

ЯЗЫК ПРОГРАММИРОВАНИЯ GROOVY

Groovy — объектно-ориентированный язык программирования, разработанный для платформы Java как дополнение к языку Java с возможностями Python, Ruby и Smalltalk. Groovy использует Java-подобный синтаксис с динамической компиляцией в JVM байт-код и напрямую работает с другим Java кодом и библиотеками. Язык может использоваться в любом Java-проекте или как скриптовый язык. Groovy завершил процесс стандартизации в Java Community Process JSR 241. Возможности Groovy (отличающие его от Java):
Статическая и динамическая типизация
Встроенный синтаксис для списков, ассоциативных массивов, массивов и регулярных выражений
Замыкания
Перегрузка операций

История

Первым упоминанием о языке Groovy было сообщение в блоге Джеймса Стрэчена (англ. James Strachan (programmer))[3] от августа 2003 года. Позднее было выпущено несколько версий между 2004 и 2006 годами. После того, как начался процесс стандартизации JCP, нумерация версий была изменена, и версию называют «1.0». Версия «1.0» была выпущена 2 января 2007 года. В декабре 2007 года вышел Groovy 1.1, эта версия вскоре была перенумерована как «1.5» вследствие значительных изменений в языке. Стрэчен покинул проект за год до релиза Groovy 1.0 в 2007 году, а в июле 2009 года Стрэчен написал в своём блоге, что возможно не создал бы Groovy, если бы в 2003 году прочитал книгу Мартина Одерского с соавторами о программировании на языке Scala (вышедшую в 2007 году)[4]. Проект разработки языка и комитет JSR-241 с 2007 года возглавляет Гийом Лафорж (Guillaume Laforge).

Поддержка IDE

Программирование на Groovy поддерживается в основных интегрированных средах разработки программного обеспечения[5], в частности: IntelliJ IDEA начиная с версии 7 или для более ранних версий с использованием JetGroovy Plugin; Eclipse с использованием Groovy Eclipse; Netbeans — встроен в NetBeans IDE.

Использование Groovy

Последние версии генератора отчётов iReport, основанного на Java-библиотеке JasperReports, позволяют встраивать в отчёты выражения на Groovy и писать на нём дополнительную логику. Система непрерывной интеграции Jenkins позволяет использовать сценарии автоматизации, созданные на Groovy.

Установка Groovy

Groovlets — возможность запускать скрипты на Groovy как сервлеты. GroovyBeans — версия Groovy для JavaBeans. В отличие от Java, в Groovy исходный код может быть выполнен как обычный скрипт, если содержит код вне определения класса или класс с методом main или Runnable или GroovyTestCase:

```
#!/usr/bin/env groovy
println "I can execute this script now!"
```

Строки в Groovy: Java Strings с одинарными кавычками и GStrings с двойными кавычками.

```
def javaStyleString = 'java String style'
def GStringsStyleString = "$javaStyleString"
def j = '$javaStyleString'
def bigGroovyString = "
$javaStyleString
$GStringsStyleString
println bigGroovyString"
```

Groovy неявно генерирует методы для доступа к переменным (setColor(String color) и getColor()):

```
class AGroovyBean {
    String color
}
```

```
def myGroovyBean = new AGroovyBean()
myGroovyBean.setColor('blue')
assert myGroovyBean.getColor() == 'blue'
myGroovyBean.color = 'green'
assert myGroovyBean.color == 'green'
```

Groovy предлагает простой и последовательный доступ к спискам, отображениям и массивам:

```
def myList = ['One', 'Two', 'Three'] //выглядит как массив, но это список
assert myList[2] == 'Three'
```

```

myList[3] = 'Four' //добавляем элемент в список
assert myList.size() == 4
def monthMap = [ 'January' : 31, 'February' : 28, 'March' : 31 ] //определяем
ассоциативный массив
assert monthMap['March'] == 31
monthMap['April'] = 30 //добавляем элемент в ассоциативный массив
assert monthMap.size() == 4
Closure (замыкание) — это анонимная функция и объект в одном виде:
def closureFunction = a, b ->
println a
println b
closureFunction(1, 2)
return в функции указывать не обязательно — по умолчанию будет воз-
вращено значение последней упомянутой переменной. Неизменяемые классы
маркируются с помощью аннотации Immutable:
@Immutable
class ImmutableClass
String stringVariable
Integer integerVariable
def newVariable = new ImmutableClass(stringVariable : "some string integerVariable
: 23)

```