

Language Oriented Programming (LOP)
в действии,
как мы это делаем в JetBrains

Максим Мазин

JetBrains

Старший инженер-программист

Maxim.Mazin@jetbrains.com

План

- Зачем расширять языки?
- Проблемы создания DSL
 - Совместимость
 - Поддержка со стороны IDE
- Подход среды MPS
 - Основная идея
 - Демонстрация
- MPS и предметно-ориентированные языки

Зачем расширять языки?

- **Универсальные языки**
 - Ограниченный набор конструкций общего назначения
 - Избыточность в описании предметной области
 - Низкая сопровождаемость кода
- **Предметно-ориентированные языки**
 - Более высокий уровень абстракции
 - Лучшая сопровождаемость кода

Неявные языковые расширения

Предметно-ориентированные конструкции в универсальных языках

- Блок **synchronized** в языке *Java*
- Перегрузка операторов в языке C++

Блок `synchronized` в Java

Вместо

```
lock.lock();  
try {  
    // полезный код  
} finally {  
    lock.unlock();  
}
```

Можно написать

```
synchronized (lock) {  
    // полезный код  
}
```

Перегрузка операторов в C++

Вместо

```
Complex c = new Complex(1,2);  
Complex d = c.add(a);
```

Можно написать

```
complex c(1,2);  
complex d = c + a;
```

Языковые расширения

- Существующие расширения
 - Удовлетворяют нужды конкретных предметных областей
 - Встроены в универсальные языки
- Новые расширения
 - Трудно встраивать
 - Риск несовместимости

План

- Зачем расширять языки?
- Проблемы создания DSL
 - Совместимость
 - Поддержка со стороны IDE
- Подход среды MPS
 - Основная идея
 - Демонстрация
- MPS и предметно-ориентированные языки

Совместимость

Компоненты совместимы, когда могут быть использованы совместно даже, если были созданы независимо.

Совместимость

Конечная цель

Java-библиотеки

- Hibernate
- Spring
- Joda Time



Расширения Java

- DB Language
- DI Language
- Dates Language

Совместимость

Библиотеки и расширения

- Библиотеки совместимы
 - Java + Hibernate = ОК
 - Java + Spring = ОК
 - Java + Hibernate + Spring = ОК
- Языковые расширения — нет
 - Java + extension A = ОК
 - Java + extension B = ОК
 - Java + extensions A, B = возможна неоднозначность

Совместимость

Грамматическая неоднозначность

- **Расширение A**

```
int resultsCount = <some_code>;  
string s = "We found ${resultCount} results";
```

- **Расширение B**

```
int resultCount = <some_code>;  
string t = "We found {resultCount} results";
```

- **A + B** программа становится **неоднозначной**

```
string s = "Account balance is ${balance}";
```

Разработка языка

Требования

- Общая инфраструктура
 - Синтаксис
 - Система типов
 - Операционная семантика
 - и т.д.

Разработка языка

Требования

- **Общая инфраструктура**
 - Синтаксис
 - Система типов
 - Операционная семантика
 - и т.д.
- **Поддержка со стороны IDE**
 - Редактор
 - Подсветка ошибок
 - Рафакторинги
 - Контроль версий
 - Отладчик
 - и т.п.

Разработка языка

Проблемы

- **Ресурсоемкая задача**
 - Сложные алгоритмы обработки кода
 - Разнообразии поддерживаемых технологий
 - Необходимость высокой квалификации разработчиков
- **Совместимость инфраструктуры**
 - Совместимость систем типов различных расширений
 - Поддержка возможных расширений рефакторингами
 - и т.д.

Существующие решения

	Совместимость	Языковая инфраструктура
LISP	Да	Нет
Внутренние языки в Ruby / Groovy	Да	Нет
XText framework	Нет	Да

Общие проблемы

- Существующие решения обеспечивают
 - Либо совместимость языковых расширений
 - Либо языковую инфраструктуру
- Но не одновременно

План

- Зачем расширять языки?
- Проблемы создания DSL
 - Совместимость
 - Поддержка со стороны IDE
- **Подход среды MPS**
 - Основная идея
 - Демонстрация
- MPS и предметно-ориентированные языки

JetBrains MPS

- MetaProgramming System
- Среда для создания и расширения языков
- IDE для созданных языков и расширений

Основная идея

Текстовые грамматики приводят к
неоднозначности



Необходим иной подход



MPS работает с абстрактным
синтаксическим деревом (АСД) напрямую

Основная идея

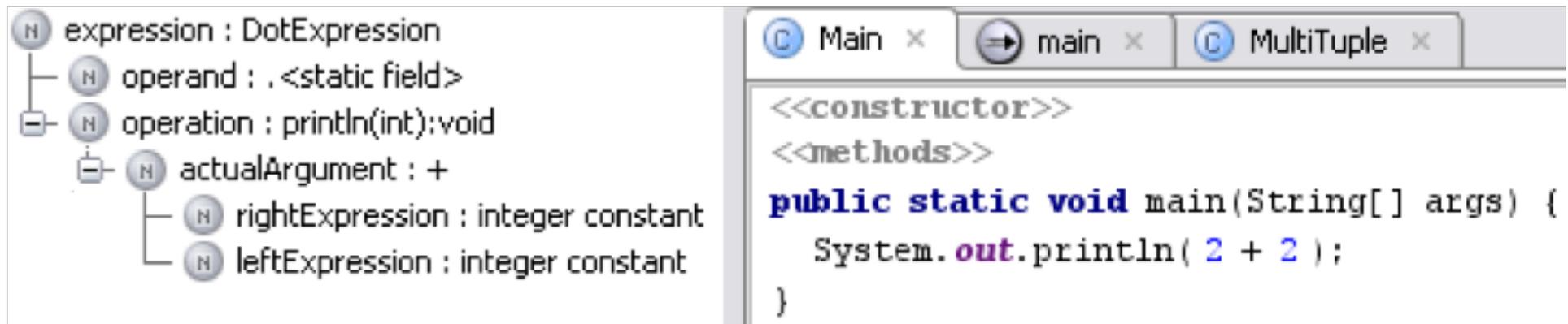
Редактирование АСД

- **Предыдущие попытки**
 - Редактирование диаграмм
 - Ограниченный набор применений

- **Решение MPS**
 - Сохраняет привычную для пользователя среду
 - Поддерживает текстоподобное редактирование

Основная идея

Проекционный редактор



- У каждого узла синтаксического дерева есть **проекция** в редакторе MPS

Основная идея

Проекционный редактор

The image shows a projection editor interface. On the left is a syntax tree for a Java expression. The root node is 'expression : DotExpression'. It has three children: 'operand : . <static field>', 'operation : println(int);void', and 'actualArgument : +'. The 'actualArgument' node has two children: 'rightExpression : integer constant' and 'leftExpression : integer constant'. On the right is a code editor window titled 'Main x' with a sub-window 'main x' and another tab 'MultiTuple x'. The code in the editor is:

```
<<constructor>>
<<methods>>
public static void main(String[] args) {
    System.out.println(2 + 2);
}
```

 The numbers '2' and '2' in the code are highlighted with colored boxes (orange and green) that correspond to the 'actualArgument' and its children in the tree.

- Каждый узел «живет» в своей ячейке
- Узел отображается в виде текстового/символьного представления
- Редактирование проекции мгновенно отражается на синтаксическом дереве

Основная идея

Обычный текст и проекции

- У текстоподобного проекционного редактора есть преимущества и недостатки
- Привыкание — около двух недель

Создание языков

- **Инфраструктура создания языков**
 - Мета модель языка (абстрактный синтаксис)
 - Система типов
 - Редакторы (конкретный синтаксис)
 - Компилятор (операционная семантика)
 - и т.д.
- **Средства IDE**
 - Автозавершение кода
 - Поиск использований
 - Переименование
 - Отладка
 - и т.д.

Готовые MPS-языки

- Реализация Java™ в среде MPS + расширения
 - Функциональные коллекции (collections language)
 - Даты (dates language)
 - Замыкания (closures language)
 - Регулярные выражения (regex language)
 - и т.д.
- Языки для определения языков
 - Реализованы с помощью самих себя (bootstrapping)
- Разнообразные языки (XML и т.п.)

Демонстрация

Добавление конструкции read lock

Java

```
ReadWriteLock l = ...  
l.readLock().lock();  
try {  
    // полезный код  
}  
finally {  
    l.readLock().unlock();  
}
```



Java + Расширение

```
ReadWriteLock l = ...  
read (l) {  
    // полезный код  
}
```

План

- Зачем расширять языки?
- Проблемы создания DSL
 - Совместимость
 - Поддержка со стороны IDE
- Подход среды MPS
 - Основная идея
 - Демонстрация
- MPS и предметно-ориентированные языки

Domain-Specific Languages

- Среда MPS прекрасно подходит для создания DSL
- Общие языковые части могут быть переиспользованы (например, выражения)
- Можно внедрять конструкции универсальных языков программирования (например, Java) внутрь своих DSL

Domain-Specific Languages

Пример

```
plan Regular
```

```
value QUANTITY BASE_RATE
```

```
PAST : 10.0 USD_KWH
```

```
1999 - 12 - 01 : 12.0 USD_KWH
```

```
event USAGE ( TAXABLE )
```

```
1999 - 10 - 01 : amount : BASE_RATE * usage
```

```
account : base-usage
```

```
event SERVICE CALL ( TAXABLE )
```

```
1999 - 10 - 01 : amount : fee * 0.5 + USD 10.0
```

```
account : service
```

```
1999 - 12 - 01 : amount : fee * 0.5 + USD 15.0
```

```
account : service
```

```
event TAX ( TAXFREE )
```

```
1999 - 10 - 01 : amount : fee * 0.055
```

```
account : tax
```

Существующие приложения

- YouTrack – bug & issue tracker
 - Целиком разработана в MPS
- Языки для web-разработки
 - DNQ (Data Navigation and Queries)
 - Webr
 - Spring language
 - и т.д.

Существующие приложения

The screenshot shows the YouTrack web interface. At the top, there is a navigation bar with the YouTrack logo, a 'New Issue' button, and links for 'Issues', 'Dashboard', and 'Administration'. On the right, it says 'Welcome, yamaxim' and 'Help'. Below the navigation bar is a search bar containing the query 'for: me #Unreso'. A dropdown menu is open over the search bar, showing 'Unresolved' as the selected state and a saved search for 'for: me #Unresolved'. The search results show a list of issues. The first issue is JT-85, which is 'Unresolved' and 'Open', created by 'Exception Robot' on '07:58'. The second issue is JT-8496, 'Normal' priority, 'Bug' type, created by 'kradima' on 'Mar 16'. The third issue is JT-8488, 'Normal' priority, 'Feature' type, created by 'yamaxim' on 'Mar 15'. The fourth issue is JT-4649, 'Normal' priority, 'Bug' type, created by 'Kai-Philipp Schoellmann' on 'Mar 15'. The fifth issue is JT-6628, 'Normal' priority, created by 'Vadim Gurnv'. Each issue entry includes a star icon, a status icon, a priority label, a type label, the creator's name, and a comment count.

Navigation: New Issue | Issues | Dashboard | Administration | Welcome, yamaxim | Help

Search: All Projects | for: me #Unreso | Search

Left Sidebar: Tags (117) | Saved Searches (20) | Filters

Issue List:

- JT-85: Unresolved, Open, yamaxim, No Subs, No affect, No fix ve, No Revis, Any Brov, Any OS. by: Exception Robot 07:58. 2 comments.
- JT-8496: Normal, Bug, Submitte, yamaxim, Custom F, No affect, Next EAF, No Revis, Any Brov, Any OS. by: kradima Mar 16. add button.
- JT-8488: Normal, Feature, Submitte, yamaxim, REST AP, No affect, No fix ve, No Revis, Any Brov, Any OS. by: yamaxim Mar 15. add button.
- JT-4649: Normal, Bug, Submitte, yamaxim, No Subs, Daring E, Next EAF, No Revis, Any Brov, Any OS. by: Kai-Philipp Schoellmann Mar 15. 7 comments.
- JT-6628: Normal, by: Vadim Gurnv.

Лицензия

- Open-sourced
- Apache 2.0 License

**MPS — совсем
бесплатный**

Ссылки

Скачать MPS

<http://www.jetbrains.com/mps>

Блог про MPS

<http://blogs.jetbrains.com/mps>

Вопросы?