

План прокачки до Junior Android Developer

Срок: 20.01.2026 — 24 недели (до середины июля) + выход на офферы к сентябрю

Формат: 2 часа в день минимум, ежедневно.

Автор плана: Сломов Д.С.

Как пользоваться этим планом

- 1) Каждый день выполняю блок “Теория” (20-40 мин) и “Практика” (80-100 мин).
- 2) В конце дня: 5 минут — запись в Learning Log (что понял, что не понял, что улучшить).
- 3) В конце недели: “Итог недели” — мини-ревью, оформление репозитория и список вопросов.
- 4) Минимальный стандарт: всё коммичу. Даже учебные вещи. Но “на витрину” (Pinned) ставим только сильное.

GitHub: что выкладывать

Выкладывать стоит почти всё (чтобы видеть прогресс и привычку к Git), но показывать работодателю — только лучшее.

Публично (обычно): 2-3 сильных проекта + 1-2 аккуратных демо (network/db/state).

Учебные маленькие задачи можно держать публично или в одном “kotlin-playground” репо, но не закреплять.

Ежедневный шаблон (2 часа)

- 0:00–0:10 — разогрев: повтор конспекта + постановка цели дня.
- 0:10–0:40 — теория: прочитать/посмотреть материал по теме дня.
- 0:40–1:50 — практика: код + мини-задачи/фича.
- 1:50–2:00 — Learning Log + коммит.

План по неделям

Месяц 1 — База Kotlin + база Android + дисциплина кода

Неделя 1 (20.01.2026 (Вт) — 26.01.2026 (Пн))

Неделя 1 — Kotlin фундамент I: типы, null-safety, data/sealed, extensions

Фокус:

Закрепить базовые конструкции Kotlin и начать писать аккуратный код маленькими блоками.

Что должно получиться к концу недели:

- Уверенно используешь val/var, типы, when/if, циклы, функции.
- Понимаешь null-safety и умеешь проектировать модели через data/sealed.
- Есть первый тренировочный репозиторий с задачами + заметки (Learning Log).

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
20.01.2026 (Вт) День 1	Kotlin: типы, val/var, функции. Чтение 20-30 мин.	10 упражнений: ввод/вывод, условия, циклы. 2 задачи: fizzbuzz, подсчет статистики.	Папка week01/day01 + коммит.
21.01.2026 (Ср) День 2	Null-safety: nullable типы, ?. ?: !, let.	5 задач на null: безопасный парсинг, обработка опциональных полей, цепочки вызовов.	Мини-консольная модель User? -> UI-модель.
22.01.2026 (Чт) День 3	Data class: equals/hashCode, copy, destructuring.	Смоделировать 5 сущностей (Car, Engine, Owner...). Сделать преобразования сору + валидации.	Тестовые данные + вывод результата.
23.01.2026 (Пт) День 4	Sealed class + when (exhaustive), enum vs sealed.	Сделать state-машину: Loading/Content/Error/Empty. Написать функцию reduce(event).	State machine в отдельном файле.
24.01.2026 (Сб) День 5	Extensions + scope functions: apply/also/run/with/let.	Рефакторинг вчерашнего кода: убрать дублирование, добавить 2-3 extension функции.	Коммит с рефакторингом + краткий вывод в Learning Log.
25.01.2026 (Вс) День 6	Коллекции: List/Set/Map, базовые операции.	10 задач на коллекции: фильтры, группировки, подсчеты (frequency map).	Сборник задач day06.
26.01.2026 (Пн) День 7	Повторение: перечитать Learning Log, закрыть пробелы.	Недельный челлендж: «мини-каталог» (модели + поиск/сортировка) в консоли.	Релиз-тэг v0.1 + README (как запускать).

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Неделя 2 (27.01.2026 (Вт) — 02.02.2026 (Пн))

Неделя 2 — Kotlin фундамент II: коллекции глубже, ООП-дизайн, generics, immutability

Фокус:

Научиться мыслить объектами и данными: чистые модели, неизменяемость, обобщения.

Что должно получиться к концу недели:

- Уверенно используешь map/filter/fold/groupBy/associate.
- Понимаешь generics и variance на базовом уровне (out/in).
- Мини-библиотека моделей + преобразования + 5-10 unit-тестов (первые).

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
27.01.2026 (Вт) День 1	FP-операции коллекций: map/filter/flatMap.	Решить 6 задач (трансформации, фильтры, flatMap).	Коммит day01.
28.01.2026 (Ср) День 2	fold/reduce, groupBy, counting.	Сделать аналитический модуль: статистика по списку объектов (min/max/avg/topN).	Файл analytics.kt + примеры.
29.01.2026 (Чт) День 3	Generics: T, ограничения, базовая variance.	Написать generic контейнер + 3 функции расширения (mapLike, filterLike).	Коммит day03.
30.01.2026 (Пт) День 4	ООП-дизайн: SRP, инкапсуляция, интерфейсы.	Спроектировать «мини-библиотеку» домена (например, трекер привычек в консоли).	UML-набросок в README (простая схема).
31.01.2026 (Сб) День 5	Immutability: val, копирование, избегаем shared mutable state.	Рефакторинг: заменить mutable структуры на immutable + copy.	Коммит с пометкой refactor.
01.02.2026 (Вс) День 6	Тестирование на Kotlin (JUnit): что тестировать.	Написать 5-10 тестов на analytics и reduce(state).	Запускаемые тесты + зелёный прогон.
02.02.2026 (Пн) День 7	Итог недели: обзор ошибок, стиль кода.	Сделать короткий «код-ревью самому себе»: 10 улучшений (format, naming).	Checklist в README + v0.2.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Неделя 3 (03.02.2026 (Вт) — 09.02.2026 (Пн))

Неделя 3 — Android основа + Jetpack Compose I

Фокус:

Понять каркас Android и собрать первые Compose-экраны.

Что должно получиться к концу недели:

- Создаёшь проект, понимаешь Activity/Intent/Context и жизненный цикл (база).
- Пишешь Compose UI: Column/Row, Text/Button, Modifiers, Preview.
- Мини-приложение: 1-2 экрана, состояние (remember) работает.

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
03.02.2026 (Вт) День 1	Android: структура проекта, Gradle, Manifest, Activity lifecycle (обзор).	Создать проект Compose. Запустить на эмуляторе/телефоне. Поиграться с Logcat.	Проект week03-app + первый коммит.
04.02.2026 (Ср) День 2	Compose: Composable, Preview, базовые layout.	Собрать экран «Profile»: аватар-заглушка, имя, кнопка.	Коммит day02.
05.02.2026 (Чт) День 3	State: remember/mutableStateOf, recomposition (интуитивно).	Сделать «Counter» + 2 состояния (enabled/disabled).	Коммит day03.
06.02.2026 (Пт) День 4	Списки: LazyColumn, items, ключи.	Сделать экран списка (20 элементов) + клик по элементу (пока log).	Коммит day04.
07.02.2026 (Сб) День 5	Input: TextField, валидации (простые).	Добавить «поиск» по списку (фильтрация по введенному тексту).	Поиск работает без лагов.
08.02.2026 (Вс) День 6	Навигация (через состояние или Navigation Compose обзор).	Сделать экран «Details» и переход из списка (можно временно через Navigation Compose).	2 экрана + back.
09.02.2026 (Пн) День 7	Итог: стиль UI, читаемость.	Добавить Material3 компоненты, отступы, пустое состояние. README со скринами.	v0.1 UI mini app.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Неделя 4 (10.02.2026 (Вт) — 16.02.2026 (Пн))

Неделя 4 — Мини-проект UI: 3-4 экрана (список -> детали -> фильтр -> настройки)

Фокус:

Собрать аккуратный UI-проект и научиться доводить до завершённого вида.

Что должно получиться к концу недели:

- Приложение с 3-4 экранами, навигацией и аккуратным состоянием.
- Есть базовая структура пакетов: ui / model / util.
- GitHub: README, скриншоты, понятные коммиты.

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
10.02.2026 (Вт) День 1	Планирование: экраны, навигационный граф, модели данных.	Создать репозиторий ui-mini. Набросать макеты Compose + навигацию.	Skeleton проекта.
11.02.2026 (Ср) День 2	Compose: top app bar, scaffold, spacing.	Реализовать экран списка (данные заглушки) + поиск.	Рабочий список.
12.02.2026 (Чт) День 3	Экран деталей: аргументы навигации.	Реализовать Details + отображение полей + back.	Переходы готовы.
13.02.2026 (Пт) День 4	Фильтр/сортировка: состояние и события.	Сделать экран фильтра (чекбоксы/слайдеры) и применить к списку.	Фильтры работают.
14.02.2026 (Сб) День 5	Настройки: DataStore (пока можно mock/в памяти) или простой state.	Экран настроек (тема, сортировка по умолчанию).	Настройки применяются.
15.02.2026 (Вс) День 6	Полировка: пустые состояния, ошибки, loading-заглушка.	Добавить состояния UI и единый UiState sealed class.	UiState внедрен.
16.02.2026 (Пн) День 7	Финал: оформление репо.	README: стек, скрины, как запустить. Пины не ставим, но репо публичное.	Release v1.0.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Месяц 2 — Сеть, хранение, состояния, архитектура

Неделя 5 (17.02.2026 (Вт) — 23.02.2026 (Пн))

Неделя 5 — Coroutines: suspend, dispatchers, structured concurrency, cancellation

Фокус:

Научиться писать асинхронный код без хаоса.

Что должно получиться к концу недели:

- Понимаешь suspend и scopes (viewModelScope/CoroutineScope).
- Умеешь отменять корутины и обрабатывать исключения.
- Есть набор упражнений + экран с имитацией загрузки/ошибок.

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
17.02.2026 (Вт) День 1	Что такое корутины, suspend, launch/async (обзор).	Консольные упражнения: delay, параллельные запросы (async + await).	Коммит упражнений day01.
18.02.2026 (Ср) День 2	Dispatchers: Main/IO/Default, withContext.	Сделать «fake repository» с задержкой и переключением контекста.	Файл fake_repo.kt.
19.02.2026 (Чт) День 3	Structured concurrency: scope, children, supervisor.	Написать пример: несколько задач, одна падает, понять поведение.	Заметки в Learning Log.
20.02.2026 (Пт) День 4	Cancellation: cooperative cancellation, ensureActive.	Сделать пример отмены загрузки (кнопка Cancel) в Compose.	Экран cancel-demo.
21.02.2026 (Сб) День 5	Exceptions: try/catch, CoroutineExceptionHandler.	Симуляция ошибок: timeout, network error -> map to UiError.	Error mapping.
22.02.2026 (Вс) День 6	Таймауты: withTimeout/withTimeoutOrNull.	Сделать 3 сценария: успех, ошибка, таймаут. Отобразить в UI.	3 сценария работают.
23.02.2026 (Пн) День 7	Итог недели: мини-рефакторинг.	Собрать маленький модуль “async-	v0.1 async demo.

		demo-app" в отдельном репо.	
--	--	--------------------------------	--

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Неделя 6 (24.02.2026 (Вт) — 02.03.2026 (Пн))

Неделя 6 — Flow: cold/hot, StateFlow/SharedFlow, операторы, combine

Фокус:

Научиться думать потоками данных и собирать стейт экрана.

Что должно получиться к концу недели:

- Понимаешь разницу cold/hot и где использовать StateFlow.
- Собираешь UiState через combine и операторные цепочки.
- Есть экран со state machine: Loading/Content/Error/Empty.

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
24.02.2026 (Вт) День 1	Flow basics: builder, collect, map/filter.	Упражнения: flowOf/flow { emit } + 5 операторов.	day01 exercises.
25.02.2026 (Ср) День 2	StateFlow: state in ViewModel, collectAsState.	Сделать экран поиска: запрос -> debounced flow -> результаты (fake).	Search flow demo.
26.02.2026 (Чт) День 3	SharedFlow: events (snackbar, navigation events).	Добавить одноразовое событие (Toast/Snackbar) через SharedFlow.	Event flow done.
27.02.2026 (Пт) День 4	combine: объединяем query + filter + source.	Собрать итоговый список через combine и сортировку.	Combine pipeline.
28.02.2026 (Сб) День 5	Backpressure и отмена: flatMapLatest, debounce.	Добавить debounce 300ms + flatMapLatest для запроса.	Без лишних запросов.
01.03.2026 (Вс) День 6	State machine: sealed UiState + reducer.	Сделать универсальный UiState и покрыть 4 состояния.	UiState ready.
02.03.2026 (Пн) День 7	Итог: мини-проект.	Оформить “state-machine-screen” как модуль/репо + README.	v0.1 flow demo.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Неделя 7 (03.03.2026 (Вт) — 09.03.2026 (Пн))

Неделя 7 — Networking: Retrofit + Serialization, ошибки, DTO -> Domain

Фокус:

Научиться получать данные из API и превращать их в доменные модели.

Что должно получиться к концу недели:

- Настроен Retrofit + OkHttp + Kotlin serialization (или Moshi).
- Есть обработка ошибок (HTTP, сеть, парсинг) и маппинг DTO.
- Экран: загрузка списка + retry.

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
03.03.2026 (Вт) День 1	HTTP основы: GET/POST, коды ответов, JSON.	Выбрать публичный API (например, список объектов). Запрос через Retrofit.	Первый успешный запрос.
04.03.2026 (Ср) День 2	Serialization: DTO data classes, @Serializable.	Описать DTO и распарсить ответ. Логи результата.	DTO работает.
05.03.2026 (Чт) День 3	OkHttp: interceptors (логирование), timeouts.	Добавить логирующий interceptor и таймауты.	Сетевые логи.
06.03.2026 (Пт) День 4	Ошибки: HTTP error body, IOException.	Сделать mapper ошибок в UiError и показать на экране.	Error UI.
07.03.2026 (Сб) День 5	DTO -> Domain mapping.	Создать доменные модели и маппинг, скрыть DTO от UI.	Domain layer ready.
08.03.2026 (Вс) День 6	Retry: кнопка и повторный запрос.	Экран списка: loading + content + error + empty + retry.	Состояния экрана.
09.03.2026 (Пн) День 7	Итог недели: выделить сетевой модуль.	Небольшой рефакторинг: network package + README.	v0.1 network demo.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Неделя 8 (10.03.2026 (Вт) — 16.03.2026 (Пн))

Неделя 8 — Persistence: Room + DataStore, кеширование списка, offline

Фокус:

Научиться хранить данные локально и делать offline-first (упрощенно).

Что должно получиться к концу недели:

- Room: Entity/Dao/Database, простые запросы.
- DataStore: 1-2 настройки (тема/сортировка).
- Кеш списка + избранное, работает без сети.

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
10.03.2026 (Вт) День 1	Room setup: Entity/Dao/Database.	Подключить Room в demo проект. Создать 1 таблицу.	Вставка/чтение работает.
11.03.2026 (Ср) День 2	Flow из Room: наблюдение таблицы.	Dao возвращает Flow<List<...>> и UI обновляется.	Reactive DB.
12.03.2026 (Чт) День 3	Миграции базово: что это и зачем (прочитать).	Сделать простую миграцию (добавить колонку) в учебном проекте.	Migration ok.
13.03.2026 (Пт) День 4	DataStore preferences: ключи, чтение/запись.	Настройка: сортировка/тема, сохраняется между запусками.	Preference persisted.
14.03.2026 (Сб) День 5	Offline-first: источник истины - БД.	Репозиторий: UI подписан на БД; сеть только обновляет БД.	Offline-first flow.
15.03.2026 (Вс) День 6	Избранное: toggle favorite.	Добавить поле isFavorite и кнопку в UI + сохранение в БД.	Favorites ready.
16.03.2026 (Пн) День 7	Итог: собрать «network+db» в один мини-проект.	README + скрины + список известных ограничений.	Release v1.0 мини- проекта.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Месяц 3 — Продвинутая разработка: фон, навигация, качество

Неделя 9 (17.03.2026 (Вт) — 23.03.2026 (Пн))

Неделя 9 — Navigation: multi-screen, аргументы, back stack, state restoration

Фокус:

Навигация как в реальном приложении + сохранение состояния.

Что должно получиться к концу недели:

- Навигация между 3-5 экранами через Navigation Compose.
- Аргументы, deep-link (по желанию), back stack сценарии.
- Состояние списка/поиска сохраняется при возврате.

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
17.03.2026 (Вт) День 1	Navigation Compose: NavHost, routes.	Вынести навигацию в отдельный файл, настроить граф.	NavGraph ready.
18.03.2026 (Ср) День 2	Аргументы: pass id, type-safe подход (минимально).	Экран Details получает id и загружает данные.	Args working.
19.03.2026 (Чт) День 3	Back stack: popUpTo, singleTop.	Сценарии: логин -> main, чистим back stack.	Back stack scenario.
20.03.2026 (Пт) День 4	State restoration: rememberSaveable, сохранение scroll.	Сохранить query/filters и позицию списка.	Restore ok.
21.03.2026 (Сб) День 5	Навигационные события из ViewModel (SharedFlow).	Сделать event NavigationEvent и обработать в UI.	Nav events.
22.03.2026 (Вс) День 6	Тестируем сценарии вручную.	Пройти 10 сценариев переходов и зафиксировать баги в Issues.	10 issues.
23.03.2026 (Пн) День 7	Итог: закрыть баги.	Исправить минимум 5 issues, обновить README.	v1.1.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Неделя 10 (24.03.2026 (Вт) — 30.03.2026 (Пн))

Неделя 10 — DI: Hilt (модули, зависимости, scopes)

Фокус:

Внедрение зависимостей, чтобы код масштабировался и тестировался.

Что должно получиться к концу недели:

- Hilt настроен: Application, @AndroidEntryPoint, @HiltViewModel.
- Network/DB/Repository предоставляются через модули.
- Минимум 1 unit-test с fake зависимостью.

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
24.03.2026 (Вт) День 1	DI концепции: зачем, что решает (прочитать).	Подключить Hilt в проект (плагин, зависимости).	Проект собирается.
25.03.2026 (Ср) День 2	Application + entry points.	Добавить @HiltAndroidApp и @AndroidEntryPoint.	Инжект работает.
26.03.2026 (Чт) День 3	Модули: provide Retrofit/OkHttp.	Сделать NetworkModule и проверить создание ApiService.	Network DI done.
27.03.2026 (Пт) День 4	Модули: provide Room/Dao.	Сделать DatabaseModule и проверить Dao injection.	DB DI done.
28.03.2026 (Сб) День 5	Repository DI + интерфейсы.	Repository интерфейс + реализация, биндинг в module.	Repo DI done.
29.03.2026 (Вс) День 6	Scopes: Singleton, ViewModelScoped (обзор).	Проверить жизненный цикл зависимостей (простые логи).	Scopes understood.
30.03.2026 (Пн) День 7	Итог: тестируемость.	Написать 1 unit-test на ViewModel с fake repository.	Тест зеленый.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Неделя 11 (31.03.2026 (Вт) — 06.04.2026 (Пн))

Неделя 11 — WorkManager: фоновые задачи, constraints, уведомления

Фокус:

Делать фоновые обновления и понимать ограничения Android.

Что должно получиться к концу недели:

- WorkManager запускает периодическую задачу с constraints.
- Обновление кеша в фоне (упрощенно) + уведомление/лог.
- Понимаешь разницу: WorkManager vs foreground service (обзор).

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
31.03.2026 (Вт) День 1	WorkManager basics: OneTime vs Periodic, constraints.	Добавить зависимость, создать простой Worker.	Worker runs.
01.04.2026 (Ср) День 2	Periodic work + constraints (Wi-Fi/charging).	Настроить periodic обновление кеша 1 раз в день (для теста чаще).	Periodic configured.
02.04.2026 (Чт) День 3	Передача параметров и результат.	Worker: загрузка данных (fake/real) и сохранение в БД.	Cache updated.
03.04.2026 (Пт) День 4	Уведомления: базово (Notification).	Показать уведомление о завершении обновления.	Notification shown.
04.04.2026 (Сб) День 5	Устойчивость: retry/backoff.	Добавить backoff criteria и обработку ошибок.	Retry works.
05.04.2026 (Вс) День 6	Отладка и логирование WorkManager.	Проверить выполнение в разных условиях, записать заметки.	Learning Log update.
06.04.2026 (Пн) День 7	Итог: оформить в проекте.	Документировать: зачем, как включить, ограничения.	v1.2.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Неделя 12 (07.04.2026 (Вт) — 13.04.2026 (Пн))

Неделя 12 — Полировка качества: edge cases, дебаг, логирование, трейсинг

Фокус:

Научиться доводить продукт до нормального состояния.

Что должно получиться к концу недели:

- Список багов/улучшений через Issues.
- Применены принципы обработки ошибок и пустых состояний.
- Код чище: именование, разделение ответственности.

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
07.04.2026 (Вт) День 1	Собрать список проблем: UX, ошибки, производительность.	Создать 15 issues (баги/улучшения) по текущему проекту.	Issues ready.
08.04.2026 (Ср) День 2	Логирование: где логировать, уровень логов.	Добавить логирование ключевых сценариев (без шума).	Logs added.
09.04.2026 (Чт) День 3	Edge cases: нет сети, пустой ответ, медленный ответ.	Доработать UI: пустые/ошибочные состояния + retry.	Edge cases handled.
10.04.2026 (Пт) День 4	Дебаг: breakpoints, inspect variables.	Поймать 2-3 бага и исправить с дебаггером.	3 issues closed.
11.04.2026 (Сб) День 5	Трейсинг простыми метриками (время загрузки).	Замерить время загрузки списка и оптимизировать (где возможно).	Perf note.
12.04.2026 (Вс) День 6	Рефакторинг: убрать длинные функции, вынести маппинг.	Закрыть ещё 5 issues.	10 issues closed.
13.04.2026 (Пн) День 7	Итог: релизный проход.	Сделать release сборку, обновить README, changelog.	Release v1.3.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Месяц 4 — Архитектура уровня компании + тесты

Неделя 13 (14.04.2026 (Вт) — 20.04.2026 (Пн))

Неделя 13 — Clean-ish архитектура: data/domain/ui, repository, use cases

Фокус:

Сделать архитектуру, которую можно объяснить на собеседовании.

Что должно получиться к концу недели:

- Проект разбит на слои (data/domain/ui) без фанатизма.
- DTO и Entity не протекают в UI.
- 1-3 use case класса, понятные границы.

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
14.04.2026 (Вт) День 1	Архитектура: слои и зависимости (прочитать/конспект).	Набросать текущие зависимости пакетов, найти нарушения.	Схема в README.
15.04.2026 (Ср) День 2	Data layer: network + db + mappers.	Вынести мапперы, спрятать DTO/Entity.	Data cleaned.
16.04.2026 (Чт) День 3	Domain layer: модели и use cases.	Сделать 1-2 use case (LoadItems, ToggleFavorite).	Use cases done.
17.04.2026 (Пт) День 4	UI layer: ViewModel только вызывает use cases.	Рефакторинг ViewModel под use cases.	VM simplified.
18.04.2026 (Сб) День 5	Ошибки: доменные ошибки vs UI ошибки.	Создать DomainError и маппинг в UiError.	Errors structured.
19.04.2026 (Вс) День 6	Код-ревью: читаемость и контракты.	Убрать дубли, добавить документацию к публичным классам.	Docs improved.
20.04.2026 (Пн) День 7	Итог недели.	Короткий текст в README: почему такая архитектура и trade-offs.	v1.4.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Неделя 14 (21.04.2026 (Вт) — 27.04.2026 (Пн))

Неделя 14 — Unit-тесты: use case/repo, fakes, моки

Фокус:

Научиться писать тестируемый код и проверять бизнес-логику.

Что должно получиться к концу недели:

- Покрыты use cases и маппинг ошибок.
- Используются fakes (и/или моки) для зависимостей.
- Тесты быстрые, читаемые, повторяемые.

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
21.04.2026 (Вт) День 1	Что тестировать: границы домена.	Написать тесты на 1 use case (успех/ошибка).	2-3 теста.
22.04.2026 (Ср) День 2	Fake repository: in-memory реализация.	Сделать fake repo и использовать в тестах.	Fake repo ready.
23.04.2026 (Чт) День 3	Coroutines test: runTest, dispatcher.	Перевести тесты на runTest, убрать flakiness.	Stable tests.
24.04.2026 (Пт) День 4	Тесты на маппинг DTO->Domain и Domain->UI.	Добавить 3-5 тестов на мапперы.	Mapper tests.
25.04.2026 (Сб) День 5	Тест ViewModel (минимально): UiState transitions.	Проверить переходы Loading->Content/Error.	VM test.
26.04.2026 (Вс) День 6	Покрытие: выбрать 10 критичных сценариев.	Добавить тесты до 10-15 штук суммарно.	Coverage improved.
27.04.2026 (Пн) День 7	Итог: документация.	README секция "Testing": как запускать, что покрыто.	v1.5.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Неделя 15 (28.04.2026 (Вт) — 04.05.2026 (Пн))

Неделя 15 — UI-тестирование минимум + инструменты качества

Фокус:

Сделать 2-3 smoke UI теста и знать, как стабилизировать UI.

Что должно получиться к концу недели:

- Есть 2-3 UI теста на критичные флоу.
- Понимаешь принцип test tags и idling.
- Проект собирается в CI локально (хотя бы руками).

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
28.04.2026 (Вт) День 1	Compose UI testing: rule, setContent.	Написать тест на отображение пустого состояния.	UI test 1.
29.04.2026 (Ср) День 2	Test tags и поиск элементов.	Добавить testTag к кнопкам/полям и тест на добавление элемента.	UI test 2.
30.04.2026 (Чт) День 3	Навигация в тестах.	Тест: список -> детали -> назад.	UI test 3.
01.05.2026 (Пт) День 4	Стабилизация: deterministic state.	В тестах использовать fake repo, без реальной сети.	Stable UI tests.
02.05.2026 (Сб) День 5	Static analysis: ktlint/detekt (по желанию).	Подключить один инструмент и исправить ключевые предупреждения.	Lint improved.
03.05.2026 (Вс) День 6	CI базово (опционально): GitHub Actions.	Настроить workflow: assembleDebug + unit tests.	CI running.
04.05.2026 (Пн) День 7	Итог недели.	Обновить README: quality checks, badges (если есть).	v1.6.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Неделя 16 (05.05.2026 (Вт) — 11.05.2026 (Пн))

Неделя 16 — Gradle + модульность (базово): core/network/db

Фокус:

Понять сборку и научиться выделять модули без боли.

Что должно получиться к концу недели:

- Понимаешь buildTypes/flavors на базовом уровне.
- Проект (по желанию) разбит на 2-4 модуля.
- Зависимости между модулями корректные.

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
05.05.2026 (Вт) День 1	Gradle: modules, dependencies, version catalogs (обзор).	Почистить зависимости, убрать дубли, упорядочить версии.	Gradle cleanup.
06.05.2026 (Ср) День 2	buildTypes: debug/release, signing (обзор).	Собрать release локально, проверить, что всё ок.	Release builds.
07.05.2026 (Чт) День 3	Flavors (обзор) и конфиги (BuildConfig).	Добавить baseUrl через build config (debug vs release).	Config works.
08.05.2026 (Пт) День 4	Модульность: выделяем core.	Создать модуль :core и вынести утилиты/модели.	core module.
09.05.2026 (Сб) День 5	Выделить :data (network/db) или :network/:db.	Переместить код и починить DI.	Data module.
10.05.2026 (Вс) День 6	Проверка сборки и запуск тестов.	Починить все импорты, убедиться, что тесты проходят.	Green build.
11.05.2026 (Пн) День 7	Итог: решение и ограничения.	README: почему так, какие модули и зависимость. граф.	v1.7.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Месяц 5 — Капстоун-проект: портфолио уровня “берём на джуна”

Неделя 17 (12.05.2026 (Вт) — 18.05.2026 (Пн))

Неделя 17 — Капстоун: выбор идеи + ТЗ + дизайн скелета

Фокус:

Выбрать сильный проект для портфолио и запланировать работу спринтами.

Что должно получиться к концу недели:

- Есть идея, user stories, список экранов и навигация.
- Создан репозиторий capstone, настроены базовые модули.
- Собран UI-скелет 3-4 экранов без логики.

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
12.05.2026 (Вт) День 1	Выбор идеи: критерии (интересно, демонстрирует стек, реалистично).	Сформулировать 1 главную ценность + 5-7 фич (MVP).	MVP список.
13.05.2026 (Ср) День 2	ТЗ: user stories + acceptance criteria.	Описать 10-15 user stories и приоритизировать (MVP/Next).	Backlog готов.
14.05.2026 (Чт) День 3	Дизайн: вайрфреймы (простые).	Нарисовать 4-6 экранов (в FigJam/бумаге) и зафиксировать.	Wireframes.
15.05.2026 (Пт) День 4	Setup: проект, зависимости, CI (если было).	Создать репо, настроить Compose, навигацию, темы.	Project skeleton.
16.05.2026 (Сб) День 5	Архитектура: слои, модули (решение).	Создать пакеты/модули, заготовки repo/usecase/vm.	Arch skeleton.
17.05.2026 (Вс) День 6	UI скелет: список/детали/поиск/настройки.	Собрать экраны с заглушками и переходами.	UI navigates.
18.05.2026 (Пн) День 7	Итог: план спринта 18.	Создать issues на неделю 18 (10-15 задач), обновить README.	Sprint plan.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Неделя 18 (19.05.2026 (Вт) — 25.05.2026 (Пн))

Неделя 18 — Капстоун: сеть + домен + список (первый end-to-end)

Фокус:

Сделать рабочий вертикальный срез: API -> маппинг -> UI список.

Что должно получиться к концу недели:

- Есть реальный API вызов и отображение списка.
- UiState, ошибки, retry работают.
- Архитектурные границы соблюдены.

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
19.05.2026 (Вт) День 1	API контракт: выбрать endpoints, DTO.	Реализовать Retrofit сервис и один запрос.	API call ok.
20.05.2026 (Ср) День 2	Маппинг DTO->Domain.	Создать domain модели и mapper + тест на маппер.	Domain ready.
21.05.2026 (Чт) День 3	Repository: fetch + error mapping.	Repository возвращает Result/Outcome (успех/ошибка).	Repo ok.
22.05.2026 (Пт) День 4	Use case: LoadList.	Use case + unit тест (успех/ошибка).	UseCase tested.
23.05.2026 (Сб) День 5	ViewModel + UiState.	Экран списка подписан на state и показывает loading/error.	List screen works.
24.05.2026 (Вс) День 6	Retry + pull-to-refresh (по желанию).	Добавить кнопку retry и обновление.	Refresh ok.
25.05.2026 (Пн) День 7	Итог: polishing list.	Отступы, состояние пустого списка, скелетон.	Sprint review notes.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Неделя 19 (26.05.2026 (Вт) — 01.06.2026 (Пн))

Неделя 19 — Капстоун: локальное хранение + offline-first + избранное

Фокус:

Сделать кеш и устойчивость без интернета.

Что должно получиться к концу недели:

- Room кеширует список и является источником истины.
- Избранное сохраняется локально.
- Сеть обновляет БД; UI подписан на БД.

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
26.05.2026 (Вт) День 1	Room schema: Entity/Dao.	Создать таблицу, методы upsert, getAllFlow.	DB ready.
27.05.2026 (Ср) День 2	Repository: db as source of truth.	UI читает Flow из Dao, сеть обновляет db.	Offline-first.
28.05.2026 (Чт) День 3	Синхронизация: refresh strategy.	Решить: когда обновлять (pull-to-refresh, on start).	Strategy documented.
29.05.2026 (Пт) День 4	Избранное: toggle, хранение.	Добавить поле isFavorite + кнопку + сохранение.	Favorites persisted.
30.05.2026 (Сб) День 5	Обработка ошибок сети при refresh.	Если сеть упала, показываем данные из кеша + баннер.	Graceful degrade.
31.05.2026 (Вс) День 6	DataStore: настройки (сортировка/тема).	Сохранить настройку и применить в списке.	Settings work.
01.06.2026 (Пн) День 7	Итог недели: тесты.	2-3 unit теста repo/usecase на offline-first.	Tests added.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Неделя 20 (02.06.2026 (Вт) — 08.06.2026 (Пн))

Неделя 20 — Капстоун: Paging + фоновые обновления + финальная полировка

Фокус:

Добавить «продуктивность»: paging, фон, UX детали, README.

Что должно получиться к концу недели:

- Paging 3 (или упрощенная пагинация) работает.
- WorkManager обновляет кеш (по расписанию) или по условиям.
- README и скрины готовы, проект выглядит «как у разработчика».

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
02.06.2026 (Вт) День 1	Paging концепция: источник данных, load states.	Подключить Paging и вывести список с load states.	Paging works.
03.06.2026 (Ср) День 2	Paging + Room (RemoteMediator) (опционально) или упрощение.	Если сложно: оставить paging только по сети. Главное - показать skill.	Decision made.
04.06.2026 (Чт) День 3	Фон: WorkManager refresh.	Сделать periodic refresh и уведомление/лог.	Background refresh.
05.06.2026 (Пт) День 4	UX: ошибки, пустые состояния, доступность.	Пройти 15 сценариев, закрыть баги.	Issues closed.
06.06.2026 (Сб) День 5	Тесты: ключевая логика.	Добавить 5-7 unit тестов, 1-2 UI smoke теста.	Test suite improved.
07.06.2026 (Вс) День 6	README: архитектура, как запустить, стек.	Добавить схему слоев, скрины, gif (если хочешь).	Portfolio-ready README.
08.06.2026 (Пн) День 7	Релиз кандидата.	Собрать release, tag v1.0, написать "What I learned".	Capstone v1.0.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Месяц 6 — Собесы и трудоустройство

Неделя 21 (09.06.2026 (Вт) — 15.06.2026 (Пн))

Неделя 21 — Интервью-база Android: жизненный цикл, потоки, память, DI, Room/Retrofit, архитектура

Фокус:

Собрать ответы на частые вопросы и закрепить на примерах из твоего капстоуна.

Что должно получиться к концу недели:

- Конспект ответов на 40-60 вопросов (коротко).
- Умеешь объяснить архитектуру своего проекта за 3-5 минут.
- Знаешь типовые грабли и как их избегать.

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
09.06.2026 (Вт) День 1	Жизненный цикл Activity/Compose: что где хранить.	Сделать карточки Q/A (10-12 вопросов) + примеры из проекта.	Q/A day01.
10.06.2026 (Ср) День 2	Coroutines: structured concurrency, dispatchers, cancellation.	Карточки 10 вопросов + объяснение через свой код.	Q/A day02.
11.06.2026 (Чт) День 3	Flow/StateFlow/SharedFlow: когда что.	Карточки 8-10 вопросов + 1 мини-демо в проекте (если надо).	Q/A day03.
12.06.2026 (Пт) День 4	Retrofit/OkHttp: interceptors, ошибки, timeouts.	Сделать "рассказ" про сетевой слой 3-4 минуты.	Narrative ready.
13.06.2026 (Сб) День 5	Room/DataStore: транзакции, миграции (база).	Карточки 8-10 вопросов + проверка своего кода.	Q/A day05.
14.06.2026 (Вс) День 6	DI: Hilt, scopes, тестируемость.	Сделать 1 упражнение: заменить реализацию через bind.	DI exercise.
15.06.2026 (Пн) День 7	Итог недели: мок-собес 1.	Провести 1 час: отвечаешь на вопросы вслух + записываешь	Список пробелов.

		слабые места.	
--	--	---------------	--

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Неделя 22 (16.06.2026 (Вт) — 22.06.2026 (Пн))

Неделя 22 — Алгоритмы базово: массивы/строки, map/set, два указателя

Фокус:

Собес-минимум по алгоритмам: чистое решение и объяснение.

Что должно получиться к концу недели:

- Решено 10-14 задач уровня easy.
- Умеешь объяснять решение: вход/выход, шаги, сложность.
- Есть репо/папка algorithms с решениями и заметками.

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
16.06.2026 (Вт) День 1	Массивы: сумма, минимум/максимум, префиксные суммы.	Решить 2 задачи + написать объяснение.	2 solutions.
17.06.2026 (Ср) День 2	HashMap/HashSet: частоты, проверка уникальности.	Решить 2 задачи (two-sum, anagram).	2 solutions.
18.06.2026 (Чт) День 3	Строки: палиндром, нормализация.	2 задачи + фокус на edge cases.	2 solutions.
19.06.2026 (Пт) День 4	Два указателя: окно, сортировка.	2 задачи (remove duplicates, merge).	2 solutions.
20.06.2026 (Сб) День 5	Стек/очередь (минимум): валидные скобки.	1-2 задачи.	2 solutions.
21.06.2026 (Вс) День 6	Ревью: стиль решений, именование.	Отрефакторить 3 решения, добавить тесты где уместно.	Refactor done.
22.06.2026 (Пн) День 7	Мок-собес 2 (алго).	Решить 1 задачу вслух за 30-40 мин + разбор.	Notes.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Неделя 23 (23.06.2026 (Вт) — 29.06.2026 (Пн))

Неделя 23 — Системное мышление: проектирование экранов, кеширование, state, trade-offs

Фокус:

Научиться говорить как разработчик: аргументы и компромиссы.

Что должно получиться к концу недели:

- Умеешь проектировать фичу: данные, UI state, источники истины.
- Можешь объяснить offline-first, paging, фоновые задачи.
- Капстоун готов к обсуждению 30 минут.

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
23.06.2026 (Вт) День 1	Как спроектировать экран: события, стейт, эффекты.	Сделать шаблон: UiState + UiEvent + UiEffect для одного экрана.	Template doc.
24.06.2026 (Ср) День 2	Кеширование: стратегия, инвалидация, stale-данные.	Описать стратегию кеша в капстоуне (параграф в README).	Cache strategy.
25.06.2026 (Чт) День 3	Paging: где bottlenecks.	Разобрать свой paging: что загрузка, что хранение, как тестировать.	Notes.
26.06.2026 (Пт) День 4	Ошибки: user-friendly и recoverable.	Сделать таблицу ошибок (тип -> UI реакция).	Error table.
27.06.2026 (Сб) День 5	DI и тестируемость как аргумент.	Подготовить рассказ: как DI помог тестам/архитектуре.	Story.
28.06.2026 (Вс) День 6	Разбор капстоуна как на собесе.	Сделать презентацию в README: архитектура, модули, data flow.	Interview-ready doc.
29.06.2026 (Пн) День 7	Мок-собес 3 (system design light).	Интервью вслух: “как сделать offline-first список?”	Notes.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

Неделя 24 (30.06.2026 (Вт) — 06.07.2026 (Пн))

Неделя 24 — Резюме + отклики + мок-собесы + финальная упаковка GitHub

Фокус:

Перевести навыки в оффер: резюме, портфолио, отклики, интервью.

Что должно получиться к концу недели:

- Резюме 1 страница, GitHub профиль с 2-3 pinned проектами.
- 30-50 откликов (в реальности может растянуться) и трекинг.
- 3-5 мок-интервью + план доработок.

День	Теория (20-40 мин)	Практика (80-100 мин)	Выход / коммит
30.06.2026 (Вт) День 1	Резюме: структура и формулировки.	Собрать резюме на 1 страницу + список технологий + проекты.	Draft resume.
01.07.2026 (Ср) День 2	GitHub: pinned проекты, README, скрины.	Выбрать 2-3 лучших репо, обновить README и оформить профиль.	Portfolio ready.
02.07.2026 (Чт) День 3	Шаблон сопроводительного текста.	Написать 2 версии: короткая и развернутая.	Cover texts.
03.07.2026 (Пт) День 4	Отклики: составить список компаний/вакансий.	Сделать таблицу трекинга (company, link, status, notes).	Tracking table.
04.07.2026 (Сб) День 5	Мок-собес 4 (Android).	60 минут: вопросы по стеку + разбор.	Gaps list.
05.07.2026 (Вс) День 6	Мок-собес 5 (capstone deep dive).	30-40 мин рассказ о проекте + вопросы "почему так?".	Improvement plan.
06.07.2026 (Пн) День 7	Финал: план на следующие 4 недели (август).	Составить расписание: отклики, интервью, точечные подтяжки.	Next steps.

Definition of Done недели:

- Код собирается и запускается.
- Есть минимум 5-7 осмысленных коммитов за неделю.
- Learning Log заполнен каждый день.
- README обновлен (если это проектная неделя).

После 24 недель: как дожать до оффера (август–сентябрь)

- 1) Продолжаю отклики каждую неделю + трекинг (таблица).
- 2) Каждую неделю: 1 мок-собес + 3-4 задачи по алгоритмам.
- 3) Раз в неделю: улучшение капстоуна (по фидбеку) без ломания основных фич.
- 4) Точечные темы по вакансиям (например, Views/RecyclerView, если часто спрашивают).