

Тестовое задание для кандидатов на позицию Android Developer

Основное задание

Используя серверное API <https://pokeapi.co/>, написать Android-приложение в котором:

1. Отобразить список из 30 покемонов на главном экране. Каждый элемент списка должен как минимум содержать название покемона и его изображение.
2. Добавить возможность просмотра детальной информации о покемоне. Для этого сделать отдельный экран, на который можно попасть, кликнув по любому элементу списка покемонов.
3. Детальная информация как минимум должна включать следующие поля: рост (height), вес (weight), род покемона (type, например: птица, насекомое, ядовитый и т. д.), информация (stats; включить поля attack, defense, hp).
4. Реализовать постраничную подгрузку элементов списка покемонов следующим образом: во время скроллинга списка при достижении его конца загрузить следующие 30 элементов и отобразить в списке.
5. Добавить кнопку на главный экран, при нажатии на которую список покемонов будет заново проинициализирован с началом на произвольном элементе из базы данных покемонов на сервере. При этом минимальная длина списка всегда должна быть не меньше 30 элементов.
6. Добавить 3 флажка (CheckBox) на главный экран с надписями attack, defense и hp, реализующих следующее поведение. При активации флажка attack/defense/hp, осуществить поиск покемона с самым большим показателем в поле attack/defense/hp среди покемонов из текущего списка. Переместить найденного покемона в начало списка, проскроллить список вверх, если это необходимо, визуально выделить данный элемент списка. Если выделены два или три флажка, осуществить поиск покемона, у которого выбранные показатели больше остальных. Если невозможно однозначно определить самого сильного покемона по выбранным показателям, поступить на своё усмотрение.

Дополнительные задания

1. Дополнение к пункту 5: выполнить поиск самого сильного покемона по одному выбранному показателю наиболее оптимальным способом.
2. Вместо поиска (пункт 5) можно сделать сортировку текущего списка покемонов по выбранным показателям.
3. Реализовать кеширование уже загруженных с сервера элементов списка таким образом, чтобы в случае отсутствия интернет-соединения при открытии приложения была доступна вся его функциональность.

Обязательные требования

Код выполненного задания должен быть структурирован, легко читаем и содержать необходимые комментарии. Вся прилагаемая документация должна быть на английском языке. Важно: тестовое задание даётся для проверки у Вас некоторых вполне определенных знаний и умений, и в число этих умений не входит умение «делать примитивное решение на скорую руку, потому что задача простая, зачем лишние сложности». Требования из формулировки к заданию необходимо соблюсти все, независимо от того, кажутся они вам целесообразными применительно к данной задаче, или нет. После основной задачи к ней предложены три дополнительных задания. Они не обязательны для выполнения, однако существенно добавляют очков при проверке. Но стоит помнить, что лучше сделать немного, но хорошо, чем много, но кое-как.

Требования к технологиям

- Язык Java или Kotlin
- Система сборки Gradle
- Интерфейс простой, но аккуратный
- Для списков использовать RecyclerView
- Минимальная версия Android — любая
- Можно использовать библиотеки

Время на выполнение тестового задания

В среднем выполнение подобной задачи у разработчика уровня Middle может занять до 8 часов, у разработчика уровня Junior — до 16 часов.

Публикация

Проект должен быть опубликован в приватном репозитории на bitbucket.org.