

ЭЛЕМЕНТЫ ФУНКЦИОНАЛЬНОГО ПРОГРАММИРОВАНИЯ В ЯЗЫКЕ RUBY

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Кораблёв Д. А.

Моженкова Е. В. – магистр техн. наук, ассистент, Сечко Г. В. – канд. техн. наук, доцент

Анализируются проблемы соотношения императивных и декларативных подходов в современном программировании. Показывается, что такой высокоуровневый язык программирования общего назначения, как Ruby, который принято считать процедурным и объектно-ориентированным языком, содержит элементы функционального подхода к программированию.

Ruby – интерпретируемый высокоуровневый язык программирования, разработанный Юкихио Мацумото в 1995 году [1-5]. Целью разработки было создание лёгкого в использовании, интерпретируемого языка программирования, где важна не скорость работы написанной программы, а малое время разработки, понятность и простота синтаксиса. Достоинствами языка Ruby являются лаконичный и простой синтаксис, автоматический сборщик мусора и ряд других, а также то, что язык не требует предварительного объявления переменных, хотя для интерпретатора желательно, чтобы переменным присваивалось пустое значение nil.

Для отнесения языка программирования к функциональным необходимо наличие в нём ряда подходов [6-8], одним из которых является наличие функций – объектов первого класса, причём все, что можно делать с данными, можно делать и с функциями (вроде передачи функции другой функции в качестве параметра).

Ruby является мультипарадигменным языком, ориентированным на несколько парадигм программирования. Это значит, что Ruby поддерживает как процедурный стиль (определение функций и переменных вне классов), так и объектно-ориентированный (всё является объектом, даже числа), но кроме них и функциональный стиль (анонимные функции, возврат последнего вычисленного значения и т.д.).

Каждая функция в Ruby является методом.

`proc { |x,y| x + y } #=> Анонимная функция`

```
class ClassName
  def self.method_a    #=> Вызывается без создания экземпляра класса
    puts "First Method"
  end

  def method_a        #=> Вызывается с созданием экземпляра класса
    puts "Second Method"
  end
end

ClassName.method_a    #=> "First Method"
ClassName.new.method_a #=> "Second Method"
```

Вышеприведенный пример подтверждает наличие элементов функционального программирования в Ruby.

Список использованных источников:

1. Фитцджеральд М. Изучаем Ruby. – Санкт-Петербург: Издательство: БХВ-Петербург, 2008. – 331 с.
2. Флэнаган Д., Ю. Мацумото Ю. Язык программирования Ruby. – Санкт-Петербург: Питер, 2011. – 494 с.
3. Фернандес О. Путь Rails. Подробное руководство по созданию приложений в среде Ruby on Rails. – Санкт-Петербург: Символ-Плюс, 2009. – 768 с.
4. Фултон Х. Программирование на языке Ruby. – Санкт-Петербург: ДМК Пресс, 2007. – 688 с.
5. Томас Д. Х. Хэнссон Д.Х. Гибкая разработка веб-приложений в среде Rails. – Санкт-Петербург: Питер, 2008 – 464 с.
6. Аксёнов В.И. PYTHON как язык, поддерживающий несколько парадигм программирования // 49-я науч. конф. аспирантов, магистрантов и студентов БГУИР по направлению 8: Информационные системы и технологии: тез. докл. (Минск, 4 мая 2013 года). – Мн.: БГУИР, 2013. – 91 с. с ил. – С. 59-60.
7. Функциональное мышление: Часть 1. Разработка программ в функциональном стиле [Электронный ресурс] – Режим доступа: <http://www.ibm.com/developerworks/ru/library/j-ft1/index.html>. – Дата доступа: 20.02.2014.
8. Селивошко В.И., Шинкевич А.А. Функциональное программирование в JVM // 49-я науч. конф. аспирантов, магистрантов и студентов БГУИР по направлению 8: Информационные системы и технологии: тез. докл. (Минск, 4 мая 2013 года). – Мн.: БГУИР, 2013. – 91 с. с ил. – С. 81-83.