МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»



**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

по выполнению лабораторных работ

по дисциплине «Программирование мобильных устройств»

для студентов направлений 09.03.03 «Прикладная информатика» и 09.03.02 «Информационные системы и технологии».

Ставрополь

2021

# **Лабораторная работа 11. Работа с потоками.**

**Цель работы:** научиться распределять задачи между потоками.

**Формируемые компетенции:** ПК-7

**Теоретическая часть:**

Каждый процессор может обрабатывать несколько потоков одновре­менно. Приложения, написанные ранее, выполнялись в одном потоке, кото­рый называется главным. Все стандартные вычисления выполняются процес­сором в главном потоке. Также, этот поток отвечает за обновление пользова­тельского интерфейса.

Неправильная работа с этим потоком может привести к тому, что при­ложение станет "не отвечать". Это частая проблема, которую начинающие разработчики не берут во внимание.

Чтобы избежать данной проблемы, существует подход, названный "асинхронное программирование". Данный метод позволяет решать все ма­лозначимые задачи в отдельном потоке, не загружая основной, и не "замора­живая" интерфейс.

Естественно, в андроид такая возможность присутствует. Делается это с помощью генерации нового потока Thread.

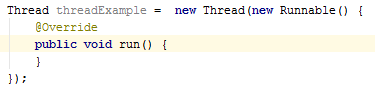


Рисунок 1 – Генерация нового потока

В переопределённом методе run() необходимо записать код, который будет выполняться в отдельном потоке.

Рассмотрим простой пример, который загружает изображение из ин­тернета и помещает в ImageView. Для этого выполним следующие шаги:

* 1. Создадим кнопку, которая будет загружать картинку:

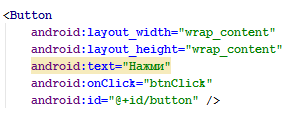


Рисунок 2 – Создание кнопки

* 1. Создадим ImageView, который будет отображать изображение:

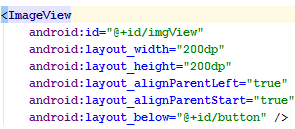


Рисунок 3 – Создание ImageView

* 1. Чтобы приложение могло воспользоваться интернетом, необходимо добавить соответствующее разрешение в AndroidManifest.xml. Что такое разрешения и как они работают разберёмся в другой лабораторной работе.

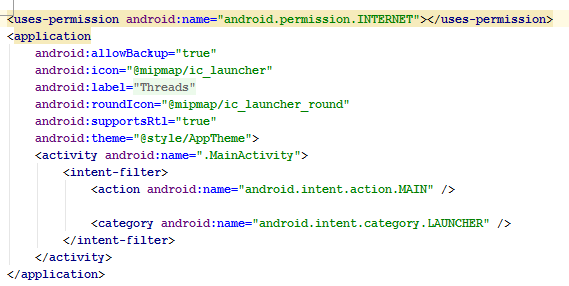


Рисунок 4 – Добавление разрешения

* 1. Создадим отдельно метод для загрузки изображения и метод для помещения его в ImageView.

Запрос на загрузку файла подразумевает обработку входного потока, который мы получаем с помощью метода getContent(). С помощью decodeStream() мы распознаём поток байт в изображение формата bmp.

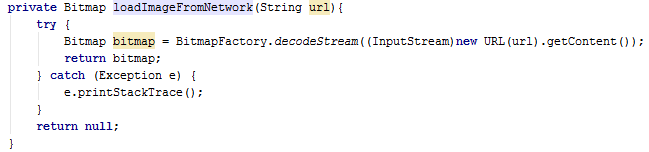


Рисунок 5 – Метод для загрузки изображения из интернета

Возьмём в качестве примера логотип СКФУ, который доступен для скачивания на официальном сайте по следующему URL-адресу: http://www.ncfu.ru/templates/current/images/logotype.png

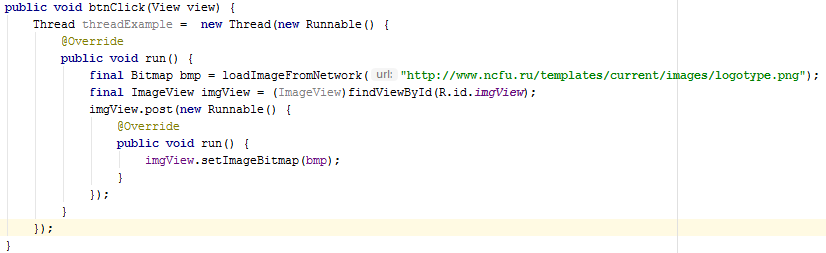


Рисунок 6 – Вызов метода загрузки

Определяя переменные константами (final) мы запрещаем возможные изменения объектов в процессе обработки метода.

Метод post() позволяет асинхронно изменить содержимое нашего ImageView. Таким образом, мы создали действие, которое будет выполняться в отдельном потоке (Thread). Чтобы заставить выполнить этот код, требуется вызвать метод start() у threadExample.



Рисунок 7 – Запуск потока

Второй способ использования параллельного потока более приближен к реалиям современной разработки ПО. AsyncTask ­– именно этот класс мы будем расширять создания нового потока.

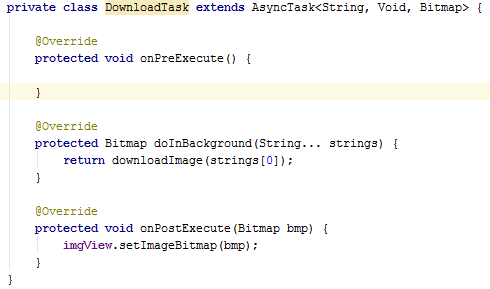


Рисунок 8 – Пример создания асинхронного задания

Наследуясь от класса AsyncTask необходимо переопределить следую­щие методы:

onPreExecute() – метод, который выполняется перед запуском основной задачи. Здесь можно изменить интерфейс, чтобы показать, что скоро нач­нётся выполняться асинхронный метод (например, запустить прогрессбар).

doInBackground() – в этом методе будет выполняться код в отдельном потоке. Следует отметить, что здесь нельзя трогать интерфейс, иначе, Вы рискуете снова нарваться на "Приложение не отвечает". Результат выполне­ния этого метода будет передан в onPostExecute().

onProgressUpdate() – метод, который позволяет вручную устанавливать прогресс выполнения метода doInBackground().

onPostExecute() – после выполнения метод doInBackground() выпол­нится код, написанный здесь. Здесь можно снова обновить интерфейс.

Если Вам не нужен какой-нибудь метод, то можете просто не переоп­ределять его. Параметры AsyncTask тоже подобраны не случайно. Первый тип – показывает тип параметра, который будет передан в метод doInBackground() посредством execute(). Второй тип – передаваемый тип в onProgressUpdate(). И последний – Тип параметра, переданного по окончании doInBackground и полученного в onPostExecute().

Воспользуемся той же структурой xml-файла и заменим лишь метод onClick().



Рисунок 9 – Выполнение задания

**Оборудование и материалы:** для выполнения данной лабораторной работы необходим компьютер с установленной операционной системой Windows 7 (8,8.1,10) и программными продуктами: AndroidStudio.

**Указания по технике безопасности:** к выполнению лабораторных работ допускаются студенты, ознакомившиеся с правилами работы в лаборатории, прошедшие инструктаж безопасности.

**Задания:** для выполнения лабораторной работы необходимо выполнить следующее:

1. Рассчитать результат и отобразить его в интерфейсе.
2. Загрузить несколько любых изображений из интернета с отображением строки состояния (ProgressBar).

**Варианты для задания 1:**

1. В одномерном массиве, состоящем из n вещественных элементов вычислить:
   1. Сумму отрицательных элементов массива
   2. Произведение элементов массива, расположенных между максимальным и минимальным элементом этого массива.
2. В одномерном массиве, состоящем из n вещественных элементов вычислить:
   1. Сумму положительных элементов массива.
   2. Произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементом этого массива.
3. В одномерном массиве, состоящем из n целых элементов вычислить:
   1. Произведение элементов массива с четными индексами.
   2. Сумму элементов массива, расположенных между первым и последним нулевыми элементами.
4. В одномерном массиве, состоящем из n вещественных элементов вычислить:
   1. Сумму элементов с нечетными индексами.
   2. Сумму элементов массива, расположенных между первым и последним отрицательными элементами.
5. В одномерном массиве, состоящем из n вещественных элементов вычислить:
   1. Максимальный элемент массива.
   2. Сумму элементов массива, расположенных до последнего положительного элемента.
6. В одномерном массиве, состоящем из n вещественных элементов вычислить:
   1. Минимальный элемент массива
   2. Сумму элементов массива, расположенных между первым и последним положительными элементами.
7. В одномерном массиве, состоящем из n вещественных элементов вычислить:
   1. Номер максимального элемента массива.
   2. Произведение элементов массива, расположенных между первым и вторым нулевыми элементами.
8. В одномерном массиве, состоящем из n вещественных элементов вычислить:
   1. Номер минимального элемента массива.
   2. Сумму элементов массива, расположенных между первым и вторым отрицательными элементами.
9. В одномерном массиве, состоящем из n вещественных элементов вычислить:
   1. Максимальный по модулю элемент массива.
   2. Сумму элементов массива, расположенных между первым и вторым положительными элементами.
10. В одномерном массиве, состоящем из n вещественных элементов вычислить:
    1. Минимальный по модулю элемент массива.
    2. Сумму модулей элементов массива, расположенных после первого элемента, равного нулю.
11. В одномерном массиве, состоящем из n вещественных элементов вычислить:
    1. Номер минимального по модулю элемента массива.
    2. Сумму модулей элементов массива, расположенных после первого отрицательного элемента.
12. В одномерном массиве, состоящем из n вещественных элементов вычислить:
    1. Номер максимального по модулю элемента массива.
    2. Сумму элементов массива, расположенных после первого положительного элемента.
13. В одномерном массиве, состоящем из n вещественных элементов вычислить:
    1. Количество элементов, лежащих в диапазоне от A до B.
    2. Сумму элементов массива, расположенных после максимального элемента.
14. В одномерном массиве, состоящем из n вещественных элементов вычислить:
    1. Количество элементов массива, равных нулю.
    2. Сумму элементов массива, расположенных после минимального элемента.

**Содержание отчета:** отчет по лабораторной работе должен быть выполнен в редакторе MSWord и оформлен согласно требованиям. Требования по форматированию: Шрифт TimesNewRoman, интервал – полуторный, поля левое – 3 см., правое – 1,5 см., верхнее и нижнее – 2 см. Абзацный отступ – 1,25. Текст должен быть выровнен по ширине.

Отчет должен содержать титульный лист с темой лабораторной работы, цель работы и описанный процесс выполнения вашей работы. В конце отчета приводятся выводы о проделанной работе.

В отчет необходимо вставлять скриншоты выполненной работы и добавлять описание к ним. Каждый рисунок должен располагаться по центру страницы, иметь подпись (Рисунок 1 – Создание подсистемы) и ссылку на него в тексте.

**Контрольные вопросы:**

1. Поток.
2. Task.

**Список литературы, рекомендуемый к использованию по данной теме:**

1. Пархимович М. Н. , Липницкий А. А. , Некрасова В. А. Основы интернет-технологий: учебное пособие / Пархимович М. Н. , Липницкий А. А. , Некрасова В. А. – М.: Архангельск: ИПЦ САФУ, 2013 – 366 с.; То же [Электронный ресурс]. - URL:http://biblioclub.ru/index.php?page=book\_red&id=436379&sr=1
2. Соколова В. В. Разработка мобильных приложений: учебное пособие/ Соколова В. В. Томск: Издательство Томского политехнического университета, 2015 – 176с. То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book\_red&id=442808&sr=1