

# Вопросы к экзамену по разработке мобильных приложений

**В билете должно быть два теоретических и один практический вопрос**

## Теоретические вопросы

1. Язык Kotlin: особенности языка
2. Язык Kotlin: основные типы данных
3. Язык Kotlin: концепции и элементы языка для поддержки null-безопасности
4. Язык Kotlin: синтаксис и семантика условных выражений языка
5. Язык Kotlin: синтаксис и семантика циклов
6. Язык Kotlin: синтаксис и семантика создания обычных функций
7. Язык Kotlin: синтаксис и семантика создания lambda-функций
8. Язык Kotlin: функции высшего порядка, понятие и примеры
9. Язык Kotlin: массивы
10. Язык Kotlin: коллекции
11. Язык Kotlin: функции, предназначенные для преобразования коллекций
12. Язык Kotlin: функции, предназначенные для агрегирования коллекций
13. Язык Kotlin: способы создания sequence
14. Язык Kotlin: сравнение коллекций и sequence

15. Язык Kotlin: особенности реализации объектно-ориентированной парадигмы в структуре класса
16. Язык Kotlin: свойства (properties) класса
17. Язык Kotlin: enum и sealed classes
18. Язык Kotlin: особенности реализации объектно-ориентированной парадигмы в наследовании
19. Язык Kotlin: делегирование классов
20. Язык Kotlin: делегирование свойств, lazy, Delegates.observable
21. Язык Kotlin: поддержка параллельного программирования посредством корутин (запуск и получение ответов, контроль)
22. Язык Kotlin: передача данных между корутинами с помощью Flow
23. Язык Kotlin: диспетчеры, CoroutineScope
24. Сравнение функционального и императивного стиля в разработке
25. Принципы SOLID: принцип S
26. Принципы SOLID: принцип O
27. Принципы SOLID: принцип L
28. Принципы SOLID: принцип I
29. Принципы SOLID: принцип D
30. Язык Kotlin: Unit-тестирование
31. Язык Kotlin: data-классы и их сгенерированные методы
  1. Разработка приложений под Android: основные компоненты
  2. Разработка приложений под Android: методика создания приложений
  3. Разработка приложений под Android: структура проекта в Android Studio
  4. Разработка приложений под Android: роль и назначение файла Manifest.xml
  5. Разработка приложений под Android: настройка и использование Gradle в Android-разработке

6. Разработка приложений под Android: локализация приложения
7. Разработка приложений под Android: работа с базой данных, библиотека Room
8. Разработка приложений под Android: паттерн Repository (репозиторий)
9. Разработка приложений под Android: внедрение зависимостей с использованием Hilt
10. Архитектурный подход Clean Architecture: описание
11. Разработка приложений под Android: применение Clean Architecture
12. Разработка приложений под Android: принципы декларативного UI с использованием Jetpack compose
13. Разработка приложений под Android: компоненты Jetpack compose, предназначенные для компоновки
14. Разработка приложений под Android: компоненты Jetpack compose, предназначенные для взаимодействия с пользователем
15. Разработка приложений под Android: понятие, назначение и способ использования Navigation
16. Разработка приложений под Android: библиотека Retrofit
17. Разработка приложений под Android: Unit-тестирование (общие принципы и реализация)
18. Разработка приложений под Android: Unit-тестирование UseCase и ViewModel
19. Разработка приложений под Android: автоматическое UI-тестирование
20. Методы создания мобильных приложений (обзор всех методов)
21. Разработка приложений под Android: понятие, назначение и цикл жизни Activity
22. Архитектурный шаблон MVVM: описание
23. Разработка приложений под Android: реализация архитектурного шаблона MVVM

24. Разработка приложений под Android: использование StateFlow в MVVM
25. Разработка приложений под Android: особенности асинхронного программирования в Android
26. Разработка приложений под Android: отладка с использованием средств Android Studio
27. Разработка приложений под Android: понятие смены конфигурации и её учёт при программировании
28. Разработка приложений под Android: управление состоянием в Jetpack Compose
29. Разработка приложений под Android: обработка ошибок и отображение состояний (Loading, Success, Error) в MVVM + Compose
30. Разработка приложений под Android: кэширование данных с использованием Room и Retrofit
31. Разработка приложений под Android: модули проектов

## **Практические задания**

### **Задание 1**

Напишите программу решения квадратного уравнения с использованием паттерна MVVM для Android. Обязательно осуществляйте проверку корректности ввода.

### **Задание 2**

Напишите модель для программы решения квадратного уравнения и Unit-тесты для нее

### **Задание 3**

Напишите ViewModel для программы решения квадратного уравнения и Unit-тесты для нее

## Задание 4

Напишите программу решения квадратного уравнения и UI-тесты для нее

## Задание 5

Напишите программу решения квадратного уравнения с использованием MVVM и DI (Hilt)

## Задание 6

Напишите программу решения неравенства вида  $ax + b > 0$  с использованием паттерна MVVM для Android. Обязательно проверьте корректность ввода.

## Задание 7

Напишите модель для программы решения неравенства вида  $ax + b > 0$  и Unit-тесты для нее. Обязательно проверьте корректность ввода.

## Задание 8

Напишите ViewModel для программы решения неравенства вида  $ax + b > 0$  и Unit-тесты для нее. Обязательно проверьте корректность ввода.

## Задание 9

Напишите программу решения неравенства вида  $ax + b > 0$  и UI-тесты для нее. Обязательно проверьте корректность ввода.

## Задание 10

Напишите программу решения неравенства вида  $ax + b > 0$  с использованием MVVM и DI (Hilt). Обязательно проверьте корректность ввода.

## Задание 11

Напишите программу решения неравенства вида  $ax + b < 0$  с использованием паттерна MVVM для Android. Обязательно проверьте корректность ввода.

## Задание 12

Напишите модель для программы решения неравенства вида  $ax + b < 0$  и Unit-тесты для нее. Обязательно проверяйте корректность ввода.

## Задание 13

Напишите ViewModel для программы решения неравенства вида  $ax + b < 0$  и Unit-тесты для нее. Обязательно проверяйте корректность ввода.

## Задание 14

Напишите программу решения неравенства вида  $ax + b < 0$  и UI-тесты для нее. Обязательно проверяйте корректность ввода.

## Задание 15

Напишите программу решения неравенства вида  $ax + b < 0$  с использованием MVVM и DI (Hilt). Обязательно проверяйте корректность ввода.

## Задание 16

Напишите программу решения неравенства вида  $ax + b \geq 0$  с использованием паттерна MVVM для Android. Обязательно проверяйте корректность ввода.

## Задание 17

Напишите модель для программы решения неравенства вида  $ax + b \geq 0$  и Unit-тесты для нее. Обязательно проверяйте корректность ввода.

## Задание 18

Напишите ViewModel для программы решения неравенства вида  $ax + b \geq 0$  и Unit-тесты для нее. Обязательно проверяйте корректность ввода.

## Задание 19

Напишите программу решения неравенства вида  $ax + b \geq 0$  и UI-тесты для нее. Обязательно проверяйте корректность ввода.

## Задание 20

Напишите программу решения неравенства вида  $ax + b \geq 0$  с использованием MVVM и DI (Hilt). Обязательно проверяйте корректность ввода.

## Задание 21

Напишите программу решения неравенства вида  $ax + b \leq 0$  с использованием паттерна MVVM для Android. Обязательно проверяйте корректность ввода.

## Задание 22

Напишите модель для программы решения неравенства вида  $ax + b \leq 0$  и Unit-тесты для нее. Обязательно проверяйте корректность ввода.

## Задание 23

Напишите ViewModel для программы решения неравенства вида  $ax + b \leq 0$  и Unit-тесты для нее. Обязательно проверяйте корректность ввода.

## Задание 24

Напишите программу решения неравенства вида  $ax + b \leq 0$  и UI-тесты для нее. Обязательно проверяйте корректность ввода.

## Задание 25

Напишите программу решения неравенства вида  $ax + b \leq 0$  с использованием MVVM и DI (Hilt). Обязательно проверяйте корректность ввода.

## Задание 26

Создайте программу для создания заметок (заголовок+содержание) и их удаления. Используйте Navigation

## Задание 27

Создайте программу для создания заметок (заголовок+содержание) и их удаления, сохраняйте их посредством ROOM

## Задание 28

Реализуйте репозиторий для программы создания заметок (заголовок+содержание), произведите его Unit-тестирование

## Задание 29

Необходимо разработать клиент для взаимодействия с сервером, предоставляющим REST API для получения данных о музыкальных альбомах. Клиент должен выполнять следующие задачи: Для выполнения запросов к API необходимо использовать библиотеку Retrofit. Пример запроса:

```
GET http://10.0.2.2:9080/search?title=blue
```

Ответ от сервера возвращается в формате JSON и содержит данные о музыкальных альбомах:

```
[
{
  "id": "1
  "title": "Blue Train
  "artist": "John Coltrane
  "price": 56.99
}
]
```

## Задание 30

Создайте консольное приложение с использованием паттерна Decorator. Осуществите ввод информации о преподавателях и выведите тех, у кого средний балл учащихся выше 4. Преподаватель характеризуется ФИО, полом и средним баллом учащихся. Дополнительно:

- если имеет категорию — дата последнего подтверждения и номер приказа;
- если имеет кандидатскую степень — тема диссертации, научное направление, дата защиты.

Обязательно проверяйте корректность ввода.

## Задание 31

Создайте консольную программу для ввода заметок (название+содержание) и вывода всех заметок с использованием SOLID. Обязательно проверяйте корректность ввода.

## Задание 32

Создайте консольное приложение на Kotlin, реализующее следующие функции в чистом функциональном стиле (каждая — однострочная через точечные вызовы):

1. Для неотрицательного целого числа — сумма его цифр;
2. Для строки из слов — первый символ первого максимально длинного слова с нечётной длиной;
3. Для строки из неотрицательных целых чисел — множество цифр, присутствующих в каждом числе;
4. Для строки из целых чисел — побитовое И предпоследних цифр всех чисел (игнорируя числа, где такой цифры нет);
5. По номеру  $n$  —  $n$ -е число Фибоначчи (без рекурсии с экспоненциальной сложностью).

Приложение должно позволять выбирать задачу, вводить данные, проверять корректность ввода, выводить результат и повторять выполнение. Все функции должны быть чистыми.

## Задание 33

Напишите Android-приложение, которое по введённой пользователем дате определяет дату следующего дня. Отобразите её и проверьте, совпадает ли количество дней в месяце исходной даты с количеством дней в месяце полученной даты. Обязательно проверяйте корректность ввода и корректно обрабатывайте граничные случаи (конец месяца, високосный год и т.п.).

## Задание 34

Необходимо разработать клиент для взаимодействия с сервером, предоставляющим REST API для получения данных о гражданах. Для выполнения запросов к API необходимо использовать библиотеку Retrofit. Пример запроса:

```
GET http://10.0.2.2:9080/search?name=john
```

Ответ от сервера возвращается в формате JSON и содержит данные о гражданах:

```
[  
{
```

```
"id": "1"
"name": "John Smith"
"age": 35,
"address": "123 Main St"
}
]
```

### Задание 35

Создайте консольное приложение с использованием паттерна Decorator. Осуществите ввод информации об электронных книгах и выведите те, что стоят более 3000 рублей. Книга характеризуется моделью, производителем и стоимостью. Дополнительно:

- при дисплее на электронных чернилах — цветность (ч/б или цветной) и наличие подсветки;
- при сенсорном интерфейсе — тип сенсора (ёмкостный, резистивный и т.д.).

Обязательно проверяйте корректность ввода.

### Задание 36

Создайте консольное приложение на Kotlin, реализующее следующие функции в чистом функциональном стиле (каждая — однострочная через точечные вызовы):

1. Для неотрицательного целого числа — сумма его нечётных цифр;
2. Для строки из слов, разделённых одним или несколькими пробелами — последний символ первого максимально длинного слова с нечётным числом символов;
3. Для строки из неотрицательных целых чисел, разделённых одним пробелом — множество цифр, присутствующих хотя бы в двух числах;
4. Для строки из целых чисел — побитовое ИЛИ предпоследних цифр всех чисел (игнорируя числа, где такой цифры нет);
5. По заданному числу Фибоначчи найдите его номер в последовательности (не используйте факты, которые вы не можете доказать самостоятельно; допускается только линейное или логарифмическое решение без предварительного знания индекса).

Приложение должно позволять выбирать задачу, вводить данные, проверять корректность ввода, выводить результат и повторять выполнение. Все функции должны быть чистыми.

### **Задание 37**

С клавиатуры вводится информация о студентах: фамилия, имя, оценки. Выведите на экран информацию о трёх лучших студентах по среднему баллу. В случае, если у нескольких студентов средний балл совпадает, то выведите большее число студентов (пока не будут выведены все студенты или не будут полностью исчерпаны студенты с тремя лучшими баллами). Вывод надо осуществлять в порядке убывания среднего балла, а для одинаковых средних баллов — в алфавитном порядке по фамилии и имени. Обязательно проверяйте корректность ввода.

### **Задание 38**

С клавиатуры вводится информация о студентах: фамилия, имя, оценки. Выведите на экран информацию о трёх лучших студентах по максимальному баллу. В случае, если у нескольких студентов максимальный балл совпадает, то выведите большее число студентов (пока не будут выведены все студенты или не будут полностью исчерпаны студенты с тремя лучшими баллами). Вывод надо осуществлять в порядке убывания максимального балла, а для одинаковых максимальных баллов — в алфавитном порядке по фамилии и имени. Обязательно проверяйте корректность ввода.

## **Ссылки**

<https://developer.android.com/>

<https://kotlinlang.ru/docs/kotlin-doc.html>