

АНТОН РАЗГУЛЯЕВ

SWIFT?
НЕ ВОПРОС



Издательство АСТ
Москва

УДК 004.43

ББК 32.973

P17

Разгуляев, Антон.

P17 Swift? Не вопрос/ Разгуляев, А. — Москва: Издательство АСТ, 2024. — 448 с. — (Программирование для всех)

ISBN 978–5–17–165224–1

SWIFT — собственный язык программирования Apple, представленный в 2014 году. За 10 лет эволюции Swift стал полностью интегрирован во все устройства Apple. На нем написаны LinkedIn, Airbnb, Kickstarter, Slack, Firefox, Uber и многие другие топовые приложения. Все iOS-версии кроссплатформенных мессенджеров и социальных сетей, которыми мы пользуемся, написаны с использованием Swift.

ЕСЛИ ПРОФЕССИЯ — ТО ПРОГРАММИСТ

ЕСЛИ ЯЗЫК — ТО SWIFT

Антон Разгуляев — опытный разработчик, издатель собственных приложений в App Store, энтузиаст и ментор, собрал самое важное и актуальное для изучения Swift и SwiftUI с нуля.

Книга «SWIFT? НЕ ВОПРОС!» проведет вас от самых основ программирования до написания настоящего приложения. Вы не просто изучите теорию, но и на каждом шаге будете выполнять практические задачи, которые приведены в каждой главе вместе с решениями для самопроверки.

УДК 004.43

ББК 32.973

ISBN 978–5–17–165224–1

© Разгуляев А., текст

© ООО Издательство «АСТ»

Введение

Я хорошо помню день, когда я решил начать изучать программирование: я как раз продал свой небольшой бизнес, на основной работе загрузка тоже почти полностью отсутствовала, было свободное время и я решил найти себе интересное, перспективное и независимое от обстоятельств занятие. У меня уже был опыт самостоятельного изучения различных специализированных программ по книгам (Photoshop, Final Cut) и я решил начать именно с покупки пары книг, чтобы познакомиться с темой, получить какое-то первое впечатление и понять, в каком направлении двигаться дальше.

После работы я решил зайти в Буквоед (большой, трехэтажный, там точно должно было быть то, что мне нужно). По пути я позвонил своему старому хорошему другу, он пару лет назад начал изучать программирование, потом резко оставил свою должность менеджера по закупкам и уже работал по новой специальности и при этом явно зарабатывал. Мой друг уверенно и без колебаний заверил меня, что начинать нужно с языка **Python**, что это самый простой, доступный и популярный язык программирования. Мой друг был почти прав.



Тем не менее, меня уже интересовал язык **Swift**. Я знал две вещи:

1. на этом и только на этом языке пишутся приложения на все **iOS, macOS, iPadOs, watchOs** и **tvOs**;
2. рынок мобильных приложений растет и будет расти еще много лет.

Для меня было абсолютно очевидно, что продвигаемая по всем каналам мечта о фрилансере–разработчике, который работает сам на себя, работает откуда хочет и сколько хочет — это в первую очередь именно **Swift**.

Итак, я пришел в Буквоед, я был уверен, что полки отдела программирования будут завалены книгами про **Swift** с многообещающими названиями типа «стать программистом за 10 дней». Как я был удивлен, когда я не увидел ни одной книги про **Swift**...

А ведь Swift — это самый молодой и перспективный язык. Его первая версия была представлена всего–то в 2014 году (для сравнения исходный текст Python был опубликован в 1991 году, C++ — в 1983 году, Java — в 1995–ом). Ему еще нет и 10 лет! При этом он впитал все самое лучшее из других языков, часть функций выполняется быстрее по сравнению с другими языками.

Сами Apple на презентации языка показали вот такой слайд, который наглядно демонстрирует, какое место занимает Swift по сравнению с другими языками, если рассматривать производительность получаемых программ и продуктивность разработки:



«Выигрывая в удобстве написания кода — проигрываешь в надежности программы» — это правило справедливо для всех языков, кроме Swift!

За прошедшие годы Apple создала отличную инфраструктуру для начинающих программистов, и то что в России на момент написания этой книги так мало приверженцев Swift, это создаёт отличную нишу для нас с вами.

В общем, по прошествии времени, когда я изучил и Swift и Python, я решил записать «короткий путь» среди огромных дебрей изученной информации, и сохранить его как раз в виде этой книги.

Кто я такой? Я не академик, не профессор, я не преподаю в Университете, я не основатель крупной и успешной it-компании... Я просто тот парень, кто уже сделал несколько шагов в программировании, перечитал кучу книг, прошел множество курсов. И вот пока я не оторвался от реальной жизни человека, который только начинает что-то изучать, не стал замороженным специалистом на своей какой-то волне, пока я помню собственные ощущения, я решил записать лучший опыт восприятия языка с позиции простого ученика.



В этой книге вы не найдете тонны душной теории, которую, когда слушаешь или читаешь, вроде бы все понятно, но вот дочитал, закрыл и ничего не осталось в голове, кроме желания заняться чем-то другим.

А что будет в этой книге?

Мы будем двигаться очень маленькими шажками, чтобы ничего не прошло мимо, но в тоже время с быстрым темпом:

Сначала мы разберемся с необходимым ПО.

Скачаем, установим, запустим

и создадим тестовую среду, в которой будем учиться.

Далее, начав с самых основ, мы будем изучать язык. После минимальных объяснений начнем сразу же решать задачи, то есть

сразу начнем писать код.

Мы будем экспериментировать, сразу смотреть результат, анализировать ошибки, с самого начала мы

будем мыслить как программист.

Погнали!

Часть 1

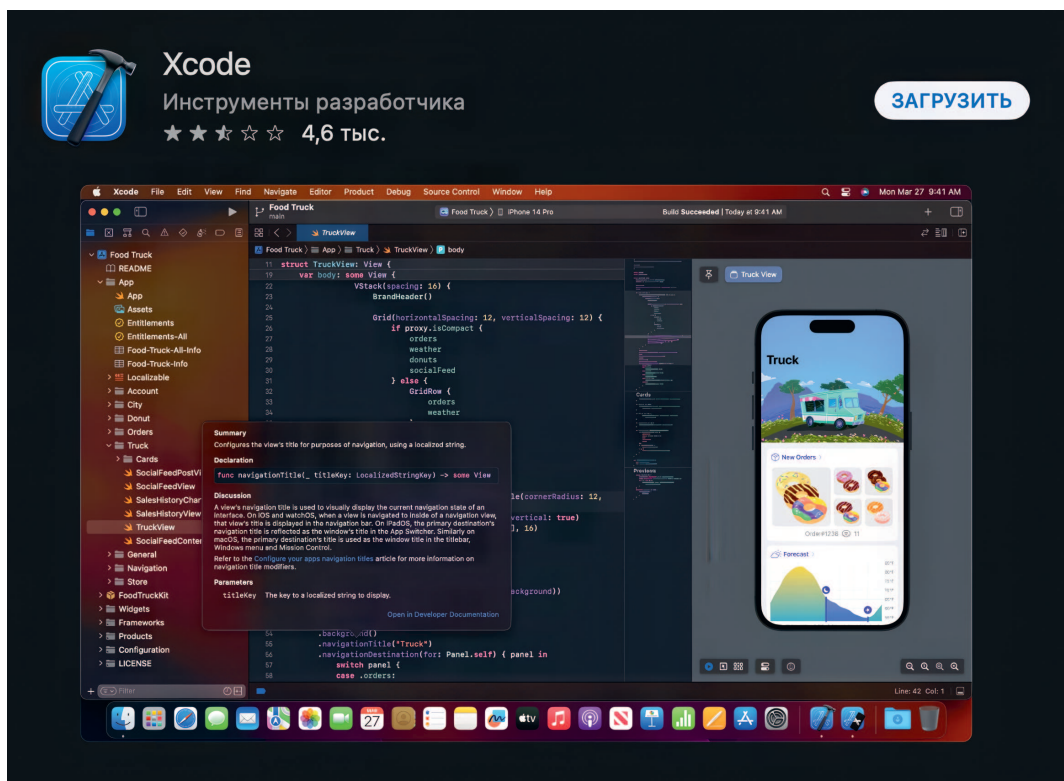
База

Как загрузить SWIFT

Swift — это название языка программирования, а само написание кода осуществляется в программе Xcode, Xcode — это основная и единственная программа для написания кода на языке Swift.

Разработчиком Xcode является компания Apple, поэтому чтобы её скачать достаточно просто открыть App Store, ввести в поиск Xcode и нажать загрузить. Программа бесплатная, никаких подписок также не требуется.

Плохая новость для обладателей компьютеров на Windows — к сожалению, на сегодняшний момент (по крайней мере на момент написания этой книги) Xcode доступен только на Mac OS.



Не обращайте внимание на низкий рейтинг, Xcode совсем не так уж плох, даже более того, в его арсенале очень много встроенных инструментов и помощников, которых нет ни в одной другой программе для других языков (в одной программе сразу точно нет). Действительно, на сложных проектах программа иногда подлагивает и зависает, что, конечно, не оправдывает ожиданий от такого разработчика, как Apple. Ожиданий, конечно, завышенных.

Запуск Xcode

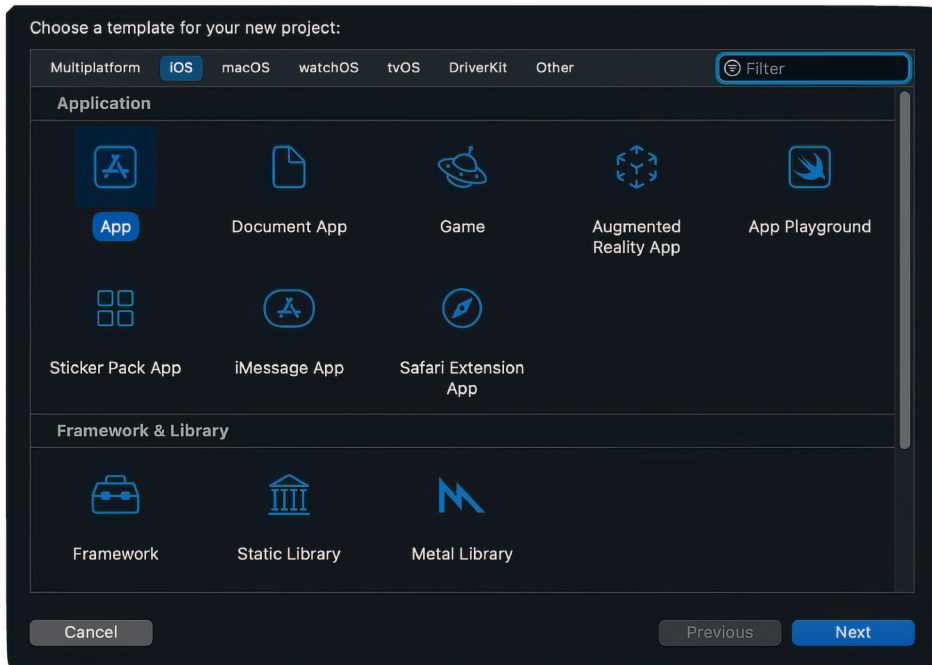
Стартовое окно при запуске Xcode выглядит следующим образом:



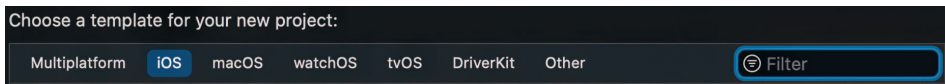
Здесь всё просто: первая строчка меню — это создание нового проекта, вторая и третья — это работа с уже существующими. Справа будут отображаться недавно открытые проекты (при первом запуске программы, естественно, там будет пусто).

Нам нужна первая строчка **Create a new Xcode project** — нажимаем её.

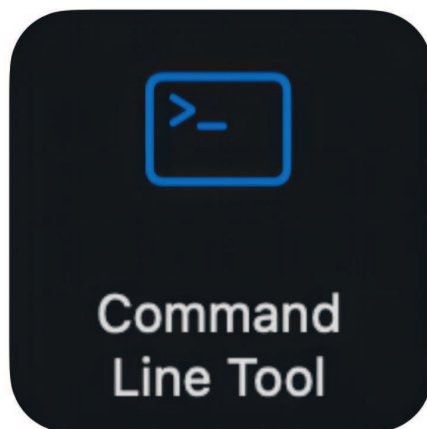
Дальше уже интереснее, открывается меню выбора шаблона проекта:



Обратите внимание, что в верхней части окна выбор типа устройства, под которое будет разрабатываться проект:



Не будем вдаваться в детали описания всех предлагаемых типов и начнем с типа Command Line Tool (он во вкладке macOS).



DICTIONARY

welcome	–	добро пожаловать
create	–	создать
new	–	новый
application	–	приложение
project	–	проект
clone	–	клонировать, сделать копию
file	–	файл, документ
something	–	что-то
exist	–	существовать
choose	–	выбирать
template	–	шаблон
game	–	игра
augmented reality	–	дополненная реальность
extension	–	расширение
framework	–	среда разработки
cancel	–	отмена
previous	–	предыдущий
next	–	следующий
line	–	строка
command line	–	командная строка
tool	–	инструмент
read	–	читать
options	–	меню
identifier	–	идентификатор
bundle	–	связывать
bundle identifier	–	уникальный идентификатор вашего приложения
constant	–	константа
account	–	учетная запись
language	–	язык
repository	–	хранилище
Git repository	–	сайт-хранилище для общего доступа к программам

Playground

Вы наверняка слышали слово «песочница» применительно к области программирования. Вот какое определение даёт Википедия¹:

Песочница — специально выделенная (изолированная) среда для безопасного исполнения компьютерных программ. Обычно представляет собой жёстко контролируемый набор ресурсов для исполнения гостевой программы — например, место на диске или в памяти. Доступ к сети, возможность общаться с главной операционной системой или считывать информацию с устройств ввода обычно либо частично эмулируют, либо сильно ограничивают. Песочницы представляют собой пример виртуализации.

Повышенная безопасность исполнения кода в песочнице зачастую связана с большой нагрузкой на систему — именно поэтому некоторые виды песочниц используют только для неотлаженного или подозрительного кода.

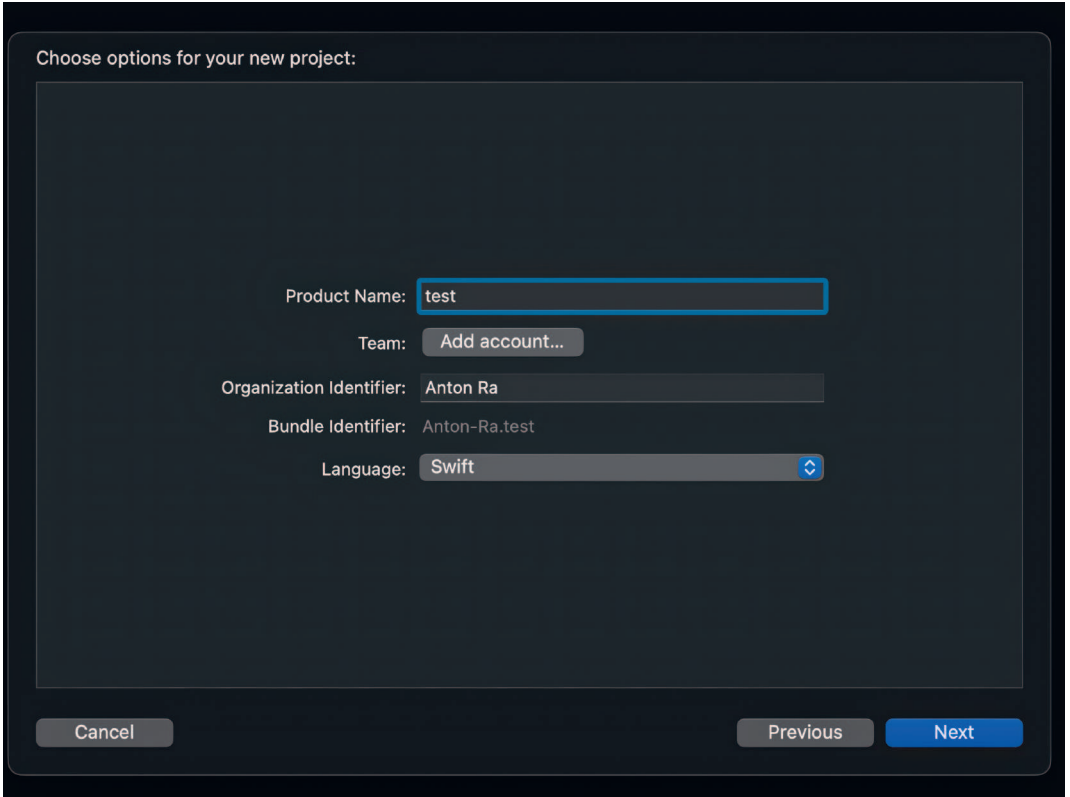
Так вот, Playground — это песочница.

У Apple есть отдельное приложение Playground (на macOS и iPadOS), но мы не будем им пользоваться, так как у него есть существенный недостаток: в Playground не работает встроенная функция ввода информации из консоли (`readLine()`). Я абсолютно на 100% уверен, что для правильного построения понимания программирования эта функция №1 для обучения.

Поэтому нашим Плейграундом будет Command Line Tool.

¹ [https://ru.wikipedia.org/wiki/Песочница_\(безопасность\)](https://ru.wikipedia.org/wiki/Песочница_(безопасность))

Итак, выбрав **Command Line Tool** откроется окно создания проекта:



Choose options for your new project:

Product Name: test

Team: Add account...

Organization Identifier: Anton Ra

Bundle Identifier: Anton-Ra.test

Language: Swift

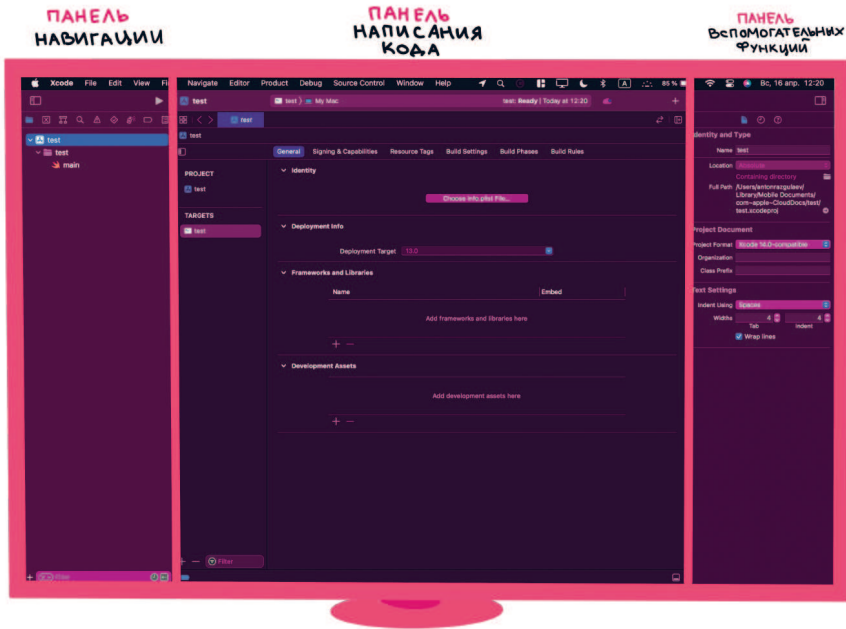
Cancel Previous Next

Назовите ваш тестовый проект как вам удобно. Тему последующей публикации наших приложений пока обсуждать не будем, хотя до этого совсем не так далеко, как сейчас кажется.

Далее сохраните ваш проект, где вам удобно. Для начала, я бы порекомендовал сохранять на рабочем столе, чтобы потом можно было быстро удалять всякие тестовые файлы. Уберите галочку *Create Git repository on my Mac*, это тоже связано с публикацией, сейчас этим заниматься не будем.

Открылся наш проект, давайте кратко рассмотрим интерфейс, который состоит из трех частей:

- панель навигации (Navigator),
- панель написания кода,
- панель вспомогательных функций (Inspector).



На рисунке выше на главной центральной панели находятся свойства нашего приложения. Отложим их изучение на потом и перейдем непосредственно к написанию кода, для этого перейдите в панели навигации в файл **main**:

