

# Android-разработчик. Продвинутый курс

Курс обеспечивает глубокое погружение в промышленную Android-разработку на языке Kotlin, и за 5 месяцев позволяет прокачать свои навыки с уровня Junior до Middle или Senior Android-разработчика.

Длительность курса: 152 академических часа

## 1 Kotlin

- |   |   |   |
|---|---|---|
| 1 | <b>Kotlin стандартная библиотека. Часть 1</b> | После этого занятия вы сможете оперировать основными конструкциями языка Kotlin JVM, сможете писать код в kotlin стиле, использовать библиотеку anko. |
| 2 | <b>Kotlin стандартная библиотека. Часть 2</b> | После этого занятия вы сможете оперировать основными конструкциями языка Kotlin JVM, сможете писать код в kotlin стиле, использовать библиотеку anko. |
| 3 | <b>Kotlin coroutines</b>                      | После занятия вы сможете использовать корутины и каналы для написания асинхронного кода.  |

## 2 Android Framework

- 1 Advanced Fragment**

Child Fragments, Headless Fragments, Fragments communications, Single Activity Application.  
После занятия вы сможете применить полученные знания при проектировании приложения, полностью построенного на Фрагментах.

---
- 2 Advanced Service**

JobService, BoundService, IPC, AIDL, Handler/Messenger  
После этого занятия сможете грамотно выбирать необходимую реализацию Android Service исходя из задачи и версии ОС, научитесь выстраивать коммуникации между сервисами, которые работают в разных процессах.

---
- 3 Advanced Views**

View Lifecycle, measurement, drawing, SurfaceView, Constraint Layout.  
После занятия вы сможете применять знания о процессах отрисовки view при создании собственных имплементаций Android View.

---
- 4 Advanced Canvas**

Canvas, Bitmap, PorterDuff, Matrix, Path.  
После занятия вы сможете использовать возможности Canvas API из Android SDK для рисования сложных оптимизированных view.

### Домашние задания

- 1** Нарисовать кастомную интерактивную View, используя Canvas, подходящую для вашего приложения.
-

## 5 UI Profiling

GPU profiling, dumpsys, GPS measurement, слои наложения.

После занятия вы сможете использовать различные инструменты для поиска потенциальных узких мест в отрисовке UI и оптимизировать низкопроизводительные решения для быстрой отрисовки ваших приложений.

---

## 6 Network

Interceptors, CallAdapter.Factory, SSL pinning, WebSocket connection, GSON.

После этого занятия вы сможете применять широкие возможности http клиента Retrofit и десериализатора Gson для построения сетевого уровня в своем приложении.

Домашние задания

- 1 Добавить сетевой слой в приложение, используя изученные фреймворки
- 

## 7 Security in Android

Android Keystore, Google tink, алгоритмы шифрования в java cipher/android SDK.

После этого занятия вы сможете выбрать оптимальный алгоритм шифрования для вашего случая, сможете организовать безопасное хранилище секретных данных

Домашние задания

- 1 Подключить Google tink/ любую другую библиотеку для шифрования и хранения секретных данных. Организовать шифрование и безопасное хранение секретных данных(токенов/секретов).
-

## 8 Proguard

После занятия вы сможете встроить Proguard обфускацию и оптимизацию в свое приложение.

Домашние задания

- 1 Подключить Proguard для минификации и обфускации. Написать proguard-rules.pro для вашего приложения.
- 

## 9 Reverse Engineering

APKtool, jadx.

После этого занятия вы сможете разобрать .apk файл и проверить код на наличие вредоносных действий.

---

## 10 Android studio tips and tricks

После этого занятия вы сможете эффективно использовать возможности Android Studio для профилирования и отладки.

# 3 Architecture

1 **Design patterns** GoF паттерны, publisher/ subscriber, LRU etc. После этого занятия вы научитесь применять на практике часто используемые паттерны проектирования из GoF, паттерны многопоточного взаимодействия и синхронизации, кеширования и т.п.

---

2 **MV\* patterns in Android** Архитектурные паттерны: MVP/ MVVM/MVI. После занятия вы сможете подбирать нужный архитектурный проект и собирать проект на основе данной архитектуры.

Домашние задания

- 1 Выбрать архитектурный паттерн для презентейшен слоя и написать фичу, следуя принципам этого подхода.
- 

3 **Clean Architecture** После занятия вы сможете выстраивать архитектуру в соответствии с принципами Clean Architecture сформулированными Бобом Мартином.

Домашние задания

- 1 Добавить домен слой, содержащий бизнес логику.
- 

4 **Android Architecture Components** LifecycleOwner, LiveData, Room, Paging, ViewModel. После занятия вы сможете применять AAC в своих приложениях, адаптировать их под конкретно выбранную архитектуру.

---

## 5 Dagger 2

Dagger2, Dagger2 Android Injector.

Вы сможете правильно организовывать DI в приложениях, используя Dagger2.

Домашние задания

- 1 Организовать dependency injection в приложении, используя dagger2
- 

## 6 RxJava 2

После занятия вы научитесь применять “недооцененные” операторы, писать собственные операторы, узнаете как работают шедулеры.

---

## 7 Multimodule project

Dagger2, Gradle modules.

После этого занятия вы научитесь проектировать мультимодульную архитектуру приложения.

Домашние задания

- 1 Разбить приложение на gradle модули по фичам
- 

## 8 Code smells + refactoring

После этого занятия вы научитесь находить плохие архитектурные решения, строить roadmap и проводить рефакторинг.

## 1 Юнит тесты

JUnit, Hamcrest matchers.

После этого занятия вы сможете покрывать unit тестами ваши классы.

Домашние задания

- 1 Покрыть бизнес логику вашего приложения Unit тестами, используя фейковые имплементации и стабы.
- 

## 2 Интеграционное тестирование на Android

Mockito, Robolectric.

После этого занятия вы сможете использовать фреймворки для интеграционного тестирования на Android, реализовывать различные виды тестовых дублеров, создавать shadowклассы.

Домашние задания

- 1 Покрыть тестами классы с Android Framework, используя моки в качестве тестовых дублеров. Использовать Robolectric для интеграционных тестов с Android SDK. Необходимо написать свою имплементацию Shadow класса.
-

3 **Automated UI Testing** Espresso framework, Spoon, MockWebserver .  
После этого занятия вы сможете писать UI тесты на Espresso в изолированной среде и запускать их на нескольких эмуляторах параллельно.

Домашние задания

- 1 Покрыть UI тестами один сценарий.  
Использовать mockwebserver для моков http респонсов.

---

4 **Бонусное занятие №1** тема занятия будет выбрана преподавателем курса на основе пожеланий учеников.

---

5 **Бонусное занятие №2** тема занятия будет выбрана преподавателем курса на основе пожеланий учеников.



# 5 Infrastructure and build processes

## 1 Инструменты для проверки качества кода

Android Lint, findbugs, checkstyle, find-sec-bugs, ktlint, detekt.

После этого занятия вы сможете внедрять и настраивать линтеры/чекстайлы для проверки качества кода и поиска потенциальных ошибок.

Домашние задания

- 1 Подключить к своему проекту checkstyle/findbugs/detekt/ktlint любой другой линтер/статический анализатор кода. Сконфигурировать его под свои нужды и свой code style.

## 2 Системы контроля версий, различные виды ветвлений, код ревью

gitflow, trunk-based development, code review.

По окончании этого занятия вы сможете сравнить различные типы ветвления в git, научитесь эффективно проводить code review своим коллегам.

## 3 Gradle. Часть 1

Gradle Task, Gradle Plugin, Gradle DSL syntax.

После занятия вы сможете писать свои Gradle Tasks и Gradle Plugin, подключать их в проект и встраивать в пайплайн.

Домашние задания

- 1 Написать Gradle task/pipeline из тасок, которая будет выполнять build/checkQuality/test цикл.

4 **Gradle. Часть 2** Gradle Task, Gradle Plugin, Gradle DSL syntax.  
После занятия вы сможете писать свои Gradle Tasks и Gradle Plugin, подключать их в проект и встраивать в пайплайн.

---

5 **Docker** Docker, DockerRegistry, Docker Compose, Docker CLI.  
После занятия вы научитесь собирать докер образ из Dockerfile, поднимать контейнер из образа, взаимодействовать с контейнером.

Домашние задания

- 1 Собрать Docker образ и запустить из него docker-контейнер с JDK, Android SDK, Gradle.  
Попробовать собрать свой проект в этом контейнере.
- 

6 **Jenkins** Jenkins, Jenkins pipeline.  
После занятия вы научитесь поднимать Jenkins в Docker контейнере и делегировать ему сборку Android проекта.

Домашние задания

- 1 Поднять docker-контейнер с Jenkins-ом.  
Настроить job/pipeline, которая будет выполнять gradle tasks build/checkQuality/test в этом контейнере.
-

## 7 **Distribution / Crash overview**

Fabric, Firebase Crashlytics, AppCentre, Google play.  
По окончании этого занятия вы сможете деплоить  
ваш .apk файл в системы для дистрибуции  
приложений.

Домашние задания

- 1 Добавить Gradle task/pipeline шаг на deploy приложения в Fabric/AppCentre. + deploy apk-файла в artifactory

# 6 Проектная работа

1 **Консультация  
по ДЗ и  
проектам**

---

2 **Консультация  
по ДЗ и  
проектам**

---

3 **Защита  
проектов**

Домашние задания

- 1 Сдать ссылку на репозиторий курсового проекта. В репозитории обязательно должен быть заполнен файл Readme.md с описанием проекта.