самоучитель

Айк Харазян

Язык Swift

Простые типы данных в Swift
Операторы и типы коллекций
Условные конструкции и циклы
Функции и замыкания
Перечисления и кортежи
Классы и структуры
ООП в Swift
Расширения и протоколы
Обобщенные типы

bhv®

Айк Харазян

самоучитель

Язык Swift

Санкт-Петербург
«БХВ-Петербург»
2016

УДК 004.438 Swift
ББК 32.973.26-018.1
Х20

Харазян А. А.

Х20 Язык Swift. Самоучитель. — СПб.: БХВ-Петербург, 2016. — 176 с.: ил. — (Самоучитель)

ISBN 978-5-9775-3572-4

Книга предназначена для самостоятельного изучения Swift — нового языка программирования для iOS и OS X. Описана версия Swift 2.0. Материал построен по принципу от более легкого к сложному, изложение сопровождается большим количеством листингов кода, для тестирования и отладки используется новая среда быстрой разработки Playground. Объяснены основы Swift, синтаксис языка и его особенности. Описаны типы данных, условные выражения, циклы, массивы, функции, кортежи, базовые операторы и другие стандартные конструкции. Кратко даны основы объектно-ориентированного программирования. Подробно рассмотрены более сложные или специфические для Swift конструкции: перечисления, замыкания, опциональные типы, классы, структуры, встроенные и обобщенные типы, расширения, протоколы, расширенные операторы и др.

Для программистов

УДК 004.438 Swift
ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Евгений Рыбаков*
Зав. редакцией	*Екатерина Капалыгина*
Редактор	*Григорий Добин*
Компьютерная верстка	*Ольги Сергиенко*
Корректор	*Зинаида Дмитриева*
Дизайн обложки	*Марины Дамбиевой*

Подписано в печать 31.07.15.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 14,19.
Тираж 1000 экз. Заказ №
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

Первая Академическая типография "Наука"
199034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-5-9775-3572-4

Оглавление

Введение

2 июня 2014 года на ежегодной конференции разработчиков WWDC компания Apple представила новый язык программирования Swift. С его помощью можно создавать приложения для iOS и OS X с еще большей легкостью, чем ранее. Swift сочетает в себе производительность и эффективность компилируемых языков программирования с простотой и интерактивностью популярных скриптовых языков. Он был вдохновлен такими языками программирования, как JavaScript, Python и Ruby.

При создании языка Swift разработчики руководствовались тремя парадигмами:

- ❒ быстрота;
- ❒ современность;
- ❒ надежность.

По словам Apple, некоторые алгоритмы, написанные на Swift, выполняются в разы быстрее, чем те же алгоритмы на Python. Этим она утверждает, что код, написанный на Swift, не только лаконичен и легко читаем, но и может конкурировать по скорости с мощными скриптовыми языками. Современность языка демонстрируют особенности, перенятые из скриптовых языков программирования, — такие как интерактивность и простота восприятия. Разработчики очень тщательно подошли к реализации возможностей, которые помогают противостоять появлению ошибок, тем самым делая код более надежным. Например, переменные всегда должны быть объявлены перед их использованием, а массивы и целые числа проверяются на перегрузку. Память очищается автоматически, а технология опциональных типов позволяет исключить возможные ошибки с отсутствием значения.

Еще одной важной особенностью Swift являются песочницы Xcode Playground, которые позволяют тестировать код и видеть результат его выполнения в реальном времени. До их появления приходилось экспериментировать на кусках кода внутри целых проектов.

Для кого предназначена эта книга?

Необходимость изучения нового языка программирования может сильно пугать новичков. Но эта книга прекрасно подойдет даже тем, кто никогда не использовал языки программирования C, C++ или Objective-C. И хотя книга не рассчитана на тех, кто знакомится с языками программирования впервые, но, все же, в начале каждой главы даны краткие объяснения основ программирования, касающиеся освещаемой темы.

Опытных же разработчиков в первую очередь заинтересуют особенности и новшества языка Swift, которым уделено особое внимание, а также материалы второй части книги по углубленному изучению программирования на Swift.

Изложение каждой новой темы в книге сопровождается подробными примерами на языке Swift. А сами главы построены по принципу последовательного изучения материала от более легкого к более сложному. Это облегчает поиск нужных глав для опытных разработчиков и помогает новичкам изучать Swift постепенно.

Какова польза от книги?

В книге рассмотрены все особенности языка Swift, которые в будущем помогут вам при разработке приложений и игр под iOS и OS X. Изучив материал книги, вы сможете полностью разбираться в коде, написанном на Swift, а также самостоятельно разрабатывать алгоритмы на этом языке. Знания, изложенные в книге, лежат в основе фреймворков Cocoa и Cocoa Touch, служащих для разработки приложений и игр под iPhone, iPad, Mac и т. д. Так что, изучив техники языка программирования Swift, вы смело можете приступить к изучению API и фреймворков и разработке приложений для iOS и OS X.

Структура книги

Книга состоит из двух больших частей.

- **Часть I. Основы Swift.** Первая часть книги знакомит читателя с тем, как реализованы в Swift стандартные структуры вроде циклов, условий и массивов. Она будет полезна для начинающих разработчиков, поскольку расскажет им об основах программирования. А опытным разработчикам эта часть даст понять, какие особенности есть в простых структурах Swift в отличие от других языков.
- **Часть II. Углубленное изучение Swift.** Вторая часть книги описывает более сложные структуры и понятия, которые не встречаются в других языках программирования. Она будет полезна для опытных разработчиков, которые хотят углубиться в разработку на Swift более сложных приложений. Для новичков же важно сначала изучить главы из первой части, и только затем приступить ко второй.

Тестируйте листинги кода!

Каждая глава книги сопровождается большим количеством листингов кода. Они позволяют лучше понять некоторые алгоритмы и операции. Мы рекомендуем вам по ходу изучения материала набивать предлагаемые программные коды непосредственно вручную. Так вы легче запомните синтаксис языка Swift, поскольку будете учиться на своих ошибках.

А помогут вам в этом уже упомянутые песочницы Playground — новый тип документа Xcode, который появился в его шестой версии, вместе с языком Swift. Playground представляет собой самостоятельный файл, куда можно вписать несколько строк кода, которые будут тут же выполнены с выводом полученного результата. Более подробно о Playground рассказано в *главе 1*.

Программа Apple для разработчиков

Чтобы получать последнее программное обеспечение для разработчиков и иметь возможность публиковать свои приложения в магазине цифровой дистрибьюции AppStore, вы, вероятно, захотите стать членом Apple Developer Programs. Найти подробную информацию о привилегиях, предоставляемых вступившим в эту программу разработчикам, вы можете на странице сайта Apple по адресу: **https://developer.apple.com/programs**.

ЧАСТЬ I

Основы Swift

В первой части этой книги мы познакомимся с простыми элементами Swift, которые встречаются в большинстве языков программирования. Но сначала мы узнаем, как Swift появился на свет, чем разработчики пользовались до него и с чего следует начать с ним знакомство.

ГЛАВА 1

Swift: как он появился и с чего начать?

1.1. Как появился Swift?

Язык для человека является универсальным средством коммуникации и обмена информацией. С его помощью мы можем высказать наши мысли друзьям, членам семьи или коллегам по работе. И компьютерные языки программирования в этом плане также не являются исключением. Принцип выражения своих мыслей в них тот же. Только на этот раз мы высказываем их компьютеру.

Языки программирования, подобно человеческим языкам, существуют в разных формах и со временем эволюционируют и совершенствуются. Главной задачей языков программирования изначально являлась необходимость взаимодействовать с инструкциями компьютера, чтобы задать набор определенных действий, которые мы хотим от него получить.

Опытные пионеры компьютерной эпохи знают, что центральный процессор компьютера понимает только набор инструкций, состоящий из нулей и единиц. Чтобы разработчики могли разрабатывать свой код на более понятном им языке, чем нули и единицы, существуют программы-компиляторы, которые переводят код, написанный на определенном языке программирования, в инструкции для процессора, состоящие из тех самых нулей и единиц.

Компромисс между возможностями языка программирования и его производительностью обеспечили такие языки программирования, как C и C++. Но пока языки C и C++ распространялись на все большее количество платформ, Apple решила пойти иным путем и разработала язык Objective-C. Построенный на основе C, Objective-C сочетал в себе мощь широко используемого языка программирования с объектно-ориентированными методологиями. На протяжении многих лет Objective-C являлся основой для разработки программ для компьютеров Macintosh и iOS-устройств.

Однако, несмотря на производительность и простоту, Objective-C носил с собой весь багаж, наследованный от C. Для разработчиков, которые разбираются в C, это не вызывает трудностей. Но новички жаловались, что Objective-C труден для понимания и разработки.