

Work with scala code analyzer

Москвитин Максим

Руководитель: Подхалюзин Александр

Постановка задачи: generified operator in infix expressions

Scala 2.10:

```
case class M[A](a:A){  
  def map[B](f:A=>B):M[B] = M(f(a))  
}
```

```
M(1) map (_.toString)  
res: M[String] = M(1)
```

```
M(1) map[String] (_.toString)  
error: ';' expected but '[' found.
```

Постановка задачи: generified operator in infix expressions

Scala 2.11:

```
case class M[A](a:A){  
  def map[B](f:A=>B):M[B] = M(f(a))  
}
```

```
M(1) map (_.toString)  
res: M[String] = M(1)
```

```
M(1) map[String] (_.toString)  
res: M[String] = M(1)
```

Парсер. Проверка синтаксиса

Из документации:

```
PostfixExpr ::= InfixExpr [id [nl]]
InfixExpr  ::= PrefixExpr
           | InfixExpr id [nl] InfixExpr
PrefixExpr ::= ['-' | '+' | '!' | '~'] SimpleExpr
```

Документация для Scala 2.11 не обновлена до конца.

Парсер. Проверка синтаксиса

НОВЫЙ синтаксис:

```
PostfixExpr ::= InfixExpr [id [nl]]
InfixExpr  ::= PrefixExpr
           | InfixExpr id [{nl} TypeArgs] [nl] InfixExpr
PrefixExpr ::= ['-' | '+' | '!' | '~'] SimpleExpr
```

Типизация инфиксных выражений

Ожидаемый тип для аргументов
выражения:

```
case class M[A](a:A){  
  def map[B](f:A=>B):M[B] = M(f(a))  
}
```

```
M(1) map[Long] (_.toString)
```

```
Type mismatch, expected: (Int) => Long, actual: (Int) => String
```

Типизация инфиксных выражений

Ожидаемый тип для результата
вычисления выражения:

```
case class M[A]() {  
  def foo[B](x: Int): M[B] = M[B]()  
}
```

```
val x: M[Long] = M[Int] foo[String] 1
```

```
Expression of type M[String] doesn't conform to expected type M[Long]
```

Intentions

Method call to infix expression:

```
case class M[A] (a: A) {  
  def foo[B] (x: A => B) :M[B] = M[B] (f(a))  
}
```

```
M(1).map[String](_.toString)
```

Before: intention is not available

```
After: M(1) map[String] (_.toString)
```

Infix expression to method call.

Полученные знания

Scala

IntelliJ IDEA custom language plugin
development

Вопросы?

Спасибо за внимание!