

Контрольная работа.
«Ключевые аспекты Android-разработки»

Цель: Систематизировать и закрепить практические навыки и теоретические знания по основным темам Android-разработки.

Задание: Каждый вариант контрольной работы включает теоретическое объяснение и практическую часть задачи. Необходимо описать концепцию и привести фрагмент кода, соответствующий условиям задания, а также объяснить ключевые моменты.

Номер варианта - соответствует последней цифре Вашего пароля.

Вариант 1: Асинхронное обновление интерфейса через Handler

Опишите, как в Android использовать Handler и Runnable для регулярного обновления интерфейса (например, отображение таймера или счетчика).

Приведите фрагмент кода, где:

- Получается экземпляр Looper основного потока.
- Запускается задача с задержкой (например, обновление текста каждые N секунд).
- Интерфейс обновляется с использованием postDelayed() (например, вывод текущего времени или числа).
- Объясните, как избежать утечек памяти при использовании Handler.

Вариант 2: Использование функциональных методов коллекций

Объясните, как работают функциональные методы коллекций (filter, map, groupBy), приведите примеры их применения в реальных сценариях. Напишите фрагмент кода на Kotlin, который:

- Создает список пользователей в виде List<Map<String, Any>>, где каждый элемент содержит ключи "name" и "age".
- Фильтрует пользователей старше 25 лет и преобразует их имена в верхний регистр.
- Отображает эти имена в RecyclerView.
- Объясните, почему mutableListOf предпочтительнее listOf для динамических данных в интерфейсе.

Вариант 3: Архитектура приложения (MVVM + Room)

Объясните, как реализовать паттерн MVVM в Android-приложении с использованием Room. Приведите пример:

- Создания Entity-класса для хранения данных пользователя с полями id, name, age.
- DAO-интерфейса с методом получения всех пользователей,
- ViewModel, которая предоставляет данные для UI, используя Room.
- Отображения данных в TextView.

Вариант 4: Асинхронная загрузка данных (ExecutorService)

Объясните, как с помощью ExecutorService выполнить фоновое форматирование данных датчика освещённости и безопасно обновить UI. Приведите пример, в котором:

- Данные с датчика обрабатываются в потоке,
- Результат отображается в TextView через runOnUiThread.
- Задача отменяется при уничтожении Activity,
- • Объясните, почему ExecutorService предпочтительнее Thread для таких задач.

Вариант 5: Управление жизненным циклом активности

Опишите, как правильно регистрировать и отменять регистрацию датчика в жизненном цикле Activity. Приведите пример реализации методов onResume() и onPause(), где:

- В onResume() происходит регистрация слушателя датчика,
- В onPause() — отмена регистрации для экономии ресурсов.
- Объясните, почему важно отменять регистрацию датчика,
- Приведите пример проверки доступности датчика.

Вариант 6: View Binding

Сравните использование findViewById() и View Binding.

- Приведите пример, как с помощью View Binding получить ссылку на Button и установить ему обработчик нажатия.
- Объясните преимущества View Binding в плане безопасности типов и производительности.
- Приведите пример использования View Binding в Activity.

Вариант 7: Работа с коллекциями и лямбдами

Опишите, как в Kotlin с помощью стандартных функций-расширений (filter, map, sortBy, forEach) можно обработать список пользователей. Приведите фрагмент кода, где:

- Создается список пользователей (например, List<Map<String, Any>>), где каждый элемент содержит "name" и "age".
- Фильтруются только совершеннолетние пользователи (age >= 18).
- Имена этих пользователей преобразуются в верхний регистр.
- Объясните, как обрабатывать null-значения в коллекции (например, если "name" или "age" отсутствуют).

Вариант 8: Работа с Intent

Объясните разницу между явным и неявным Intent. Приведите пример использования:

- Явного Intent для перехода между двумя Activity внутри одного приложения,
- Неявного Intent для запуска браузера с конкретным URL.
- Передачи данных между Activity через putExtra(),
- Проверки доступности действия для неявного Intent.

Вариант 9: Корутины в Android

Опишите, как с помощью корутин можно организовать бесконечный цикл обновления данных с датчика акселерометра. Приведите фрагмент кода, где:

- Запускается корутина в основном потоке,
- Происходит периодическое считывание данных,
- Интерфейс обновляется с использованием Dispatchers.Main.
- Корутина отменяется при уничтожении Activity.

Вариант 0: Интеграция Retrofit и Moshi

Опишите этапы интеграции Retrofit и Moshi для получения JSON-данных с сервера. Приведите пример:

- Определения интерфейса API с аннотацией @GET,
- Настройки Retrofit с Moshi в качестве конвертера,
- Вызова API в корутине и отображения результата в TextView.
- Объясните, как кэшировать данные с помощью SharedPreferences.

Методические указания к выполнению контрольной работы

Контрольная работа направлена на закрепление теоретических знаний и практических навыков в области Android-разработки. Каждый вариант включает теоретическую и практическую части. Для успешного выполнения работы необходимо следовать перечисленным требованиям:

- Теоретическая часть должна содержать краткое, но полное объяснение концепций, связанных с заданием, в объеме: 1–2 абзаца (150–300 слов).
- Код в практической части должен быть выделен в отдельные блоки с комментариями, поясняющими ключевые моменты реализации (10–20 строк кода на Kotlin).
- Включите в ответ к практической части определения ключевых терминов (например, что такое Handler, ExecutorService, MVVM, Retrofit) и принципы работы (например, как Handler взаимодействует с Looper, как ViewModel сохраняет данные при изменении конфигурации).

Требования к отчету

Результаты выполненной контрольной работы должны быть оформлены в формате текстового редактора Word, размером шрифта 14 пунктов и включать:

1. Титульный лист.
2. Текст задания, соответствующий Вашему варианту.
3. Теоретическое объяснение концепции (1–2 абзаца, 150–300 слов).
4. Практическую реализацию с кодом (10–20 строк Kotlin, выделенных в отдельные блоки).
5. Объяснение ключевых моментов кода.