

ДЗ 1

Цель задания

В этом домашнем задании вы поработаете на практике с потоками в приложении, выполните длительную работу в фоновом потоке, а также распараллелите её на несколько потоков.

Вы научитесь работать с инструментами, которые предоставляет Android для работы с многопоточностью: `Looper`, `MessageQueue`, `Handler`.

Что нужно сделать

1. Подтяните проект в папке **Multithreading** из репозитория **learning_materials / android_basic**.
 - В проекте настроено получение из сети фильма по его идентификатору.
 - Фильм представлен классом `Movie` в проекте.
2. Получите API_KEY по ссылке: <http://www.omdbapi.com/apikey.aspx>, выберите Account Type FREE.
3. Вставьте полученный API_KEY в константу `MOVIE_API_KEY` файла `Network.kt`.
4. Реализуйте `ThreadingFragment`, чтобы в нём находился `RecyclerView`, который отображает список фильмов. `Adapter` должен работать с объектами класса `Movie`.
5. Получите информацию по фильмам последовательно в главном потоке: «Матрица», «Бойцовский клуб», «Форрест Гамп», «Крёстный отец», «Властелин колец: Две крепости». Или по любым другим фильмам по вашему усмотрению. Для этого найдите идентификатор IMDB фильмов: перейдите на сайт <https://www.imdb.com>, найдите фильм в разделе Top rated movies. Перейдите в детальную информацию по фильму. В адресной строке будет URL вида <https://www.imdb.com/title/tt0167261>, выделенная часть — и есть идентификатор фильма.

Для получения информации о фильме из сети используйте метод `Network.getMovieById`.

6. Посмотрите, какая ошибка была выброшена при получении данных из сети в главном потоке, сохраните скриншот ошибки в корне проекта.
7. Получите информацию по фильмам в фоновых потоках параллельно, используя класс `Thread`. Каждый фильм запрашивается одним потоком. Отобразите полученные фильмы во фрагменте. Для оповещения о завершении получения фильмов используйте механизм колбеков.
8. Используйте `Handler`, связанный с `MessageQueue` главного потока, чтобы установить полученный список фильмов в адаптер.
9. Отобразите тост «Список обновлён» через одну секунду после получения списка фильмов. Для этого используйте `Handler`.
10. *Используйте `Thread pool` вместо потоков, чтобы получить фильмы из сети. Количество потоков должно равняться количеству ядер у процессора вашего устройства. Для этого изучите статью из дополнительных материалов.

11. *Добавьте SwipeRefreshLayout на экран со списками фильмов. Когда пользователь тянет список вниз, фильмы должны обновляться из сети.

Рекомендации

Используйте репозиторий **learning_materials/android_basic**.

Скачайте изменения в репозитории на локальную машину.

Выполните домашнее задание в папке **Multithreading**.

Отправьте коммиты в удалённый репозиторий.

Критерии оценки

- Код оформлен в соответствии с правилами <https://kotlinlang.org/docs/reference/coding-conventions.html>.
- Соблюждён принцип инкапсуляции с помощью модификаторов доступа.
- Классы являются не финальными (open, abstract) только при необходимости.
- Текстовые строки не захардкожены и используются из ресурсов.
- Получение информации по фильмам происходит в фоновых потоках.
- В проекте отсутствуют race condition, deadlock, livelock.
- Выполнены все пункты задания.

Дополнительные материалы

<https://android.jlelse.eu/executor-framework-understanding-the-basics-43d575e72310> — работа с Executor.

<https://medium.com/@frank.tan/using-a-thread-pool-in-android-e3c88f59d07f> — Using a Thread Pool in Android.

<https://developer.android.com/training/swipe/add-swipe-interface> — работа с SwipeRefreshLayout.