Брюс Тейт

Семь языков за семь недель

Практическое руководство по изучению языков программирования

Под редакцией Жаклин Картер

2-е издание, электронное

ДМК ПРЕСС
издательство
Москва, 2023

УДК 004.6
ББК 32.973.26
Т30

Тейт, Брюс.

Т30 Семь языков за семь недель. Практическое руководство по изучению языков программирования / Б. А. Тейт ; пер. с англ. А. Н. Киселева. — 2-е изд., эл. — 1 файл pdf : 386 с. — Москва : ДМК Пресс, 2023. — Систем. требования: Adobe Reader XI либо Adobe Digital Editions 4.5 ; экран 10". — Текст : электронный.

ISBN 978-5-89818-315-8

Вместе с семью языками программирования вы исследуете наиболее важные из современных моделей программирования. Вы познакомитесь с динамической типизацией, которая делает языки Ruby, Python и Perl такими гибкими. Постигнете систему прототипов, лежащую в основе языка JavaScript. Увидите, как сопоставление с образцом в языке Prolog сказалось на формировании языков Scala и Erlang. Узнаете, чем функциональное программирование на языке Haskell отличается от программирования на языках семейства Lisp, включая Clojure.

Издание предназначено для программистов разной квалификации, в том числе выбирающих для изучения новый язык программирования.

УДК 004.6
ББК 32.973.26

Электронное издание на основе печатного издания: Семь языков за семь недель. Практическое руководство по изучению языков программирования / Б. А. Тейт ; пер. с англ. А. Н. Киселева. — Москва : ДМК Пресс, 2014. — 384 с. — ISBN 978-5-94074-539-6. — Текст : непосредственный.

ISBN 978-5-89818-315-8

Содержание

Посвящение

Пять месяцев, с декабря 2009-го по апрель 2010-го, выдались самыми тяжелыми в моей жизни. Сначала моему брату (которому не было и 47 лет) потребовалась срочная операция на сердце. При этом совершенно ничто не предвещало беды. (Операция прошла успешно, и он чувствует себя хорошо.) В конце марта у моей сестры был обнаружен рак молочной железы. Но самый большой шок я испытал в начале марта, когда моей матери был поставлен страшный диагноз: рак в последней стадии. После этого она прожила всего несколько недель.

Как вы уже поняли, мне пришлось пережить горечь утраты. Но, как ни странно, этот тяжелый опыт нельзя назвать полностью отрицательным. Дело в том, что моя матушка жила в ладу с собой и окружающими, ее отношения с семьей были крепкими и наполненными, и они строились именно так, как она того хотела, в соответствии с ее убеждениями и верованиями.

Линда Лейла Тейт (Lynda Lyle Tate) воплощала свою творческую энергию в рисунках акварелью. Она делилась своим искусством с окружающим миром с помощью галереи на Медисон-авеню и через своих учеников. До того как я покинул отчий дом, у меня была возможность взять у нее несколько уроков. Для меня, обладающего техническим складом ума, этот опыт давался с большим трудом. Я начинал воспроизводить свой очередной шедевр на белом листе. Но по мере того, как рисунок приобретал более отчетливые очертания, он становился все дальше от первоначального замысла. Когда я приходил в отчаяние от своей неспособности исправить картину, ко мне подходила мама, заглядывала через плечо и говорила – что она видит. Затем легкими движениями кисти добавляла несколько штрихов, подчеркивающих глубину и некоторые детали, и я с удивлением обнаруживал, что был совсем недалеко от желаемого! Достаточно было лишь нескольких талантливых прикосновений, чтобы уберечь мое творение от катастрофы. Тогда я вздымал руки в победном жесте и возвещал всему классу о своем творении, не замечая, что каждый ученик прошел через свой собственный триумф маленькой победы.

Спустя некоторое время я узнал, что мама работала и над другими холстами. Через ее церковь и ее профессию она находила людей, сломавшихся духом. Сталкиваясь с такими людьми то тут, то там, моя мама приводила их в класс, где с помощью красок и бумаги помогала им вновь обрести себя и найти свой путь. В последнюю неделю к ней приходило много людей, удрученных скорой потерей Учителя, и для

каждого мама находила удачную шутку или доброе слово, успокаивая тех, кто приехал успокаивать ее. Я встретился с холстами человеческих душ, подправленных рукой Мастера и вдохновленных на большие дела. Я был потрясен.

Когда я сообщил матери о своем желании посвятить ей эту книгу, она сказала, что ей будет приятно, но она ничего не смыслит в компьютерах. Это – правда. Даже сама мысль о Windows делала ее беспомощной. Но, мама, ты сделала для меня все возможное. Твои своевременные слова поддержки вдохновляли меня, твоя любовь к творчеству помогла мне сформироваться, а твой энтузиазм и жизнелюбие до сих пор сопровождают меня. Вспоминая о пережитых событиях, я понимаю, что был не в силах помочь, но мне намного легче от того, что я тоже ощущаю себя холстом, побывавшим под кистью Мастера.

Эта книга посвящается Линде Лейле Тейт (Lynda Lyle Tate), 1936–2010.

Благодарности

Это – самая сложная книга из всех, написанных мной. Она же оказалась самой полезной. Очень много людей помогали мне в ее создании. Прежде всего я хочу поблагодарить свою семью. Кайла (Kayla) и Джулия (Julia), ваша манера изложения мыслей удивляет меня. Вы пока не представляете, каких высот можете достигнуть! Мэгги (Maggie), ты – мой источник радости и вдохновения.

В сообществе пользователей Ruby спасибо Дейву Томасу (Dave Thomas), познакомившему меня с новым языком, который перевернул всю мою карьеру и помог опять познать радость работы. Спасибо также Мацу (Matz) за его дружбу и совет поделиться своими знаниями с читателями. Вы пригласили меня посетить Японию, где родился Ruby, и эта поездка вдохновила меня намного сильнее, чем вы могли себе представить. Спасибо Чарльзу Наттеру (Charles Nutter), Эвану Фениксу (Evan Phoenix) и Тиму Брею (Tim Bray) за помощь в подготовке этой темы, вошедшей в книгу, – возможно, это было утомительно для вас, но вы помогли мне сформировать этот раздел книги.

В сообществе пользователей Io спасибо Джереми Трегану (Jeremy Tregunna) за помощь в подготовке нескольких примеров на Io. Ваши отзывы были одними из самых ценных – они всегда были весьма кстати и очень помогли в создании главы об этом языке. Стив Декорт (Steve Dekorte), вы создали особый язык, и я надеюсь, что он когда-нибудь займет подобающее ему положение в мире программирования. Поддержка параллелизма в Io просто превосходна, и сам язык наполнен удивительной внутренней красотой. Я точно могу сказать, насколько хорош этот язык. Спасибо вам за помощь в его установке. Спасибо также за ваши вдумчивые отзывы, которые помогли мне ухватить суть Io. Вам удалось овладеть умами первых читателей и побудить их к использованию вашего языка.

В сообществе пользователей Prolog спасибо Брайану Тарбоксу (Brian Tarbox), что поделился своим богатым опытом с моими читателями. Проекты исследования дельфинов определенно придали красок главе, посвященной языку Prolog. Отдельное спасибо Джо Армстронгу (Joe Armstrong), чьи отзывы оказали существенное влияние на формирование главы и всей книги в целом. Спасибо вам также за пример раскрашивания карты и за ваши идеи – они оказались как нельзя к месту.

В сообществе пользователей Scala спасибо моему доброму другу Венкату Субраманиаму (Venkat Subramaniam). Ваша книга о языке

Scala написана понятным языком и богата отличными примерами. Я в значительной степени опирался на эту книгу. Я высоко ценю вашу помощь, которую вы оказывали мне. Она помогла мне избежать значительных трудностей и сосредоточиться на обучении. Спасибо также Мартину Одерски (Martin Odersky), что поделился своими мыслями с читателями. Язык Scala являет собой удивительное сочетание функциональной и объектно-ориентированной парадигм программирования. Ваши знания в освещении этого языка очень пригодились мне.

В сообществе пользователей Erlang я еще раз хочу поблагодарить Джо Армстронга (Joe Armstrong). Ваши доброжелательность и энергия помогли мне облечь идеи в правильные словесные формы. Ваши предложения по построению отказоустойчивых распределенных систем действительно работают. Философия языка Erlang «позвольте приложению потерпеть неудачу» («Let it crash») пришлась мне по душе больше, чем философии в других языках, представленных в этой книге. Я надеюсь, что эти идеи получат более широкое распространение.

В сообществе пользователей Clojure спасибо Стюарту Халлоуэю (Stuart Halloway) за его отзывы и идеи, заставившие меня работать над книгой с еще большим усердием, чтобы сделать ее более полезной для моих читателей. Ваши глубокие знания Clojure и интуиция помогли мне понять, что является наиболее важным. Ваша книга также оказала существенное влияние на главу о языке Clojure и фактически изменила мои подходы к решению некоторых задач в других главах. Ваши подходы к обучению, наработанные за многолетнюю практику, пользуются заслуженной славой. Спасибо также Ричу Хикки (Rich Hickey) за освещение идей, легших в основу его языка, и за лекцию по теме: «Что значит быть диалектом языка Lisp». Некоторые идеи языка Clojure кажутся революционными и вместе с тем являются абсолютно практичными. Поздравляю – вы нашли способ провести революцию в Lisp. Уже в который раз.

В сообществе пользователей Haskell спасибо Филиппу Вадлеру (Phillip Wadler) за возможность заглянуть внутрь процесса создания языка Haskell. Мы оба питаем страсть к обучению других людей, и у вас это здорово получается. Спасибо также Саймону Пейтону-Джонсу (Simon Peyton-Jones). Мне было очень приятно общаться с вами, я постарался донести до читателей ваш опыт и ваши знания.

Существенный вклад в создание этой книги внесли рецензенты. Спасибо Владимиру Г. Ивановичу (Vladimir G. Ivanovic), Крей-

гу Рикки (Craig Riecke), Полу Бучеру (Paul Butcher), Фреду Дауду (Fred Daoud), Аарону Бедрею (Aaron Bedra), Давиду Эйзингеру (David Eisinger), Антонио Каньяно (Antonio Cangiano) и Брайану Тарбоксу (Brian Tarbox). Вы сформировали самую эффективную команду рецензентов, с которыми я когда-либо работал. Книги – намного более сложный продукт, чем думают многие. Я знаю, что рецензирование на таком уровне детализации – это неблагодарный и тяжелый труд. Спасибо всем вам, кто способствует выходу технических книг. Без вашего участия издательский бизнес просто прекратил бы свое существование.

Я также хочу поблагодарить тех, кто делился со мной идеями в выборе языков и философий программирования. Не раз важными мыслями со мной делились Нил Форд (Neal Ford), Джон Хайнц (John Heintz), Майк Перхам (Mike Perham) и Ян Воршак (Ian Warshak). Общение с вами помогло мне выглядеть более умным, чем я есть на самом деле.

Хочу также поблагодарить первых читателей рукописи книги и подталкивавших меня к продолжению работы над ней. Ваши комментарии показали, что многие из вас проявили немалое внимание при чтении. Я внес изменения в книгу, опираясь на сотни ваших комментариев, и могу обещать, что продолжу учитывать ваши пожелания в следующих ее изданиях.

Наконец, выражаю самую искреннюю благодарность коллективу издательства Pragmatic Bookshelf. Дейв Томас (Dave Thomas) и Энди Хант (Andy Hunt), вы неоднократно оказывали важное влияние на мою карьеру, сначала как программиста, а потом как автора. Это издательство сделало мой труд как писателя ненапрасным. Благодаря вам книги, не предназначенные для широкой публики, такие как эта, обретают вполне определенную материальную ценность. Спасибо всему коллективу издательства. Джеки Картер, ваши нежные руки и твердое руководство были тем, что необходимо для этой книги, и я надеюсь, что наше общение было для вас таким же приятным, как для меня. Спасибо всем, кто трудился над этой книгой, стараясь сделать ее еще лучше. Особую благодарность я хочу выразить редакторам и корректорам, улучшавшим оформление книги и исправлявшим мои ошибки, и в их числе: литературному редактору Киму Вимпсетту (Kim Wimpsett); составителю алфавитного указателя Сету Мейслину (Seth Maislin); наборщику Стиву Питеру (Steve Peter) и продюсеру Джанет Фарлоу (Janet Furlow). Без вас эта книга стала бы такой, какая она есть сейчас.

Как обычно, в книге все еще могут оставаться ошибки, не замеченные этой великолепной командой. Всем, кому они встретятся, я заранее приношу свои самые искренние извинения. Любые оплошности я допускал непреднамеренно.

Наконец, спасибо всем вам, мои читатели. Я считаю, что бумажные книги имеют определенную ценность, и я могу следовать своему увлечению и писать для вас.

Брюс Тейт (Bruce Tate)

Предисловие

Из еще не написанной книги «How Proust Can Make You a Better Programmer»

Джо Армстронг (Joe Armstrong), создатель языка Erlang

– Редактор Gmail неправильно воспринимает типографские кавычки.

– Печально, – сказала Марджери, – это признак безграмотности программиста и упадка культуры.

– Что будем делать с этим?

– Мы должны настоять, чтобы следующий программист, которого наймем, обязательно прочитал роман «A la recherche du temps perdu»[1].

– Все семь томов?

– Все семь томов.

– Это поможет ему лучше узнать правила пунктуации и правильно интерпретировать кавычки?

– Необязательно, но этот роман сделает его лучшим программистом, потому что в нем он найдет многие секреты мастерства...

Обучение программированию сродни обучению плаванию. Никакая теория не заменит практических занятий, когда обучаемый молотит руками и ногами по воде, судорожно хватая ртом воздух. Впервые погрузившись в воду с головой, вы паникуете, но когда вы качаетесь на поверхности, неторопливо вдыхая воздух, вы чувствуете легкий восторг. Вы думаете: «Я могу плавать!» По крайней мере, именно эта мысль пришла мне в голову, когда я научился плавать.

То же самое происходит, когда вы обучаетесь программированию. Первые шаги даются с большим трудом, и вам нужен хороший тренер, который поможет вам прыгнуть в воду.

Брюс Тейт (Bruce Tate) – именно такой тренер. Эта книга поможет вам сделать самый первый и трудный шаг в обучении программированию.

Предположим, что вы миновали первый сложный этап загрузки и установки интерпретатора или компилятора языка, интересного вам. Что делать дальше? Какую программу написать первой?

---

[1] Пруст М. В поисках утраченного времени. – М.: АСТ, 2011. ISBN: 978-5-17-067438-1. – *Прим. перев.*

Брюс четко и обстоятельно отвечает на этот вопрос. Просто вводите программы и их фрагменты, предлагаемые в книге, и наблюдайте получаемые результаты. Не думайте пока о создании собственной программы – просто пробуйте воспроизводить примеры из книги. Набравшись опыта, вы сможете приступить к созданию собственного проекта.

Первый шаг в обретении новых знаний заключается не в попытке создать что-то свое, а в повторении того, что было уже создано другими. Это самый быстрый способ овладеть навыками программирования.

Первые шаги в изучении нового языка программирования обычно заключаются не в глубоком проникновении в его философию, а в знакомстве с особенностями расстановки точек с запятой и запятых и исследовании сообщений, которые выдает система, когда вы допускаете ошибки. И только когда вы преодолеете первый этап, научившись вводить программы без ошибок и компилировать их, можно начинать задумываться о назначении различных языковых конструкций.

Пройдя первый этап механистического ввода программы и ее запуска, можно откинуться на спинку кресла и расслабиться. Все остальное за вас сделает ваше подсознание. В то время как сознание будет запоминать правила расстановки точек с запятой, подсознание будет стараться проникнуть в суть программных конструкций. И однажды вы обнаружите, что понимаете логику программы и роль данных конструкций в данном языке.

Поверхностное знакомство со многими языками весьма полезно. Я часто обнаруживаю, что знакомство с Python или Ruby помогает мне решить конкретную задачу. Программы, которые я загружаю из Интернета, часто написаны на разных языках, и мне нередко требуется внести в них небольшие изменения перед использованием.

Каждый язык имеет свой набор идиом, свои достоинства и недостатки. Изучив несколько разных языков программирования, вы сможете выбирать тот язык, который лучше подходит для решения текущей задачи.

Мне приятно видеть, что Брюс обладает разнообразием вкусов в отношении языков программирования. Он охватывает не только широко известные языки, такие как Ruby, но также менее распространенные и недооцененные, такие как Io. В конечном счете программирование – это понимание, а понимание – это познание идей. Соответственно, открытость новым идеям дает более глубокое понимание программирования вообще.

Гуру может сказать, что для более глубокого познания математики необходимо изучить латынь. Так же и с программированием. Чтобы полнее понимать объектно-ориентированное программирование, следует изучить логическое или функциональное программирование. Чтобы полнее понимать функциональное программирование, следует изучить программирование на языке Ассемблера.

Когда я еще только учился программированию, большой популярностью пользовались книги, проводящие сравнительный анализ разных языков программирования, но большинство из них были чересчур академичны и почти не несли практических рекомендаций по использованию сравниваемых языков. Такой подход был характерен для того времени. Вы могли многое узнать об идеях, положенных в основу языка, но не имели возможности опробовать их на практике.

В настоящее время мы можем не только знакомиться с идеями, но и пытаться применять их на практике. Это большая разница, как между стоянием на краю бассейна с мыслью «а хорошо ли мне будет в воде» и получением удовольствия от плавания.

Я настоятельно рекомендую прочитать эту книгу и надеюсь, что вам она понравится так же, как и мне.

Джо Армстронг (Joe Armstrong),
создатель языка Erlang
2 марта 2010
Стокгольм

Глава 1

Введение

Люди изучают иностранные языки по самым разным причинам. Первый язык изучается, чтобы жить. Знание родного языка помогает общаться с окружающими. Побудительные мотивы к изучению второго языка у разных людей могут быть разными. Кому-то знание иностранного языка может пригодиться для построения карьеры или для адаптации в новом окружении при смене места жительства. Но иногда решение изучить новый язык принимается не потому, что он необходим, а просто ради желания учиться. Знакомство со вторым языком может расширить кругозор. Кому-то знание каждого нового языка помогает сформировать новый способ мышления.

То же относится и к языкам программирования. В этой книге я представлю вам семь разных языков. Моя цель вовсе не в том, чтобы заставить вас, как это делают многие мамы, заставляя своих чад выпить утром ложку рыбьего жира. Я хочу пригласить вас в увлекательное путешествие, которое изменит ваши взгляды на программирование. Я не буду делать из вас экспертов, но я расскажу вам чуть больше, чем требуется, чтобы написать программу «Hello, World».

1.1. Логика описания

Часто, приступая к изучению нового языка или фреймворка, я стараюсь найти в Интернете интерактивное руководство, чтобы опробовать возможности языка в управляемом окружении. Я могу написать свой сценарий, чтобы заняться дальнейшими исследованиями, но обычно я ищу информацию, которая быстро разбудит во мне интерес, пример синтаксического сахара и описание базовых концепций.

Однако обычно для меня этого недостаточно. Если я встретил новый язык, являющийся более чем тонкой оберткой вокруг уже известного мне языка, мне потребуется более глубокое его исследование. Эта книга даст вам такую возможность целых семь раз. Здесь вы найдете ответы на следующие вопросы: