

Начинать создавать мобильные приложения под iOS можно с изучения популярного языка программирования Swift. Разработчики Apple трудились несколько лет, чтобы создать простой, безопасный и удобный язык.

Swift вобрал в себя лучшие идеи современных языков с мудростью инженерной культуры Apple. Компилятор оптимизирован для производительности, а язык оптимизирован для разработки, без компромиссов с одной или другой стороны.

Swift дружелюбен по отношению к новичкам в программировании. Это первый язык программирования промышленного качества, который также понятен и увлекателен, как скриптовый язык. Написание кода в песочнице позволяет экспериментировать с кодом Swift и видеть результат мгновенно, без необходимости компилировать и запускать приложение.

Ну что ж, вперед знакомиться с миром яблочных технологий!

ОБЩЕЕ

1. Знакомство с языком стоит начать с внимательного изучения [официальной документации от Apple](#) (или ее [хорошего перевода](#)).

Особое внимание обрати на:

- управление и работу с памятью;
- классы и структуры (особенности создания и использования);
- опционалы и работу с ними, что это и зачем;
- типы коллекций, в чем разница и особенности использования;
- функции и замыкания.



Прочие особенности Swift не менее важны для полноценного программирования под iOS и их понимание будет плюсом для прохождения обучения.

2. Обязательно закрепляй полученные знания практикой и экспериментами в [Swift Playgrounds](#).

BEST PRACTICES AND BASICS

1. Начинающим разработчикам важно хорошо разобраться со всеми [основными принципами ООП](#).
2. Будет большим плюсом понять как можно больше [паттернов проектирования](#).
3. Знание [SOLID](#) также будет плюсом.
4. Каждый программист в своей работе сталкивается с многопоточностью и конкурентным выполнением. Поэтому показателем технической грамотности будет знание основных концепций из этой области, которые изложены в статье [“Multithreading and Concurrency Fundamentals”](#).



Не ограничивайся приведенными ссылками: ищи разные источники, чтобы разобрать подходы с разных сторон и на разных примерах. Гугли или придумывай себе задачи в любой предметной области, пробуй решить их с использованием изученных принципов.

UI-КОМПОНЕНТЫ

Важно разобрать работу какого-то UI-ного компонента, например UITableView, о них есть множество уроков, например, как этот: [“Adding a UITableView programmatically”](#).

При этом важно понимать:

- как отрисовывать ячейки в таблице;
- как они переиспользуются;
- какие обязательные методы необходимо реализовать;
- в чем удобство и польза такого механизма;
- жизненный цикл приложения, с какого состояния в какое переходит;
- жизненный цикл UIViewController;
- разница UIWindow от UIView.



Полезно будет пробовать всё на собственном маленьком тестовом приложении, чтобы потом иметь возможность показать код и рассказать почему применялись те или иные решения.

ХОЧЕШЬ УЗНАТЬ БОЛЬШЕ?

- Подписывайся на наш [Telegram-канал](#) и всегда оставайся в курсе новых вакансий для iOS Developer.
- Также приглашаем тебя пройти бесплатную [программу обучения](#) по направлению iOS в NIX Solutions. Набор в группу открыт!

БУДЕМ РАДЫ ВСТРЕЧЕ!