

ЯЗЫКИ ПРОГРАММИРОВАНИЯ С РАСШИРЯЕМЫМ СИНТАКСИСОМ

П.В. Егоров

Екатеринбург,
Июнь 2006

Расширение языков

Классические способы расширения:

- Библиотеки функций (Pascal, C, ...)
- Расширение типов (ООП)

В данной работе:

- **Расширение синтаксиса**

Пример

```
syntax Sql;
```

```
string personName = "Иван";
```

```
SqlQuery q = sql(select count(*)  
                from Persons  
                where name = $personName);
```

```
int IvansCount = connection.Execute(q);
```

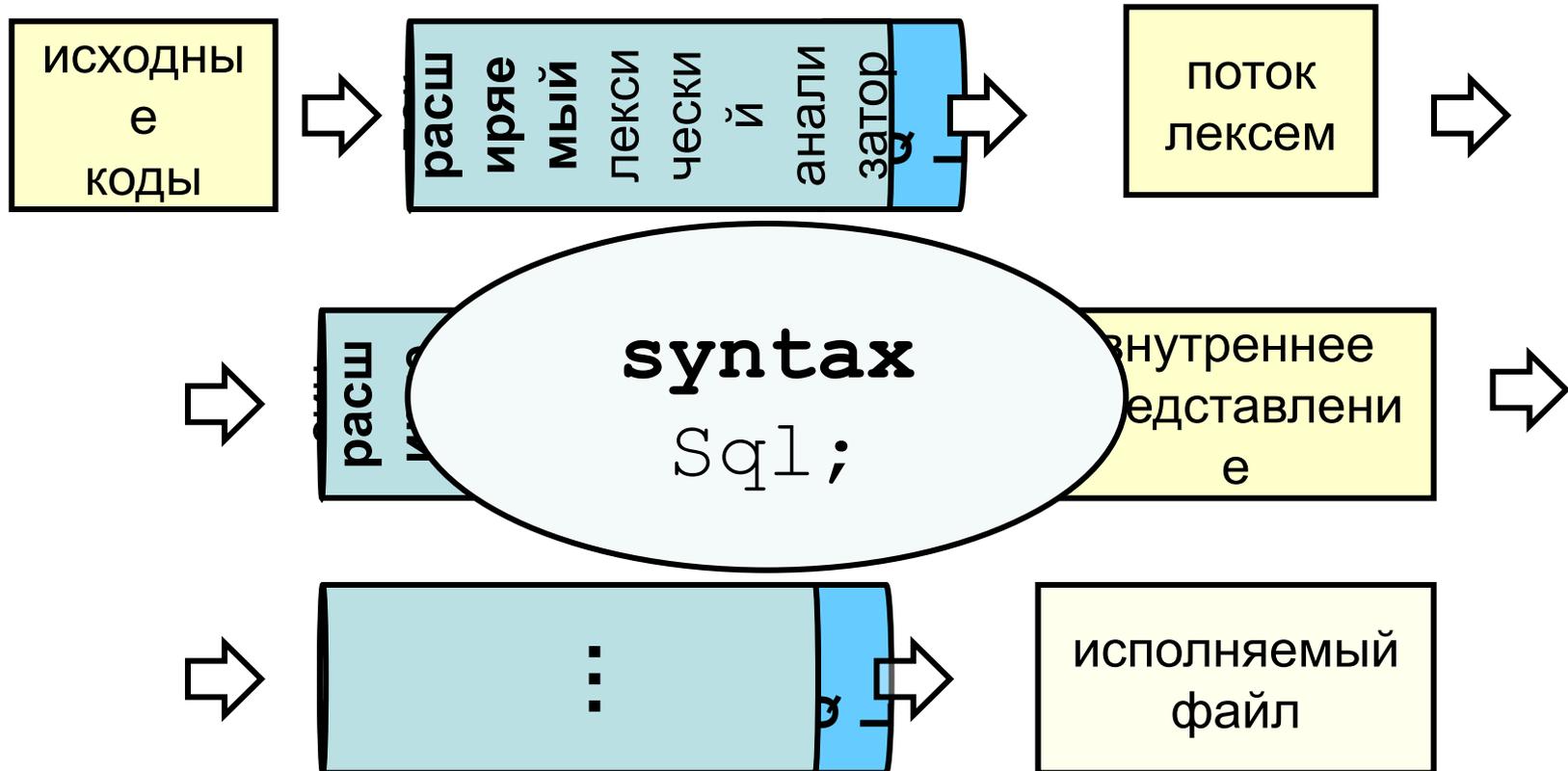
Что это даёт?

- ✓ Удобство синтаксиса
- ✓ Проверки на этапе компиляции
- ✓ Возможность гибкой поддержки многих технологий и парадигм

BEGIN
END

Трансляция

«BEGIN»
«END»



Лексический анализатор

Классические способы построения:

- На базе регулярных выражений и конечных автоматов (LEX)
- «Подглядывающий» ЛА

Расширяемый ЛА

Особенности:

- Гибкая динамическая настройка лексического анализатора
- Добавлении нового типа лексем → конфликт с добавленными ранее лексемами

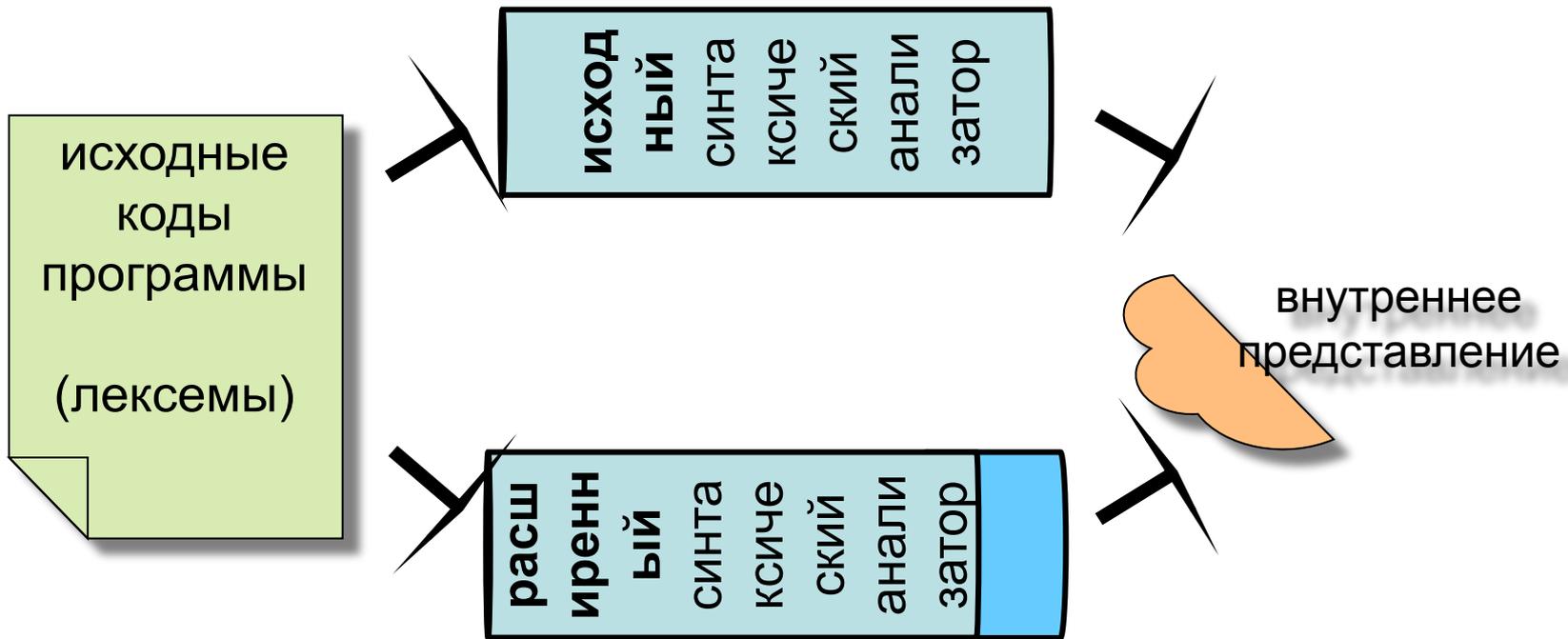
Пример: "<=" – это одна лексема "<=" или две лексемы "<" и "="

Синтаксический анализатор

- СА управляется **формальной грамматикой**
- Суть расширения – модификация формальной грамматики
- Как же её модифицировать?

Синтаксический анализатор

Совместимость расширения



Синтаксический анализатор

Расширение

- Расширение – последовательность преобразований грамматики

$$G1 \xrightarrow{T_1} G2 \rightarrow \dots \xrightarrow{T_n} G_n$$

- Каждое преобразование сохраняет совместимость
- Система безопасных преобразований

Практические результаты

Реализованы (на языке Java):

- Расширяемый «подглядывающий» ЛА
- Расширяемый предсказывающий LL(1) синтаксический анализатор
- Для модельного языка Pascal-S разработана грамматика ядра и ряд его независимых расширений

ВЫВОДЫ

Языки программирования с
расширяемым синтаксисом являются
практически реализуемыми!

За этим подходом будущее!?!..

Вопросы?