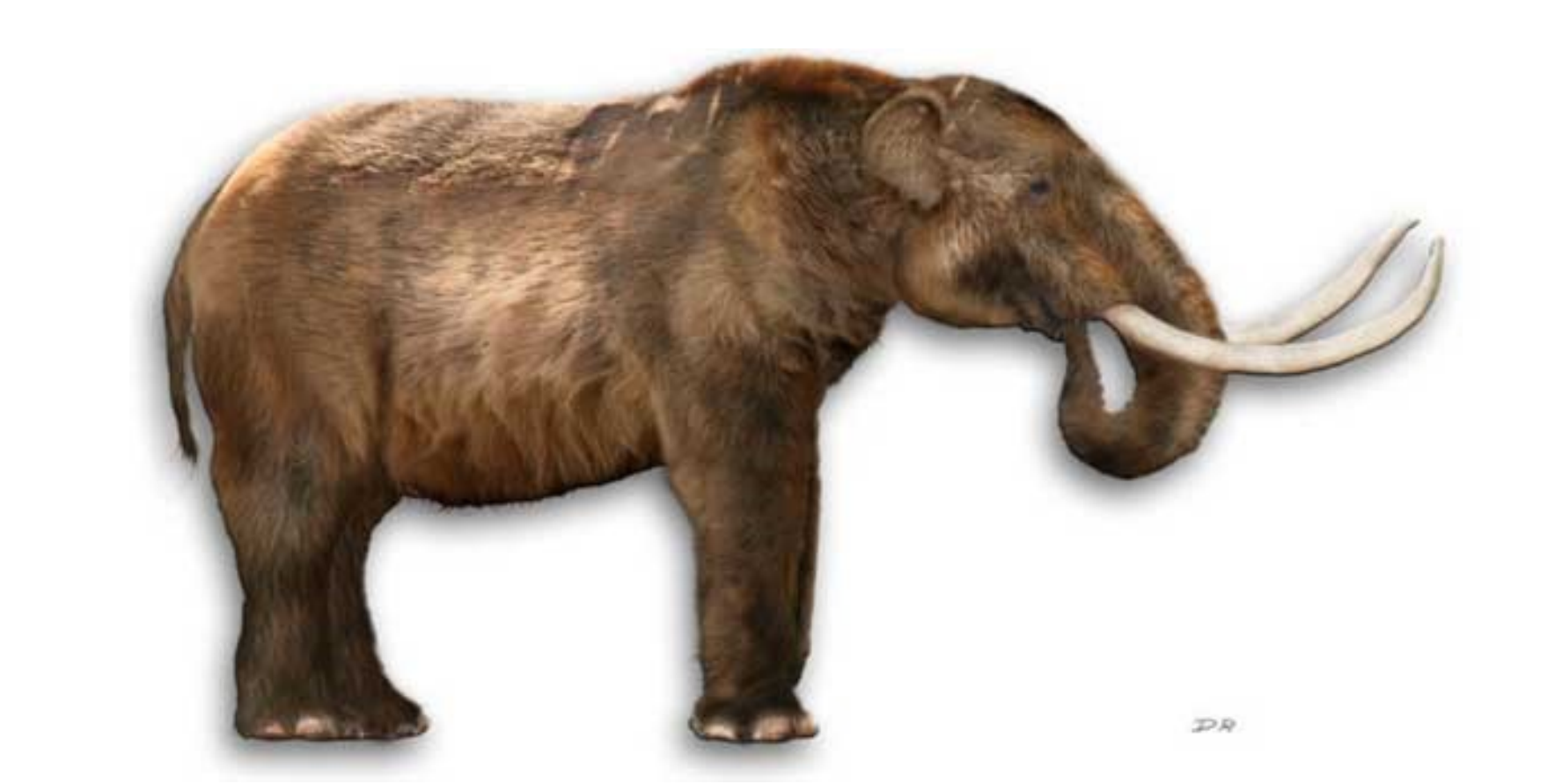
Вклад «Универсальный» в рублях на 1 год - ..%

..%

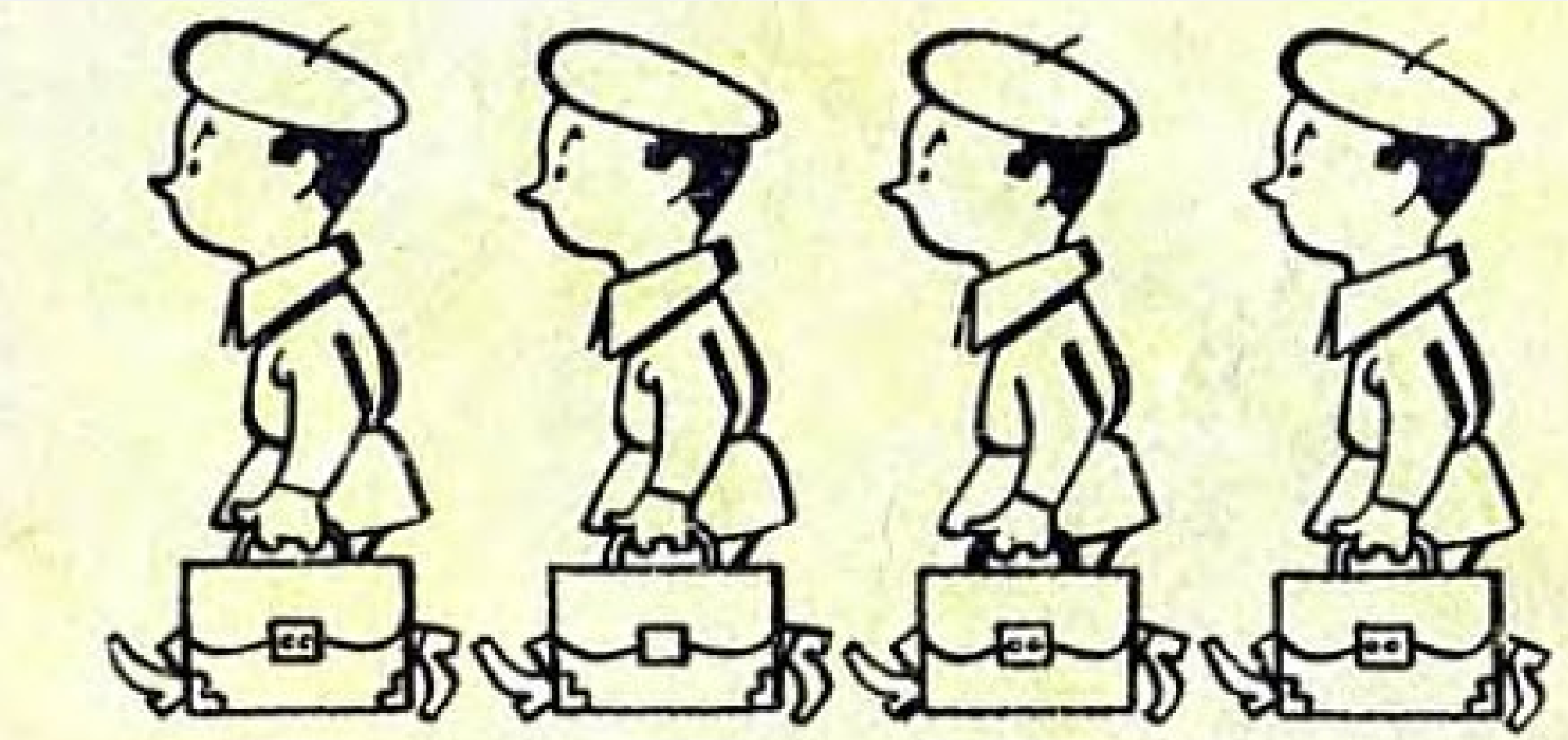
..%

Существующие проблемы

№1: Legacy системы



№2: дублирование логики



№3: просим одно, а получаем нечто иное

№4: длинный цикл разработки

Объединение логики тарификации

СБЕРБАНК ТЕХНОЛОГИИ

Систем много

Кода по тарификации тоже много

Нужно собрать!

Разработчик пишет framework

- Инструмент отдается в руки бизнесу
- Бизнес сам пишет алгоритмы

Ключевые особенности нового framework’a

СБЕРБАНК ТЕХНОЛОГИИ

Модульные изменения

Алгоритм прозрачен для бизнеса

Ключевые особенности нового framework’a

Быстрое внедрение изменений

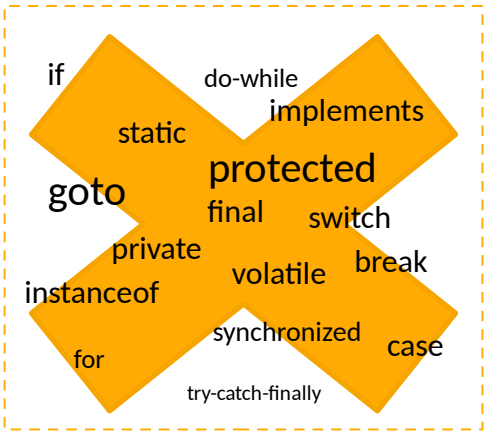
Механизм автотестов

Решение

Domain-specific language

СБЕРБАНК ТЕХНОЛОГИИ

General Purpose Language = GPL



Задачи предметной области (DSL)

Упрощенная грамматика

+

Предметная лексика

Без модели никуда!

СБЕРБАНК ТЕХНОЛОГИИ

DSL для расчета объема аквариума?

Да легко: ширина * высота * глубина

СБЕРБАНК ТЕХНОЛОГИИ

Семантическая модель

А так ли легко?

СБЕРБАНК ТЕХНОЛОГИИ

Семантическая модель тарификации

СБЕРБАНК ТЕХНОЛОГИИ

- Продукт
- Сервис
- Операция
- Тарифная сетка
- …

Шаг №1: внутренний DSL на Scala

СБЕРБАНК ТЕХНОЛОГИИ

```
if (клиент.ВИП) 15.0
else if (клиент.баланс > 10000 and клиент.пол == Мужской) 15.0
else if (клиент.баланс > 5000 and клиент.возраст < 30) 13.5
else if (суммаВклада > 100000) 13.0
else 12.0
```

Шаг №2: внешний DSL - парсеры

Scala Parser Combinators

Требования к инструменту

СБЕРБАНК ТЕХНОЛОГИИ

- **Отладка**

Требования к инструменту

СБЕРБАНК ТЕХНОЛОГИИ

- Отладка
- Framework тестирования

Требования к инструменту

СБЕРБАНК ТЕХНОЛОГИИ

- Отладка
- Framework тестирования
- Переиспользование кода

Требования к инструменту

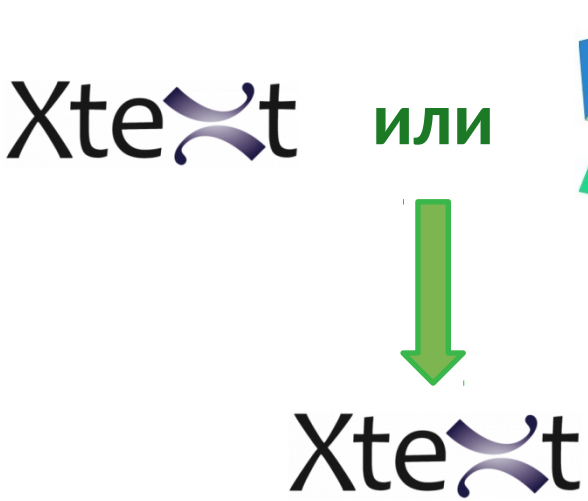
СБЕРБАНК ТЕХНОЛОГИИ

- Отладка
- Framework тестирования
- Переиспользование кода
- content assist, syntax highlighting, …

Выбор Language Workbench

или

Выбор Language Workbench



Так как: Xtext 2.9.0 (06/11/2015) – поддержка Web редактора

- Текстовые языки
- Парсер
- Модель АСТ
- IDE

Первый прототип - редактор

СБЕРБАНК ТЕХНОЛОГИИ

```
выход.комиссия := базовая_ставка(вход.код_валюты, вход.сумма);

льгота := получить_льготную_ставку_из_ПУ(вход.код_валюты, вход.сумма,

выход.комиссия := применить_льготу(выход.комиссия, льгота);

выход.комиссия := выход.комиссия = 0 => выход.комиссия + 0.5;

выход.валюта_комиссии := 810;

результат [выход.валюта_комиссии, выход.комиссия];
```

Первый прототип - тестирование

СБЕРБАНК ТЕХНОЛОГИИ

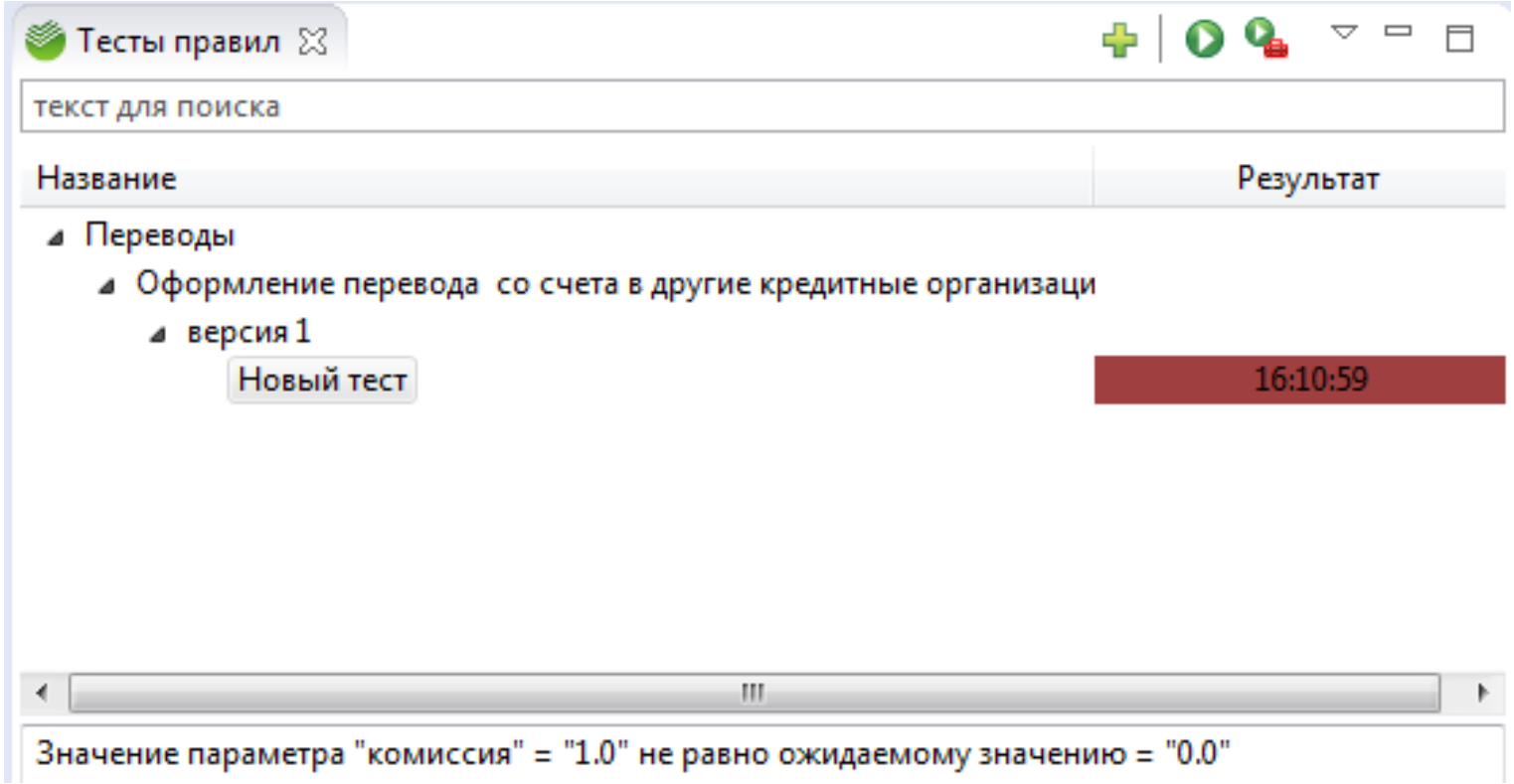
Параметры теста

текст для поиска

Название	Тип	Значение
◢ Входные		
код_валюты	Целое число	810
направление_перевода	Строка	
номер_счета_получателя	Строка	
сумма	Дробное число	15000
◢ Выходные		
валюта_комиссии	Целое число	810
комиссия	Дробное число	0

Первый прототип - тестирование

СБЕРБАНК ТЕХНОЛОГИИ



Грамматика

Грамматика Xtext

- **EBNF = расширенная форма Бэкуса-Наура**

Грамматика Xtext

СБЕРБАНК ТЕХНОЛОГИИ

- **EBNF = расширенная форма Бэкуса-Наура**
- **Определяем терминалы. Например:**

```
terminal fragment ALPHA: (‘a’..’я ’ | ‘А’..’Я’);
terminal ID: ALPHA*;
```

Грамматика Xtext

СБЕРБАНК ТЕХНОЛОГИИ

- EBNF = расширенная форма Бэкуса-Наура
- Определяем терминалы. Например:
- На основе терминалов создаем нетерминалы

```
terminal ID: ALPHA*;
Var: name = ID;
```

Грамматика – структуры

клиент.документ.дата_выдачи;

ObjectFieldAccess: variable = Var ("." objects += ID)*;

Грамматика – присваивание


```
комиссия := 0;
```

```
Assignment:
variable = ObjectFieldAccess ':='  value = AssignmentValue;
...
```

Грамматика – условное присваивание

СБЕРБАНК ТЕХНОЛОГИИ

комиссия := *валюта* = *810* => *0*;

```
AssignmentValue: Branch;
...
Branch: (left = Or then = '=>')? (right = Value) delim=';‘
```

Внешние функции

- Обращение к внешним ресурсам
- Пишутся разработкой

Внутренние функции

СБЕРБАНК ТЕХНОЛОГИИ

```
/* выделение суммы НДС */
выделить_НДС это дробное_число (сумма это
дробное_число) {
    результат сумма * 0.18 / 1.18;
}
```

Внутренние функции


```
сумма := 100;
ндс := выделить_НДС(сумма); /* 15.254237.... */
```

Что произойдет если поправить код выделить_НДС()?

```
результат сумма * 0.18 / 1.18 + 100;
```

Ничего не произойдет, так как inline

СБЕРБАНК ТЕХНОЛОГИИ

```
сумма := 100;
ндс := выделить_НДС(сумма); /* 15.254237.... */
...
выделить_НДС это дробное_число (сумма это
дробное_число) {
    результат сумма * 0.18 / 1.18;
}
```

Inline функций – хорошо ли это?

СБЕРБАНК ТЕХНОЛОГИИ

Хорошо:

- **Работающее правило не сломается**

Inline функций – хорошо ли это?

СБЕРБАНК ТЕХНОЛОГИИ

Хорошо:

- Работающее правило не сломается

Плохо:

- Исправления ошибок также не появятся

Inline функций – хорошо ли это?

СБЕРБАНК ТЕХНОЛОГИИ

Хорошо:

- Работающее правило не сломается

Плохо:

- Исправления ошибок также не появятся

Нужна версионность функций

Проблемы при проектировании грамматики – **левая рекурсия**

Левая рекурсия

- сверху вниз
- слева направо

Простой пример: 1+2+5+3+...

СБЕРБАНК ТЕХНОЛОГИИ

```
terminal fragment NUM: ('0'..'9');
```

Простой пример: 1+2+5+3+...

СБЕРБАНК ТЕХНОЛОГИИ

```
terminal fragment NUM: ('0'..'9');
Expression: Expression ('+') (Expression | NUM);
```

Получаем левую рекурсию


```
terminal fragment NUM: ('0'..'9');
Expression: Expression ('+') (Expression | NUM);
```

```
Expression readExpression() {
    left = readExpression(); // рекурсия
    ...
}
```

Избавляемся от левой рекурсии

СБЕРБАНК ТЕХНОЛОГИИ

```
terminal fragment NUM: ('0'..'9');
Expression: NUM Subexpression;
Subexpression: ('+' NUM Subexpression | );
```

Проблемы при проектировании грамматики – **backtrack**

Неоднозначность грамматики – было так

СБЕРБАНК ТЕХНОЛОГИИ

```
{
  условие1 => результат1;
  условие2 => результат2;
  ...
  условиеN => результатN;
}
```

Например

СБЕРБАНК ТЕХНОЛОГИИ

```
{
  возраст_клиента > 55 => 5;
  баланс_клиента > 10000 => 6;
  …
}
```

Добавили значение по умолчанию

СБЕРБАНК ТЕХНОЛОГИИ

```
{
  условие1 => результат1;
  условие2 => результат2;
  ...
  результатN; /* без условия */
}
```

Например


```
{
   возраст_клиента > 55 => 5;
   баланс_клиента > 10000 => 6;
   ...
   10;
}
```

Добавили в грамматику DefaultBranch

СБЕРБАНК ТЕХНОЛОГИИ

Selection:
‘{‘
    branches += (**ConditionalBranch** | **DefaultBranch**)
‘}’

Добавили в грамматику DefaultBranch

СБЕРБАНК ТЕХНОЛОГИИ

```
ConditionBranch:
        Condition ‘=>’ Result;
;
DefaultBranch:
        Result;
;
```

ConditionBranch или DefaultBranch?

СБЕРБАНК ТЕХНОЛОГИИ

10;

Ошибка – не дойдем до DefaultBranch

СБЕРБАНК ТЕХНОЛОГИИ

```
10;
```

```
ConditionBranch:
        Condition ‘=>’ Result;
;
DefaultBranch:
        Result;
;
```

Как сделать, чтобы работало

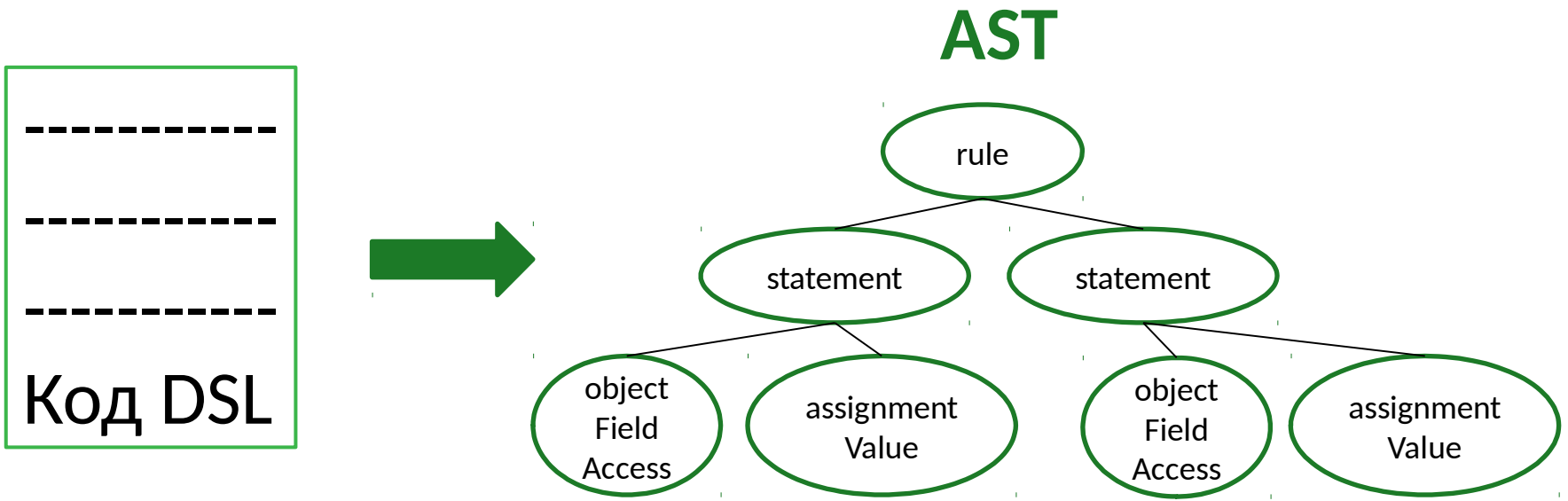
СБЕРБАНК ТЕХНОЛОГИИ

Включаем опцию backtrack = true

```
ConditionBranch:
    Condition ‘=>’ Result;
;
DefaultBranch:
    Result;
;
```

Механизм выполнения

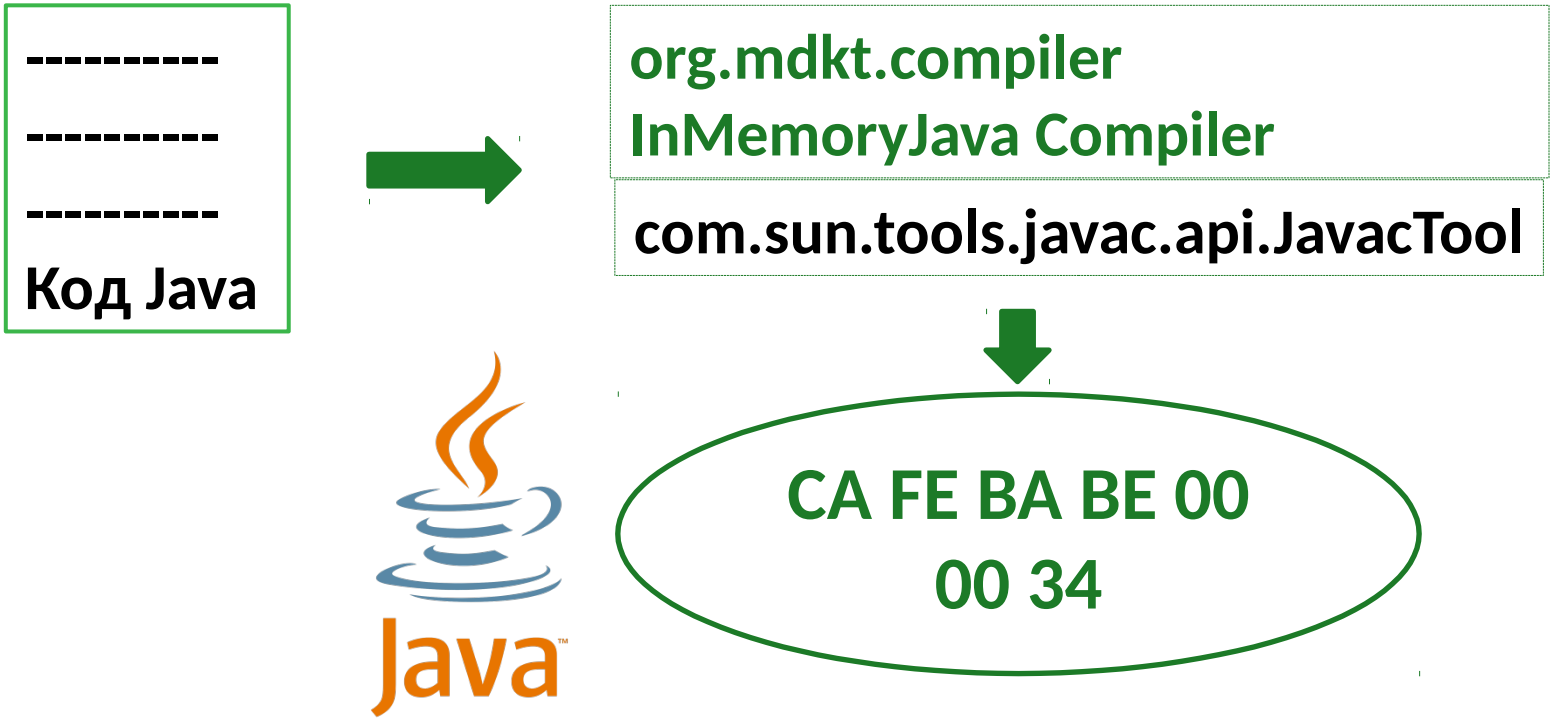
Абстрактное Синтаксическое Дерево



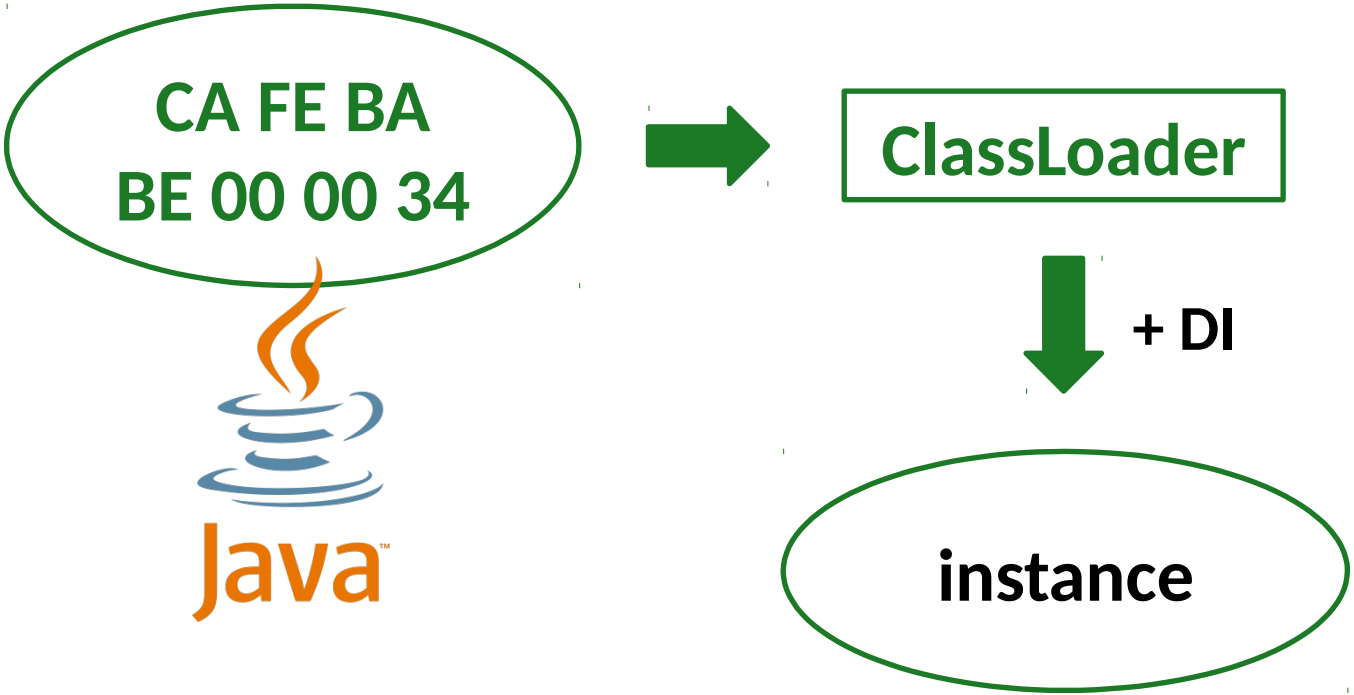
Трансляция AST в Java


```
if (greater(/*баланс_клиента*/ f_Balance.apply(), 10000.0)) {
    /*комиссия*/ out_amountFee = new BigDecimal("0.0");
else {
    /*комиссия*/ out_amountFee = new BigDecimal("50.0");
}
```

DSL – исполняемый код



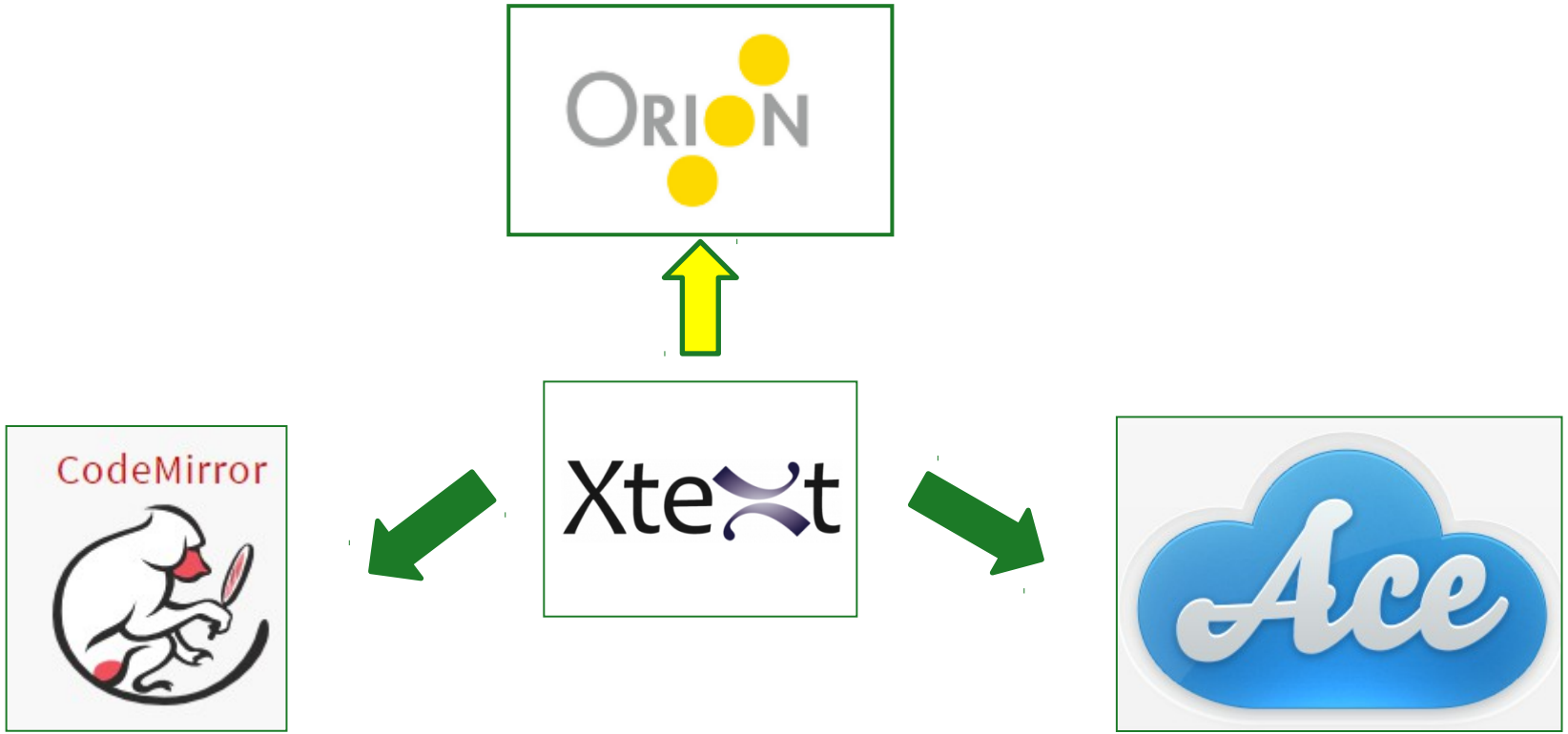
Runtime (AppServer, standalone, compute grid)



Инструменты для написания кода DSL

Xtext Web - редактор

СБЕРБАНК ТЕХНОЛОГИИ



Механизм тестирования

СБЕРБАНК ТЕХНОЛОГИИ

ВХОД

- сумма: 550
- Валюта: 810

ВЫХОД

- комиссия: 5.50

Механизм тестирования

СБЕРБАНК ТЕХНОЛОГИИ

Функции (реальный вызов)

баланс_по_счету(47501…)

Механизм тестирования

СБЕРБАНК ТЕХНОЛОГИИ

Функции (реальный вызов)

баланс_по_счету(47501…)

Результат

5 000 ?

Механизм тестирования

СБЕРБАНК ТЕХНОЛОГИИ

Функции (реальный вызов)

баланс_по_счету(47501…)

Результат

1 000 000 ?

Механизм тестирования

СБЕРБАНК ТЕХНОЛОГИИ

Функции (mock)

баланс_по_счету(47501…)

Результат

15 700

Путь к отладке – лог трассировки

СБЕРБАНК ТЕХНОЛОГИИ

```
<< 150 >> сумма := << 150 >>  получить_сумму(код_операции);
<< 0 >> Комиссия := << 150 >> сумма > 100 => 100;
<< 10 >> Комиссия := << 150 >> комиссия = 0 => 10;
```

Путь к отладке – лог трассировки


```
<< 150 >> сумма := << 150 >>  получить_сумму(код_операции);
<< 0 >> Комиссия := << 150 >> сумма > 100 => 100;
<< 10 >> Комиссия := << 150 >> комиссия = 0 => 10;
```

- **Для отладки тестов**

Путь к отладке – лог трассировки

СБЕРБАНК ТЕХНОЛОГИИ

```
<< 150 >> сумма := << 150 >>  получить_сумму(код_операции);
<< 0 >> Комиссия := << 150 >> сумма > 100 => 100;
<< 10 >> Комиссия := << 150 >> комиссия = 0 => 10;
```

- **Для отладки тестов**
- **Для разбора выполнения правил на runtime**

Тонкая грань между DSL и GPL

Domain-Specific

Использование DSL не по назначению

СБЕРБАНК ТЕХНОЛОГИИ

Литература

- **Implementing Domain-Specific Languages with Xtext and Xtend - Second Edition - Lorenzo Bettini.**
  ISBN: 9781786464965
- **Language Implementation Patterns - Terence Parr.**
  ISBN: 9781934356456
- **The Definitive ANTLR Reference - - Terence Parr.**
  ISBN: 9780978739256

Q&A

id.volkov@gmail.com

ghostler.smbox@gmail.com