УДК: 004.432.2

**А.Г. Проаспэт**

Магистрант. Государственное бюджетное образовательное учреждение высшего образования Московской области «Технологический университет»

Королёв, РФ

E-mail: [79647204962@mail.ru](mailto:79647204962@mail.ru)

**ИСПОЛЬЗОВАНИЕ ЯЗЫКА COFFEESCRIPT ДЛЯ РАЗРАБОТКИ СИСТЕМЫ УПРАВЛЕНИЯ КОНТЕНТОМ ИНТЕРНЕТ-РЕСУРСА**

***Ключевые слова:*** *CoffeeScript, JavaScript, интернет-ресурс, разработка интерфейсов, frontend, система управления контентом интернет-ресурса.*

***Аннотация:*** *Язык Coffeescript является одним из самых перспективных языков. Язык JavaScript – самый популярный язык frontend разработки – выполняется в браузере клиента и используется для создания интерактивных элементов веб-страниц. Одной попыток его совершенствования стало создание языка CoffeeScript, главное назначение которого – упростить синтаксис и придать программному коду лаконичность. Код, написанный на языке CoffeeScript при выполнении транслируется в JavaScript.*

*Интернет-ресурсы, созданные с использованием системы управления контентом, поддерживаются разработчиками, которые не являются авторами данной системы, в связи с этим необходимо повышать читабельность исходного кода для расширения возможностей модификации. Способом повысить лапидарность кода является использование языка CoffeeScript для разработки интерфейса.*

Язык CoffeeScript был разработан в 2009 году Джереми Ашкенасом, который также является автором JavaScript-библиотек Underscore.js и Backbone.js [8].

CoffeeScript был создан как «синтаксический сахар» над языком JavaScript. Синтаксический сахар – это конструкции языка программирования, не призванные улучшить функциональные показатели программного кода, но повышающие его лапидарность, делающие его более эстетичным и облегчающие его понимание как самому автору кода, так и другим разработчикам, сопровождающим его [6, с. 187].

В 2017 году язык CoffeeScript был признан одним из семи самых перспективных языков программирования [16]. Последняя версия языка – CoffeeScript 2.1.0 – была выпущена 10 декабря 2017 года [17].

Первый компилятор был написан на языке Ruby, но следующие, начиная с 2010 года, на самом CoffeeScript. Вдобавок к такому преимуществу, как трансляция в JavaScript, в веб-фреймворке Ruby on Rails, начиная с версии 3.1, присутствует поддержка кода на языке CoffeeScript [9]. Тот факт, что язык был создан на основе языков Python, Ruby и Haskell [11], поначалу вызывало некоторое недоверие к нему со стороны противников данных языков, утверждавших, что приверженцы Ruby «ходят в чужой монастырь со своим уставом» и выбирают язык CoffeeScript по той причине, что JavaScript для них является слишком сложным [10].

Haskell является функциональным языком программирования [2, с. 9], языки Ruby и Python, хотя не являются функциональными, могут использоваться для функционального программирования [7, с. 8]. Функциональный стиль программирования непривычен программистам, использующим в работе императивные языки [12]. Этим может быть вызвана критика разработанного на основе данных языков CoffeeScript, действительно визуально похожего на функциональные языки. Однако и JavaScript точки зрения синтаксиса достаточно разнороден. Он разделяет, например, многие идеи языка Scheme, но не его синтаксис - вместо этого в JavaScript используется Cи-подобный синтаксис. Результатом стал язык, позаимствовавший идеи функциональных языков, но с многословным синтаксисом, лишенным естественных конструкций для выражения этих идей [9]. Таким образом, JavaScript не имеет такого преимущества, как большая близость к императивным языкам программирования.

JavaScript позиционирует себя как объектно-ориентированный язык с наследованием, но следует заметить, что он является прототипичным языком. Он не основан на классах, как языки, используемые большинством платформ приложений [9]. В языке CoffeeScript присутствует возможность реализовать качественную абстракцию классов, основанную на прототипах. Реализовав известную модель наследования и дополнив её наследованием методов касса, он позволяет легко строить иерархии классов, не требуя от разработчика знания о цепочках прототипов [13].

Прототипное наследование, используемое в языке JavaScript, заключается в том, что экземпляры наследуются напрямую от других объектов, реализуются через фабрики или Object.create() и экземпляры могут быть составлены из множества различных объектов для упрощения выборочного наследования [14]. Считается, что прототипное наследование более и гибкое, нежели классовое, к которому возвращается CoffeeScript. Однако, оно признано более сложным для освоения начинающими разработчиками [8]. Кроме того, существует как минимум один способ реализовать прототипное наследование в языке CoffeeScript при возникновении такой необходимости.

CoffeeScript устраняет многие недостатки языка JavaScript:

* предоставляет более простой синтаксис с меньшим количеством балласта в виде сложных синтаксических конструкций, пришедших в JavaScript из разных языков программирования;
* использует пробелы как способ организации блоков кода;
* обеспечивает лаконичный синтаксис для выражения функций;
* обеспечивает классическое наследование на основе классов [9].

Код для реализации схожих программных конструкций на CoffeeScript занимает до двух раз меньше объёма, чем код на языке JavaScript [11].

То, что CoffeeScript транслируется в JavaScript и использует для компиляции Node.js [9], делает его потенциально эффективным инструментом для разработки клиентской части сложных интернет-ресурсов, в основе которых лежит клиент-серверная модель и взаимодействие с базами данных, а также при разработке систем для управления контентом. В случае с системами управления контентом вероятность, что созданный интернет-ресурс будет сопровождать разработчик, не являющийся автором используемой системы, стремится к 100% ввиду высокого распространения таких систем. Применение языка CoffeeScript упростит задачу анализа исходного кода интерфейса системы и предоставит более широкие возможности для совершенствования и кастомизации. Кроме того, программисту, знакомому с JavaScript, на изучение CoffeeScript понадобится не больше нескольких дней.

**Библиографический список:**

1. Бейтс М. CoffeeScript. Второе дыхание JavaScript. — Москва: ДМК Пресс, 2012. — 312 с.
2. Душкин Р.В. Справочник по языку Haskell. — Москва.: ДМК Пресс, 2008. — 540 с.
3. Erasmus M. Coffeescript Programming with Jquery, Rails, and Node.Js. — Birmingham, UK.: Packt Publishing, 2012. — 140 с.
4. Hatfield M. CoffeeScript Application Development Cookbook. — Birmingham, UK.: Packt Publishing, 2015. — 368 с.
5. Hudson A. Smashing CoffeeScript. — Indianapolis: Wiley, 2013. — 320 с.
6. Креншоу Д. Давайте создадим компилятор! — Москва: Наш мир, 2001. — 465 с.
7. Основные понятия по дисциплине «Функциональное программирование и интеллектуальные системы». — Екатеринбург: 2015. — 19 с.
8. CoffeeScript Руководство // w3ii.com. URL: http://www.w3ii.com/ru/coffeescript/coffeescript\_overview.html (дата обращения: 3.01.2018).
9. CoffeeScript // Национальная библиотека им. Н. Э. Баумана. URL: http://ru.bmstu.wiki/CoffeeScript (дата обращения: 3.01.2018).
10. Кто использует Coffeescript в своих проектах или на работе? // JavaScript.ru. URL: http://javascript.ru/forum/node-js-io-js/25218-kto-ispolzuet-coffeescript-v-svoikh-proektakh-ili-na-rabote.html (дата обращения: 3.01.2018).
11. CoffeeScript! // CoffeeScript на русском. URL: http://coffeescript2ru.github.io (дата обращения: 3.01.2018).
12. Почему никто не использует функциональные языки // SoftCraft. Разноликое программирование. URL: http://www.softcraft.ru/paradigm/fp/whynotfp/ (дата обращения: 3.01.2018).
13. Улучшенное наследование в CoffeeScript // Хабрахабр. URL: https://habrahabr.ru/post/190494/ (дата обращения: 3.01.2018).
14. 10 вопросов на собеседовании JavaScript программиста, ответить на которые должен каждый // proglib. Библиотека программиста. URL: https://proglib.io/p/10-important-javascript-interview-questions/ (дата обращения: 3.01.2018).
15. MySQL и Node.js — Подключение к базе данных // HACKERX. URL: https://hackerx.ru/mysql-node-js-connect-to-database/ (дата обращения: 3.01.2018).
16. 7 перспективных языков программирования // itProger. URL: https://itproger.com/news/40 (дата обращения: 3.01.2018).
17. Официальная документация // CoffeeScript. URL: http://coffeescript.org/ (дата обращения: 3.01.2018).

© А.Г. Проаспэт, 2018