

Android-разработчик. Базовый курс

Научитесь профессионально создавать приложения для самой популярной мобильной ОС. Подготовьте качественное портфолио и выложите его в Google Play. По итогам обучения - пройдёте собеседование в одну из компаний-партнеров.

Длительность курса: 132 академических часа

1 Подготовительный курс по Java

- | | | |
|---|--------------------------------------|---|
| 1 | Консультация по Java. Часть 1 | На этом занятии вы разберете все вопросы, которые возникли в процессе изучения подготовительного видео-курса по Java. |
| 2 | Консультация по Java. Часть 2 | На этом занятии вы разберете все вопросы, которые возникли в процессе изучения подготовительного видео-курса по Java. |

2 Основы Android-разработки

1 Знакомство с Android Studio

Новый проект, структура проекта, Gradle, отладка кода, Git.

После занятия вы сможете: создать и отладить новый проект в Android Studio, будете понимать структуру проекта, а также работать с системой контроля версий Git.

2 Главный компонент Android — Activity

Жизненный цикл, сохранение состояния, переходы между Activity, Intent, Context.

После занятия вы сможете: создавать новые экраны и передавать между ними данные, запускать другие приложения из своего, сохранять данные экрана при его пересоздании, а также делиться данными из приложения с друзьями.

Домашние задания

- 1 Начало работы над приложением по поиску фильмов. Создание проекта и страницы с фильмами.
 1. Создайте проект
 2. Залейте проект на GitHub
 3. Добавьте описание проекта в заголовке и Read.me
 4. На первом экране своего приложения создайте несколько картинок с фильмами, к каждой картинке подпишите название фильма и добавьте кнопку “Детали”
 5. По нажатию на Детали - выделяйте другим цветом название выбранного фильма, открывайте новое окно, где показывайте картинку и описание фильма
 6. Сохраняйте выделение фильма при

возвращении со второго экрана и при повороте экрана

7. Добавьте кнопку “Пригласить друга” и отправляйте приглашение по вашему выбору (почта, смс, социальные сети)

8. * Добавьте на втором экране checkbox “Нравится” и текстовое поле для комментария. Возвращайте значение чекбокса и текст комментария при переходе обратно на первый экран

P.S. Задание со звездочкой * - повышенной сложности. Если вы с ним не справитесь - ничего страшного, оно не является обязательным.

3 **Интерфейс экрана. Базовые элементы**

View, View group, элементы интерфейса, меню экрана, панель навигации.

После занятия вы сможете: заполнять экраны графическими элементами (кнопки, выкидное меню, поля ввода, изображения и т.д.), создавать меню экрана и панель навигации.

4 Интерфейс экрана. Material design

Material Design, элементы Material Design, анимация. После занятия вы сможете: создавать элементы в стиле Material Design, управлять ими, а также выполнять простые анимации (изменение положения, прозрачности и ориентации).

Домашние задания

- 1 Кнопка “пригласить друга”. Добавление анимации. Добавление новых фильмов в приложение.
 1. Нарисуйте интерфейс первого и второго экранов в соответствии с вашими навыками
 2. Добавьте кнопку меню и перенесите туда функционал “Пригласить друга”
 3. Добавьте Navigation Drawer. Разместите там два пункта: О приложении и Главный экран
 4. Добавьте анимацию при нажатии на картинку фильма, после которой будет открываться описание фильма
 5. * Создайте новый экран “Добавить фильм” и добавьте в Меню кнопку добавления фильма, по нажатию на которую открывается новый экран. На нем сделайте два поля: название фильма, описание. При возвращении на главный экран - добавляйте новый фильм к уже существующим (в качестве картинки используйте заглушку)
 6. * Самостоятельно изучите такие элементы как chips, bottom sheet, floating action button

P.S. Задание со звездочкой * - повышенной сложности. Если вы с ним не справитесь - ничего страшного, оно не является обязательным.

**5 Интерфейс
экрана.
Использование
стилей и
ресурсов**

Стили и темы, элементы управления, классификация ресурсов, векторы.

После занятия вы сможете: применять отдельные стили для всего приложения, конкретных экранов или элементов, создавать свои уведомления, создавать свою панель управления вверху экрана, а также узнаете, какие ресурсы в каких случаях использовать в своем приложении.

6 **Интерфейс экрана. Продвинутая верстка**

CoordinatorLayout, ConstraintLayout, Dialog.

После занятия вы сможете: использовать наиболее мощные и продвинутые контейнеры для верстки экранов, а также создавать кастомные диалоговые окна.

Домашние задания

1 Улучшение UI приложения по поиску фильмов.

1. Добавьте snackbar или toast в свое приложение
2. Используйте CoordinatorLayout или ConstraintLayout вместо текущих контейнеров
3. Добавьте на главный экран CollapsingToolbar, и когда он в развернутом состоянии - отображайте в нем какой-то задний фон
4. Перенесите весь хардкод в ресурсы приложения
5. Примените собственные стили и темы в приложении
6. Добавьте макет для приложения в ландшафтной ориентации
7. Добавьте какое-нибудь изображение в векторе в ваше приложение
8. Добавьте в NavigationDrawer пункт Выход и отображайте диалоговое окно выхода из приложения
9. * Самостоятельно изучите такие элементы UI как BottomBar и BottomSheetDialogFragment

P.S. Задание со звездочкой * - повышенной сложности. Если вы с ним не справитесь - ничего страшного, оно не является обязательным.

7 **Интерфейс экрана. Виджеты**

виджеты

После занятия вы сможете создавать разнообразные виджеты своего приложения на главном экране смартфона

8 Fragments

Жизненный цикл, FragmentManager, передача данных между Activity и Fragment'ом.

После занятия вы сможете: создавать экран в экране, несколько экранов рядом, передавать данные между несколькими экранами.

Домашние задания

1 Переход на единственную Activity и несколько Fragments. Создание виджета.

1. Переведите свое приложение на единственную Активити и несколько Фрагментов
2. * Используйте для навигации между фрагментами NavigationDrawer, BottomNavigation или BottomAppBar
3. * Используйте retain fragment без UI для сохранения состояния вашей Активити во время пересоздания
4. * Создайте виджет для своего приложения

P.S. Задание со звездочкой * - повышенной сложности. Если вы с ним не справитесь - ничего страшного, оно не является обязательным.

RecyclerView, layoutManager, Adapter, ViewHolder, ItemDecorator.

После занятия вы сможете: создавать списки и управлять ими, удалять и добавлять элементы, кастомизировать списки под свои нужды.

Домашние задания

1 Создание механизма “избранных” фильмов.

1. Переведите ваше приложение на отображение списков

2. Дополните функционал вашего приложения сохранением фильмов в список избранного (избранное пока храните в обычном ArrayList на уровне Активности).

Используйте для этого или долгое нажатие на элемент списка, или тап на ImageView в виде сердечка рядом с названием фильма

3. Создайте экран, где будет отображаться список Избранного. Сделайте так, чтобы в список можно было добавлять элементы и удалять их

4. * Самостоятельно изучите RecyclerView.ItemAnimator, создайте свои собственные анимации

P.S. Задание со звездочкой * - повышенной сложности. Если вы с ним не справитесь - ничего страшного, оно не является обязательным.

3 Работа с сетью и компоненты Android

1 Выходим в интернет

OkHttp, Retrofit, Gson, Glide.

После занятия вы сможете: загружать данные из интернета и передавать данные на удаленный сервер, загружать изображения.

2 **Полезные инструменты**

GPS и Location, Permissions, SharedPreferences.

После занятия вы сможете: определять местоположение телефона, использовать карты и навигацию, работать с пользовательскими разрешениями и сохранять данные на длительное время.

Домашние задания

1 Получение информации о фильмах из публичного API. Добавление функционала по навигации до ближайшего кинотеатра.

1. Подключите api сервера с фильмами и скачивайте фильмы с сервера

2. Используйте Glide для загрузки изображений

3. Добавьте функционал позиционирования для вашего приложения: на отдельном экране отображайте наличие кинотеатров в городе, в котором находится пользователь

4. Сохраняйте список избранного в SharedPreferences

5. * Добавьте функционал навигации до ближайшего кинотеатра, используя API карт на ваш выбор

6. * Продумайте, как будут загружаться фильмы, когда пользователь дойдет до конца списка, реализуйте свое решение

P.S. Задание со звездочкой * - повышенной сложности. Если вы с ним не справитесь - ничего страшного, оно не является обязательным.

3 **Компоненты Android. Часть 1**

Content Provider, SQLite.

После занятия вы сможете: обмениваться данными между приложениями и использовать базу данных для хранения больших объемов информации.

Service, работа в фоне и на переднем плане, Push. После занятия вы сможете: запускать длительные задачи, такие как загрузка изображений, в фоновом режиме и продолжать выполнение длительных задач, даже если пользователь свернул приложение.

Домашние задания

1 Добавление базы данных с фильмами в приложение.

1. Настройте базу данных в своем приложении и храните в ней все фильмы, ссылки на изображения

2. При заходе пользователя в приложение - обновляйте данные по фильмам в вашей БД (удаляйте все данные, заменяя их на новые, но не чаще раза в день), а если БД обновлена - загружайте данные из нее, а не из интернета

3. Загружайте данные в БД используя сервис

4. Отображайте пуш-уведомление для пользователя, что данные в БД обновлены. При тапе на уведомление - открывайте главный экран приложения

5. * Самостоятельно добавьте Swipe to refresh на главный экран, чтобы пользователь мог в любое время обновить данные (даже если в БД свежие данные)

6. * Самостоятельно настройте отправку и получение пуш-уведомлений через Firebase Cloud Messaging

P.S. Задание со звездочкой * - повышенной сложности. Если вы с ним не справитесь - ничего страшного, оно не является обязательным.

**5 Компоненты
Android. Часть
3**

BroadcastReceiver, LocalBroadcastReceiver.

После занятия вы сможете: принимать сообщения от операционной системы (например о подключении к wi-fi или к розетке), а также передавать сообщения внутри своего приложения.

1 Многопоточность в Android

Thread и AsyncTask, Handler и HandlerThread, Timer, AlarmManager, JobScheduler.

После занятия вы сможете: выполнять задачи приложения асинхронно и в фоновом режиме, выполнять задачи к определенному времени, а также подбирать способ асинхронного выполнения в соответствии с потребностями и контекстом.

Домашние задания

1 Отслеживание событий “есть интернет/нет”, “Есть wifi/нет”. Добавление функционала напоминания о выходе нового фильма.

1. Настройте BroadcastReceiver для получения событий: есть/нет интернета, есть/нет wi-fi

2. Используйте JobScheduler для обновления БД

3. Добавьте в ваше приложение функционал напоминания о выходе фильма или новой серии: пользователь должен ввести название фильма и дату, когда ему отобразится push-уведомление.

Добавьте в меню кнопку напомнить (реализуйте интерфейс напоминания в отдельном окне или диалоге), используйте AlarmManager для напоминания о выходе новой серии или фильма

4. * Используйте JobScheduler для регулярной синхронизации данных с сервером

P.S. Задание со звездочкой * - повышенной сложности. Если вы с ним не справитесь - ничего страшного, оно не является обязательным.

2 Тестирование в Android

Принципы разработки тестируемых приложений, test driven development, Unit-тесты.

После занятия вы сможете: писать Unit-тесты для своего приложения и понимать основы тестируемых приложений.

3 Тестирование для Android: Mockito

После занятия вы сможете: углубиться в тестирование и писать более обширные и сложные тесты для функционала приложения, создавать “моковые” объекты для тестирования.

Домашние задания

1 Написание Unit-тестов для приложения.

1. Напишите unit-тесты для своих классов, в том числе такие, которые используют Mockito
2. * Покройте тестами все значимые части приложения

P.S. Задание со звездочкой * - повышенной сложности. Если вы с ним не справитесь - ничего страшного, оно не является обязательным.

4 Публикация приложения

Builds, Flavors, Flavor dimensions, Play Store.

После занятия вы сможете: создавать различные версии своего приложения (например, платная и бесплатная версии), а также понимать, что требуется для публикации приложения в Play Store.

Домашние задания

1 Публикация в Google Play.

1. Выложите приложение в Google Play и обновляйте его уже там

2. * Подумайте, какой платный функционал может быть у вашего приложения и создайте для этого отдельный flavor

P.S. Задание со звездочкой * - повышенной сложности. Если вы с ним не справитесь - ничего страшного, оно не является обязательным.

5 Архитектурные компоненты. Реактивный подход. Внедрение зависимостей

1 Архитектурные компоненты. Часть 1

Понятие и примеры архитектуры приложения, ViewModel, LiveData, LifecycleOwner.

После занятия вы сможете: писать приложения в архитектурном паттерне MVVM и познакомитесь с архитектурными компонентами от разработчиков OS Android. Также вы будете понимать принципы построения архитектуры мобильных приложений

2 Архитектурные компоненты. Часть 2

Room, DataBinding.

После занятия вы сможете: использовать библиотеки от разработчиков OS Android для хранения данных (база данных Room) и для обновления данных непосредственно в макете XML (библиотека DataBinding)

Домашние задания

1. Переход на архитектурные компоненты. Обмен данными между фрагментами. Использование DataBinding.

1. Переведите свое приложение на архитектурные компоненты

2. * Обменивайтесь данными между фрагментами или фрагментом и активити используя ViewModel

3. * Используйте DataBinding в своем приложении

P.S. Задание со звездочкой * - повышенной сложности. Если вы с ним не справитесь - ничего страшного, оно не является обязательным.

3 **Архитектурные компоненты. Часть 3** WorkManager
После занятия вы сможете: выполнять любую фоновую или отложенную работу приложения в соответствии с видением разработчиков OS Android.

4 **RxJava. Часть 1** Концепция функционального программирования, Observable, Subscriber, основные операторы.
После занятия вы сможете: пользоваться самой мощной и популярной библиотекой для асинхронной работы приложения.

Домашние задания

1 WorkManager, Rx.

1. Обновляйте БД с помощью WorkManager
2. Выполняйте запросы в интернет используя Rx
3. * Переведите все запросы в БД на Rx
4. * Внедрите поиск по названию фильма в своем приложении и осуществляйте поиск на сервере или в БД через Rx

P.S. Задание со звездочкой * - повышенной сложности. Если вы с ним не справитесь - ничего страшного, оно не является обязательным.

5 **RxJava. Часть 2** onError, backpressure, Subject, продвинутые операторы.
После занятия вы сможете: использовать библиотеку RxJava на полную мощность, обрабатывать ошибки и подписываться на изменение объектов и данных.

6 Dagger 2. Часть 1

Основы: Module, Component.

После занятия вы сможете: упростить и автоматизировать внедрение зависимостей в своем приложении, сделать свое приложение более тестируемым.

Домашние задания

1 Использование Rx и Dagger.

1. В запросах на сервер через Rx - учитывайте и обрабатывайте ошибки запросов корректно, учитывайте backpressure

2. Используйте Даггер для инжекта SharedPreferences

3. * Доработайте поиск по названию используя debounce

4. * Самостоятельно изучите тему Scopes в Даггере, внедрите ViewModel с помощью Даггера и разделите ваши зависимости на скоупы

P.S. Задание со звездочкой * - повышенной сложности. Если вы с ним не справитесь - ничего страшного, оно не является обязательным.

7 Dagger 2. Часть 2

Практическое занятие: ответы на вопросы и разбор домашнего задания

8 Тестирование

- Тестируем Rx
- Тестируем Dagger

Домашние задания

1 Покрытие приложения тестами.

1. Доработайте свои тесты, чтобы они учитывали все нововведения вашего приложения, включая Dagger и Rx

1 Консультация по проекту

2 Защита проектов

Домашние задания

1 Сдача проекта

Пожалуйста, сдайте:

- ссылку на github-репозиторий с вашим проектом
- описание проекта