|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практической работе**

по дисциплине «Разработка мобильных приложений»

|  |  |
| --- | --- |
| **Выполнили:**  Студенты группыИНБО-09-21 | Грабов И.В. |
| **Проверил:**  Ассистент | Новичков Д.Е. |

Москва 2023 г.

**Цель работы:** Создать проект в Android Studio. Изучить материал и написать код с использованием тостов и выводов логов.

**Выполнение работы:**

Приложения для Android могут быть написаны с использованием языков Kotlin, Java и C ++. Инструменты Android SDK компилируют ваш код вместе с любыми файлами данных и ресурсов в APK или пакет приложений для Android. Пакет Android, представляющий собой архивный файл с расширением .apk. Этот файл содержит необходимые для работы приложения ресурсов и используется для установки приложений в ОС Android. Пакет приложений для Android, представляющий собой архивный файл с расширением .aab, содержит содержимое проекта приложения для Android, включая некоторые дополнительные метаданные, которые не требуются во время выполнения. AAB - это формат публикации, который нельзя установить на устройства Android, он откладывает создание и подписание APK на более поздний этап. Например, при распространении вашего приложения через Google Play серверы Google Play генерируют оптимизированные APK-файлы, содержащие только те ресурсы и код, которые требуются конкретному устройству, запрашивающему установку приложения.

Класс Toast предоставляет простую обратную связь об операции в небольшом всплывающем окне. Он заполняет только пространство, необходимое для сообщения, а текущая активность остается видимой и интерактивной. Toast’ы автоматически исчезают по истечении тайм-аута.

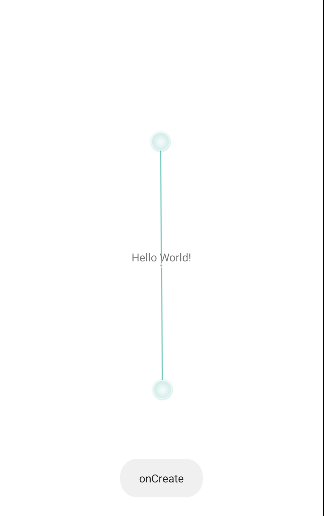
****

Рисунок 1 – Toast “onCreate”

Метод onCreate() вызывается при создании или перезапуска Activity. Система может запускать и останавливать текущие окна в зависимости от происходящих событий. Внутри данного метода настраивают статический интерфейс Activity. Инициализирует статические данные Activity, связывают данные со списками и т.д. Связывает с необходимыми данными и ресурсами. Задаёт внешний вид через метод setContentView().

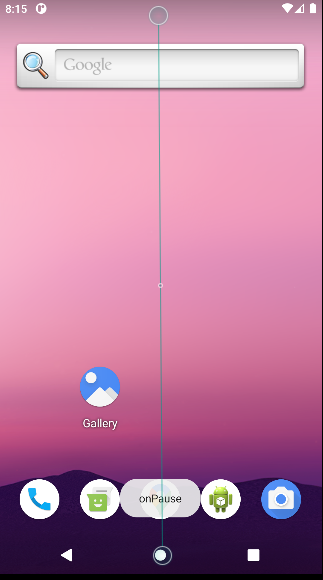
****

Рисунок 2 – Toast “onPause”

Когда пользователь решает перейти к работе с новым окном, система вызовет для прерываемого окна метод onPause(). По сути происходит свёртывание активности. Сохраняет незафиксированные данные. Деактивирует и выпускает монопольные ресурсы. Останавливает воспроизведение видео, аудио и анимацию. От onPause() можно перейти к вызову либо onResume(), либо onStop().

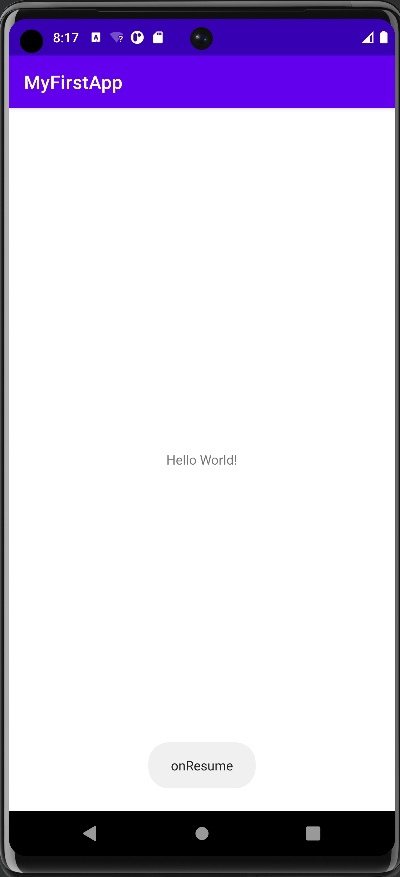


Рисунок 3 – “Toast onResume”

Метод onResume() вызывается после onStart(), даже когда окно работает в приоритетном режиме и пользователь может его наблюдать. В этот момент пользователь взаимодействует с созданным вами окном. Приложение получает монопольные ресурсы. Запускает воспроизведение анимации, аудио и видео. Также может вызываться после onPause().

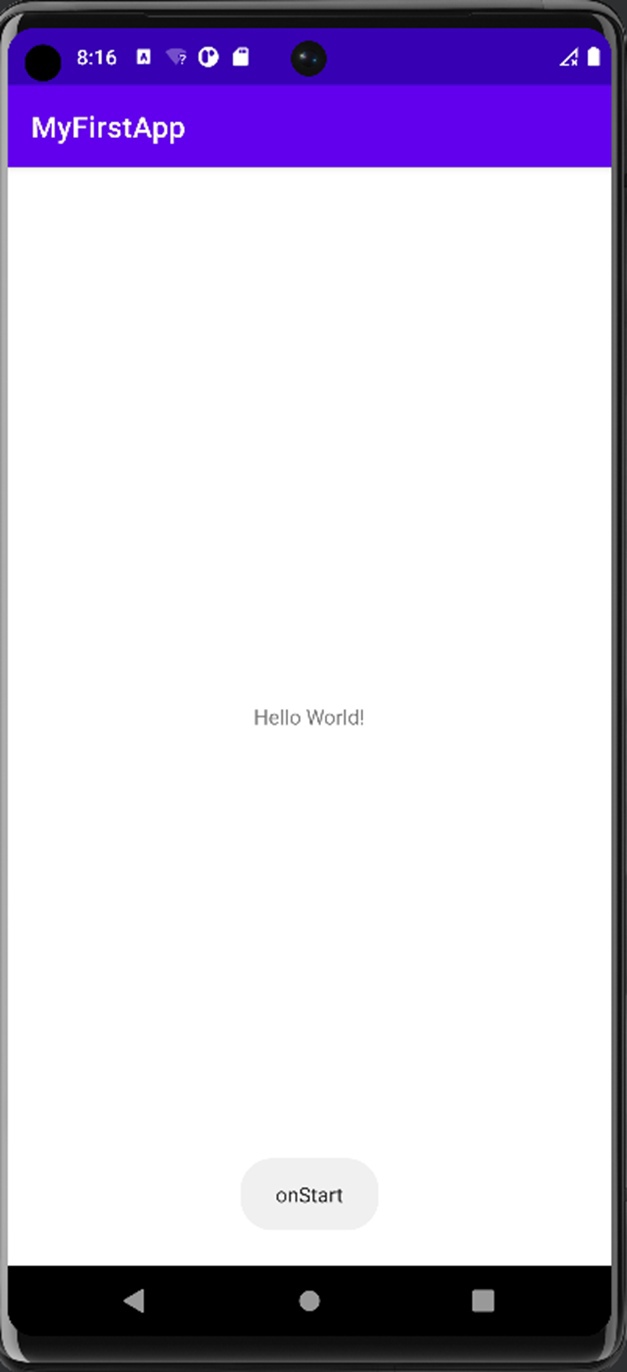


Рисунок 4 – “Toast onStart”

За onCreate() всегда следует вызов onStart(), но перед onStart() не обязательно должен идти onCreate(), так как onStart() может вызываться и для возобновления работы приостановленного приложения (приложение останавливается методом onStop()). При вызове onStart() окно ещё не видно пользователю, но вскоре будет видно. Вызывается непосредственно перед тем, как Activity становится видимой пользователю. Сопровождается вызовом метода onResume(), если Activity получает передний план, или вызовом метода onStop(), если становится скрытой.



Рисунок 5 – “Toast Destroyed”

Метод вызывается по окончании работы Activity, при вызове метода finish() или в случае, когда система уничтожает этот экземпляр активности для освобождения ресурсов. Эти два сценария уничтожения можно определить вызовом метода isFinishing(). Вызывается перед уничтожением Activity. Это последний запрос, который получает активность от системы. Если определённое окно находится в верхней позиции в стеке, но невидимо пользователю и система решает завершить это окно, вызывается метод onDestroy(). В этом случае метод удаляет все статические данные Activity. Отдаёт все используемые ресурсы.

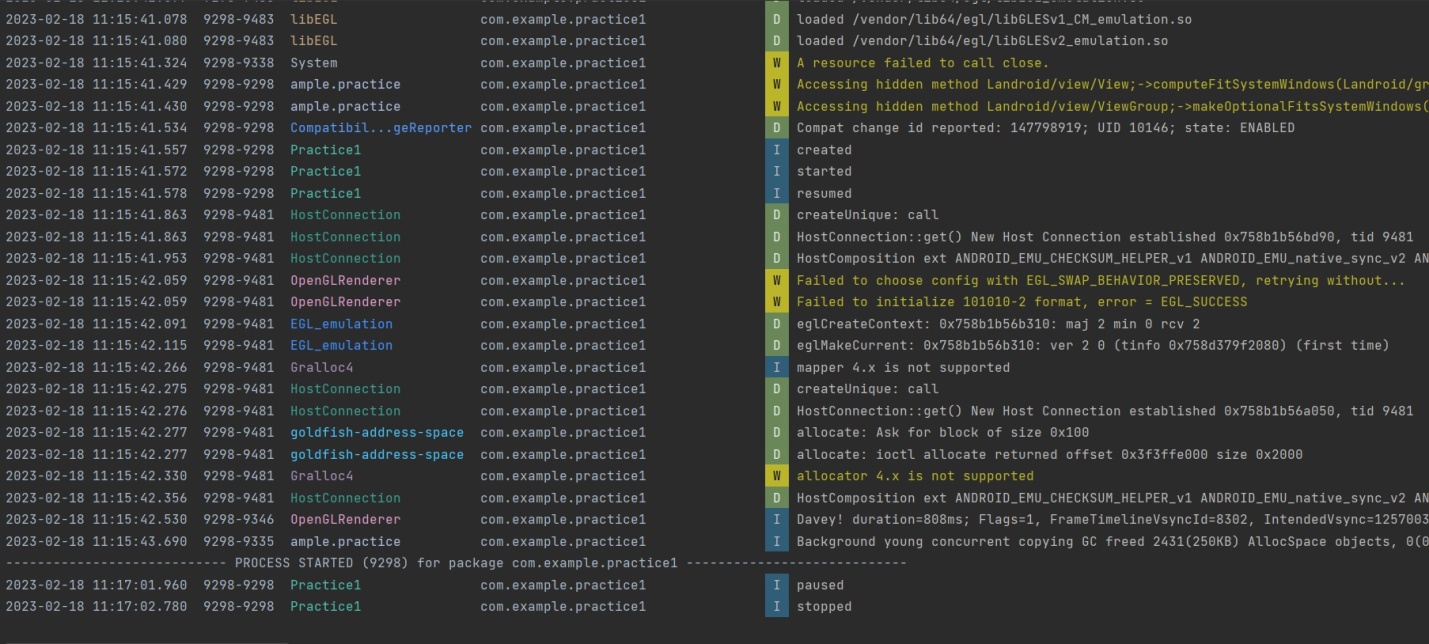


Рисунок 6 – Logcat

Окно Logcat в Android Studio помогает отлаживать ваше приложение, отображая журналы с вашего устройства в режиме реального времени — например, сообщения, которые вы добавили в свое приложение с помощью класса, сообщения от служб, работающих на Android, или системные сообщения, например, при сборке мусора. Когда приложение выдает исключение, Logcat показывает сообщение, за которым следует соответствующая трассировка стека, содержащая ссылки на строку кода. Очень часто программисту нужно вывести куда-то промежуточные результаты, чтобы понять, почему программа не работает. Особо хитрые временно размещают на экране текстовую метку и выводят туда сообщение при помощи метода textView.setText(" "). Но есть способ лучше. В Android есть специальный класс android.util.Log для подобных случаев.

**Вывод**

В ходе решения поставленной задачи было написано приложение в Android Studio. Получены навыки работы с Android Studio. Был изучен материал и написан код с использованием тостов и выводов логов.