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

**Институт информационных технологий (ИИТ)**

**Кафедра математического обеспечения и стандартизации ИТ**

**ОТЧЁТ ПО ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №10**

**по дисциплине**

**«Разработка мобильных приложений»**

|  |  |  |  |
| --- | --- | --- | --- |
| Отчет представлен к  рассмотрению:  Студенты группы ИНБО-04-20 | « » февраля 2022 г. | (подпись) | Ло В.Х. |
|  |  |  |  |
| Преподаватель | « » 2022 г. | (подпись) | Фандеев И.И. |

Москва, 2022г.

**СОДЕРЖАНИЕ**

[ОТЧЕТ ВЫПОЛНЕНИЯ ПРАКТИЧЕСКОЙ РАБОТЫ. 4](#_Toc100445063)

[1. Управление воспроизведением звука 4](#_Toc100445064)

[1.1 Выбрать тип аудио-потока 4](#_Toc100445065)

[1.2 Использовать аппаратные клавиши для управления громкостью 4](#_Toc100445066)

[1.3 Использовать аппаратные кнопки управления воспроизведением 5](#_Toc100445067)

[1.4 Реализовать запрос аудио-фокуса 6](#_Toc100445068)

[1.5 Реализовать обработку потери аудио-фокуса 6](#_Toc100445069)

[1.6 Реализовать использование Duck 7](#_Toc100445070)

[2 Работа с аппаратной частью вывода звука 8](#_Toc100445071)

[2.1 Реализовать проверку, какие аппаратные средства используются. 8](#_Toc100445072)

[2.2 Реализовать обработку изменения устройства вывода 8](#_Toc100445073)

[3. Работа с камерой 8](#_Toc100445074)

[3.1 Реализовать получение разрешения на камеру 9](#_Toc100445075)

[3.2 Реализовать получение фото с помощью других приложений 9](#_Toc100445076)

[3.3 Реализовать получение миниатюр 9](#_Toc100445077)

[3.4 Реализовать сохранение полноразмерной фотографии 10](#_Toc100445078)

[3.5 Реализовать добавление фотографии в галерею 11](#_Toc100445079)

[3.6 Реализовать масштабирование фотографий 11](#_Toc100445080)

[4. Запись видео 12](#_Toc100445081)

[4.1 Получить разрешение на камеру 12](#_Toc100445082)

[4.2 Реализовать запись видео с помощью других приложений 12](#_Toc100445083)

[4.3 Реализовать просмотр видео 12](#_Toc100445084)

[5. Управление камерой(прямая работа) 12](#_Toc100445085)

[5.1 Создать объект камеры 12](#_Toc100445086)

[5.2 Создать предпросмотр с камеры. Реализовать выбор и запуск предпросмотра. 13](#_Toc100445087)

[5.3 Реализовать изменение настройки камеры 14](#_Toc100445088)

[5.4 Реализовать выбор ориентации предпросмотра 15](#_Toc100445089)

[5.5 Реализовать получение картинки с камеры 15](#_Toc100445090)

[5.6 Реализовать перезапуск предпросмотра 15](#_Toc100445091)

[5.7 Реализовать остановку предпросмотра и освобождение камеры 16](#_Toc100445092)

[ВЫВОД 17](#_Toc100445093)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 17](#_Toc100445094)

# ОТЧЕТ ВЫПОЛНЕНИЯ ПРАКТИЧЕСКОЙ РАБОТЫ.

**1.** Управление воспроизведением звука

Если ваши приложения воспроизводят звук, важно сделать управлением привычным для пользователя. Также очень важно управлять аудио фокусом, чтобы одновременно несколько приложений не воспроизводили звук.

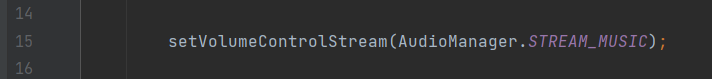
После изучения материала, вы сможете создавать приложения, которые отзываются на аппаратные клавиши управления звуком, используют аудио фокус при воспроизведении и адекватно реагируют на изменение фокуса другими приложениями или системой.

**1.1** Выбрать тип аудио-потока

Первым делом необходимо определиться с тем, какой тип аудио-потока будет использовать ваше приложение.Android поддерживает разные типы аудио-потоков для проигрывания музыки, сигнализации, уведомлений, мелодии входящих звонков, системных звуков, звука при разговоре и DTMF сигналов. Это сделано для того, чтобы обеспечить пользователю возможность независимо контролировать громкость для каждого потока. Большинство потоков ограничены системными событиями и, так что если ваше приложение не является заменой будильника, вы почти наверняка будете использовать для воспроизведения звуков поток STREAM\_MUSIC.

**1.2** Использовать аппаратные клавиши для управления громкостью

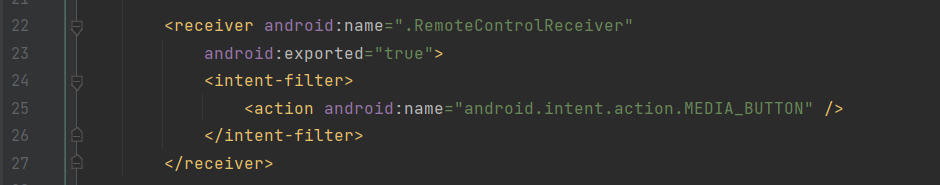
По умолчанию, клавиши управления громкостью влияют на громкость активного звукового потока. Если приложение в текущий момент не воспроизводит звук, эти клавиши будут управлять громкостью мелодии звонка. Если это игра или аудиоплеер, пользователи ждут, что клавиши управляют громкостью музыки или звуков в игре. Сейчас вы, возможно, подумали о том, чтобы отлавливать число нажатий на клавиши громкости и таким образом вычислять нужную громкость. Забудьте об этом! Android предоставляет метод setVolumeControlStream(), который напрямую управляет аудио-потоком при нажатии клавиш.



**Рисунок 2 – метод setVolumeControlStream()**

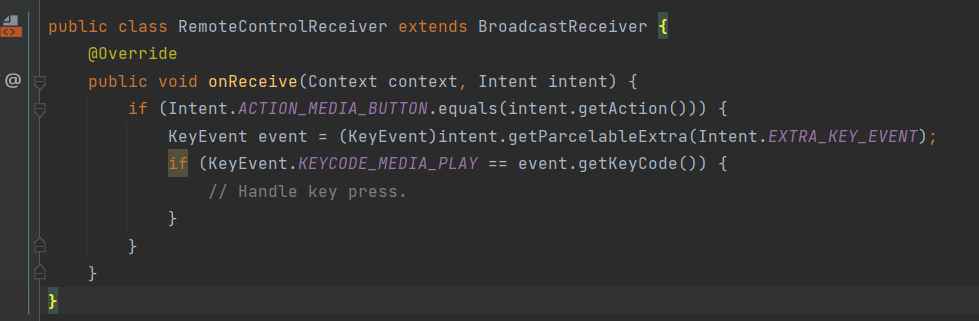
**1.3 Использовать аппаратные кнопки управления воспроизведением**

Такие кнопки, как воспроизведение, стоп, перемотка и пауза доступны на некоторых смартфонах и на большинстве гарнитур. При нажатии таких кнопок, система рассылает широковещательное намерение с действием ACTION\_MEDIA\_BUTTON.



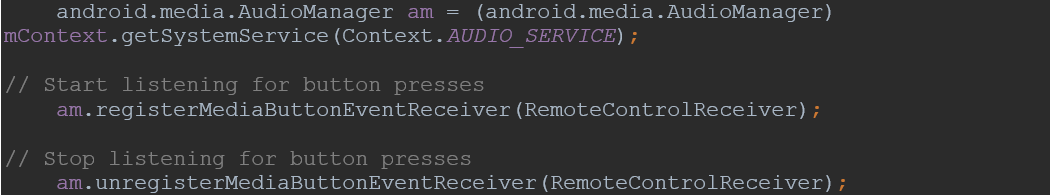
**Рисунок 3 – объект типа BroadcastReceiver в файле манифеста**

Зарегистрированный приемник должен сам извлечь нажатую клавишу.



**Рисунок 4 – получить нажатую клавишу и соответствующим образом повлиять на аудио-поток**

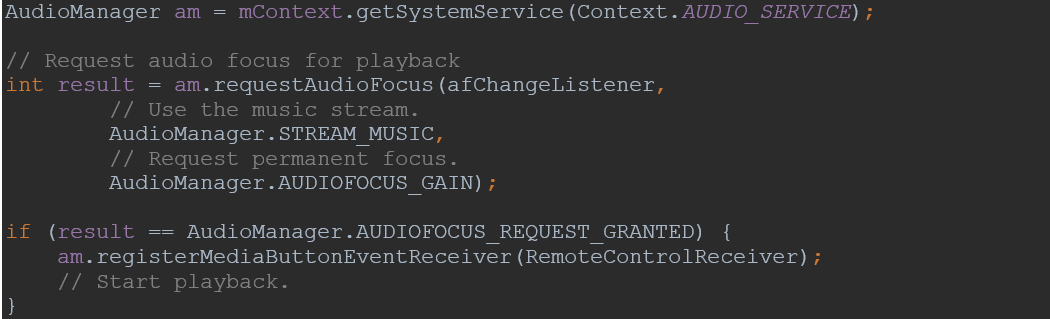
Поскольку обработчик нажатия на медиа-кнопки могут содержать несколько приложений, вы также должны программно контролировать, когда ваше приложение должно обрабатывать такие события.



**Рисунок 5 – Включать и отключать обработку**

**1.4 Реализовать запрос аудио-фокуса**

Итак, приложение должно захватить аудио-фокус нужного аудио-потока перед началом воспроизведения любых звуков. Это можно сделать с помощью метода requestAudioFocus(), который возвращает AUDIOFOCUS\_REQUEST\_GRANTED в случае успеха.



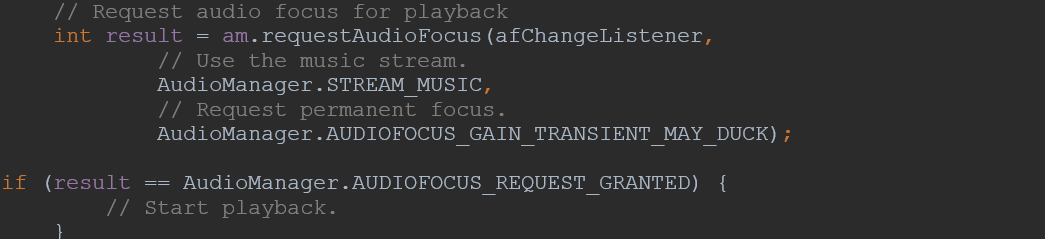
**Рисунок 6 – запросить постоянный фокус для музыкального потока**

В случае отказа от временного фокуса, любое прерванное приложение может продолжить воспроизведение.



**Рисунок 7 – Любое прерванное приложение может продолжить воспроизведение**

Для запроса временного фокуса есть дополнительный параметр: хотите ли вы использовать “ducking”. “Правильные” приложения обычно немедленно заглушают воспроизведение при потере фокуса.



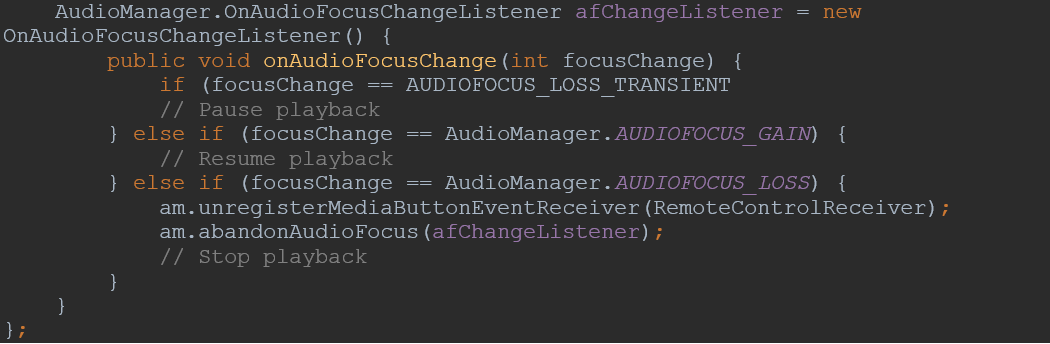
**Рисунок 8 – Немного убавив громкость**

**1.5 Реализовать обработку потери аудио-фокуса**

Если ваше приложение может запросить фокус, оно может его также потерять при запросе другого приложения. Как приложение будет обрабатывать потерю фокуса, зависит от того, как он был потерян.

Метод обратного вызова onAudioFocusChange() вызывается при изменении слушателя, которого вы зарегистрировали при запросе фокуса и принимает в качестве параметра тип события. Возможные события являются близнецами для типов фокуса, которые мы уже рассмотрели – постоянная потеря, временная потеря и временная потеря с разрешенным ducking.

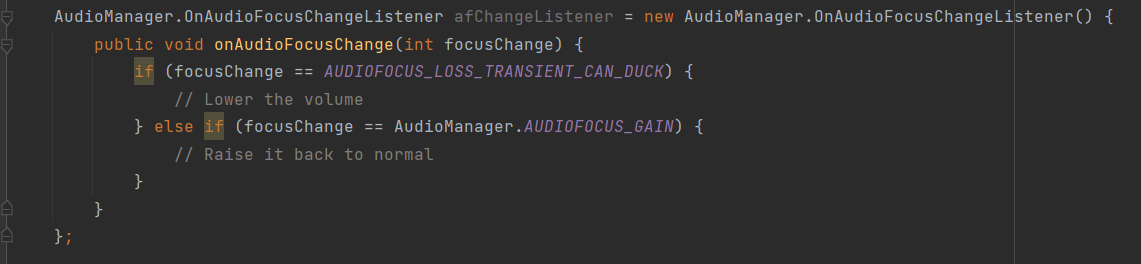
Если потеря фокуса является постоянной, значит другое приложение сейчас используется для прослушивания аудио и ваше приложение должно эффективно себя завершить. Эффективно завершить, значит остановить воспроизведение, выключить обработчики нажатия на медиа-кнопки и отказаться от фокуса. Таким образом, чтобы возобновить воспроизведение в вашем приложении, пользователю будет необходимо явно указать вашему приложению это сделать.



**Рисунок 9 – приостановить воспроизведение при временной потере фокуса и заново его возобновить как только фокус вернулся.**

**1.6 Реализовать использование Duck**

Ducking это процесс уменьшения громкости постоянного аудио-потока (вместо полной остановки воспроизведения), чтобы сделать временный звук лучше слышимым для пользователя.

****

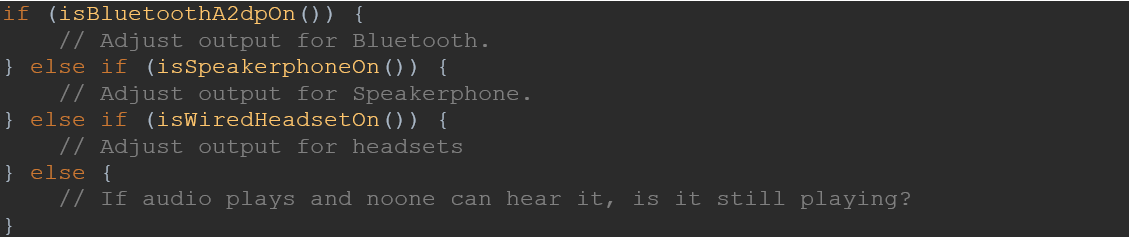
**Рисунок 10 – Реализовать использование Duck**

**2** Работа с аппаратной частью вывода звука

2.1 Реализовать проверку, какие аппаратные средства используются.

Работа вашего приложения может зависеть от доступных аппаратных средств.

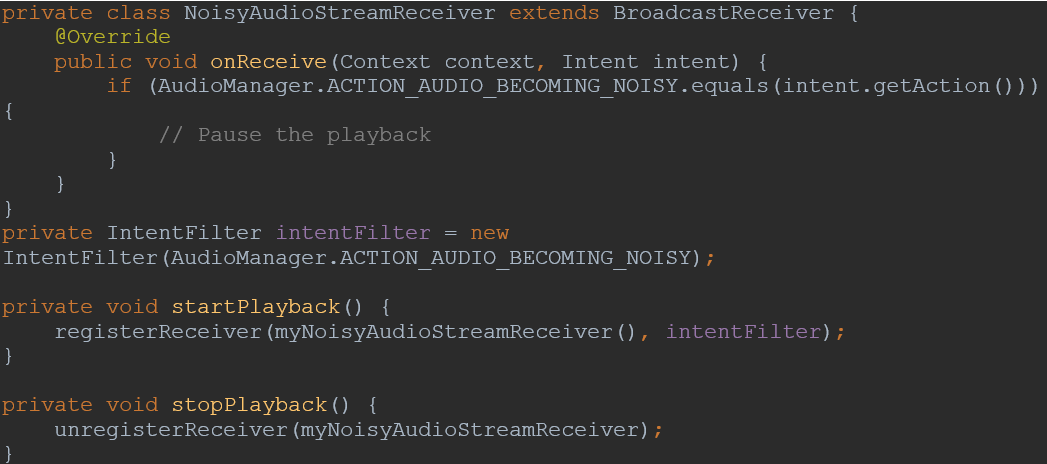
Вы можете использовать класс AudioManager, чтобы узнать куда в текущий момент выводится аудио: на динамик, проводную гарнитуру или на bluetooth устройство.



**Рисунок 11 – Проверяем, какие аппаратные средства используются**

2.2 Реализовать обработку изменения устройства вывода

Если наушники были отсоединены или bluetooth устройство было отключено, аудио поток автоматически переходит в динамик. Если вы слушали музыку на большой громкости, это может стать сюрпризом.



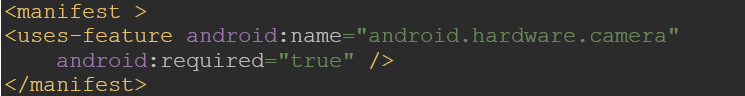
**Рисунок 12 –** Реализовать обработку изменения устройства вывода

3. Работа с камерой

Android позволяет передавать большие файлы между устройствами, используя технологию Android Beam. Технология имеет простой интерфейс и позволяет пользователям начать передачу просто соприкоснувшись устройствами. Android Beam автоматически передает файл с одного устройства на другое и информирует пользователя после окончания процесса.

3.1 Реализовать получение разрешения на камеру

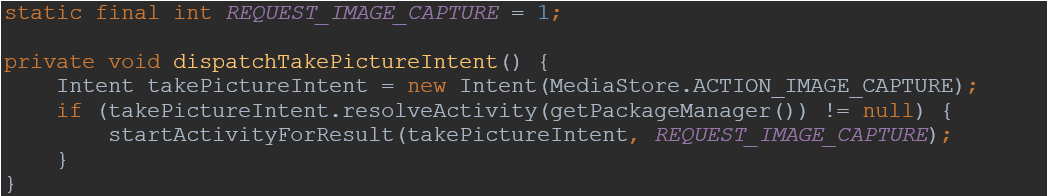
Если получение снимков – важная часть вашего приложения, то показывайте его на Google Play только для устройств, которые имеют камеру. Для этого добавьте элемент <uses-features> в файл манифеста:



**Рисунок 13 –** Реализовать получение разрешения на камеру

3.2 Реализовать получение фото с помощью других приложений

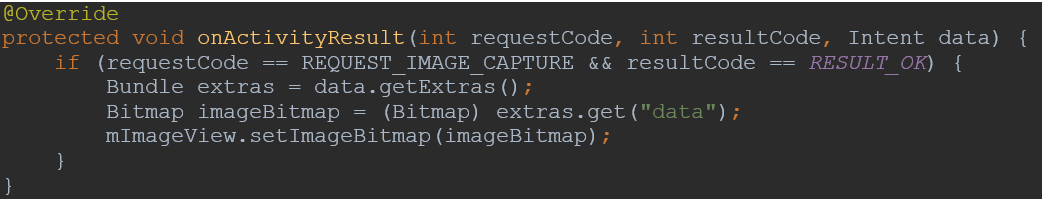
Чтобы использовать другие приложение для каких-либо действий, в Android применяются намерения – объекты типа Intent, в которых описывается, что конкретно вы хотите сделать. Этот процесс включает три составляющих: само намерение – объект типа Intent, вызов внешнего явления – объекта типа Activity и некоторый код, который обрабатывает данные, полученные от внешнего явления.



**Рисунок 14 – Получение фотографии с камеры**

**3.3 Реализовать получение миниатюр**

Наверняка сам факт срабатывания затвора камеры это не предел желаний, и после съемки вы захотите получить картинку с камеры в ваше приложение, чтобы что-то с ней сделать.



**Рисунок 15 – получить изображение и отобразить его в компоненте ImageView**

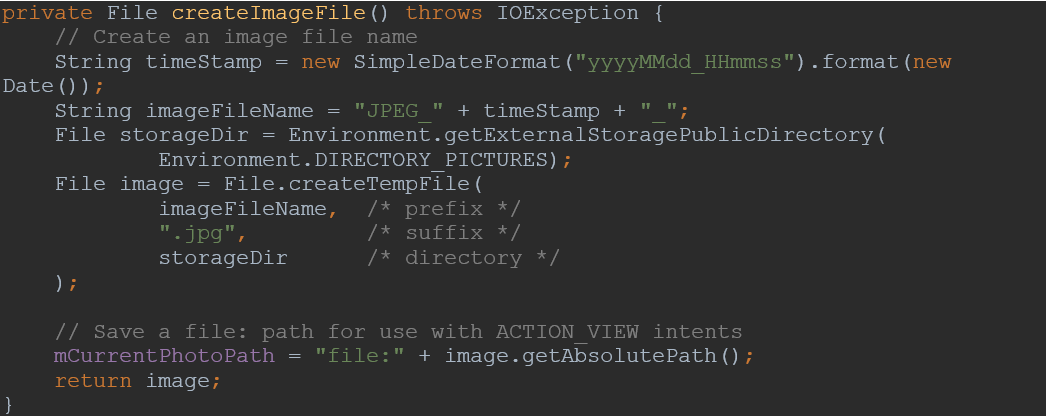
3.4 Реализовать сохранение полноразмерной фотографии

Приложение Камера сохраняет полноразмерное изображение, если передать ему имя файла для сохранения. Вы должны передать имя со всеми спецификаторами, чтобы приложение могло сохранить фотографию.

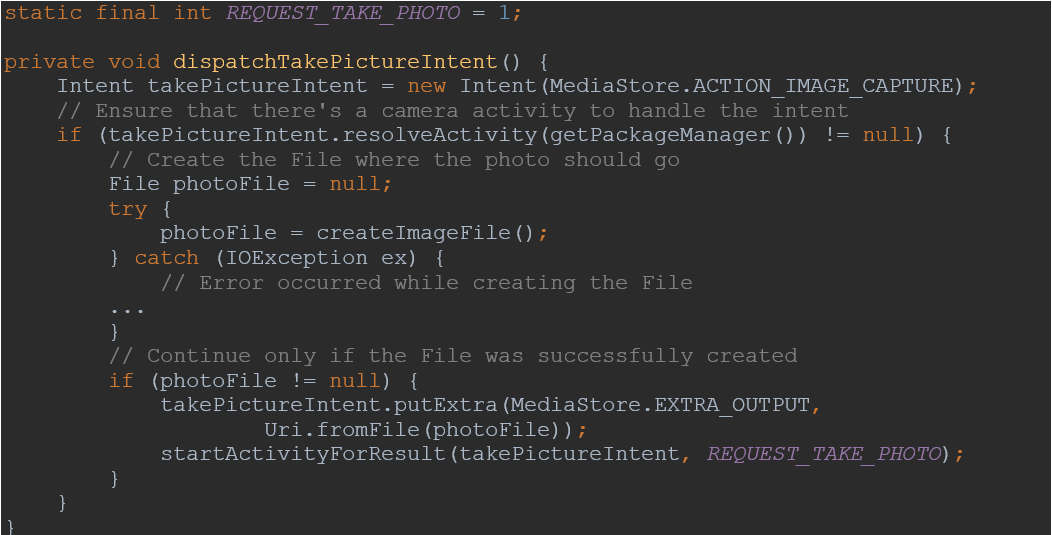


**Рисунок 16 – записывать в публичный каталог и читать**

После выбора директории для файлов, необходимо выбрать устойчивое к коллизиям имя файла. Вы можете также сохранить директорию в переменной, чтобы использовать ее позже. Пример кода, который возвращает уникальное имя файла для новой фотографии, используя текущие дату и время:



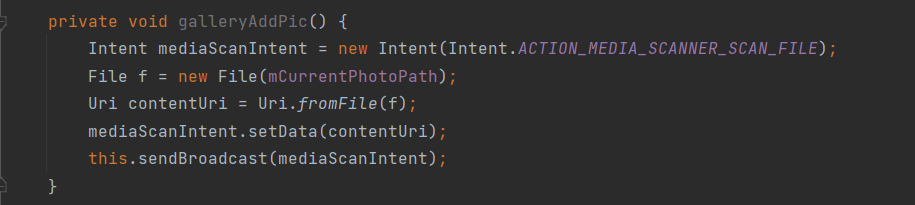
**Рисунок 17 – Код, который возвращает уникальное имя файла для новой фотографии**



**Рисунок 18 – метод для получения имени файла**

**3.5 Реализовать добавление фотографии в галерею**

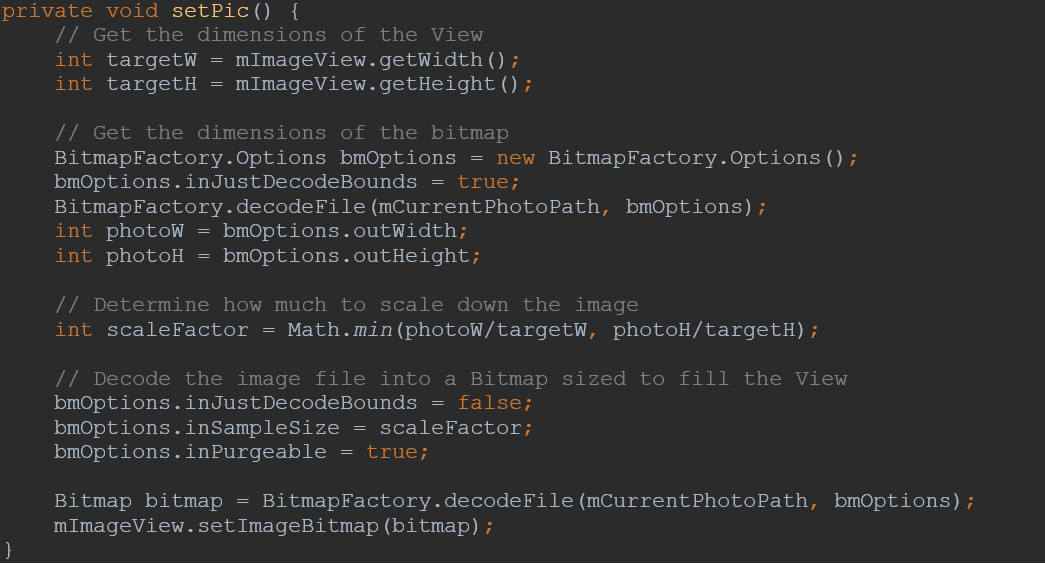
После создания фотографии с помощью намерения, вы знаете, где она находится, поскольку сообщили куда ее сохранить. Для всех остальных случаев, простейший способ получить доступ к созданной фотографии – использовать системный медиа-провайдер.



**Рисунок 19 – Использовать системный медиа-сканер**

**3.6** Реализовать масштабирование фотографий

Управление множеством полноразмерных фотографий требует хитрого подхода при ограниченной памяти. Если вы заметили, что приложение тратит всю доступную память уже после отображения нескольких фотографий, вы можете значительно уменьшить размер динамической памяти за счет расширения JPEG в массив памяти, который масштабируется в соответствии с размером компонента, в котором отображается картинка.

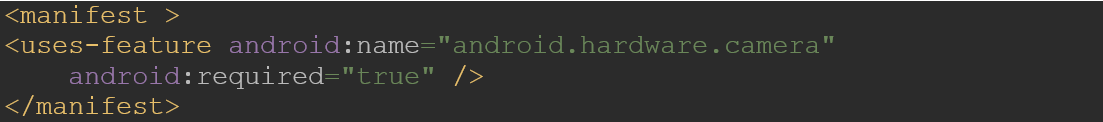


**Рисунок 20 –** Реализовать масштабирование фотографий

4. Запись видео

4.1 Получить разрешение на камеру

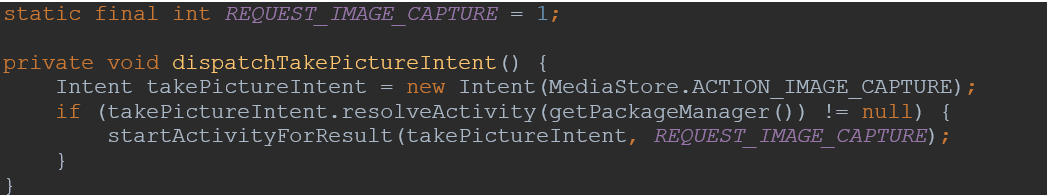
Чтобы указать, что вашему приложению требуется камера, добавьте тег <uses-feature> в файл манифеста:



**Рисунок 21 –** Получить разрешение на камеру

4.2 Реализовать запись видео с помощью других приложений

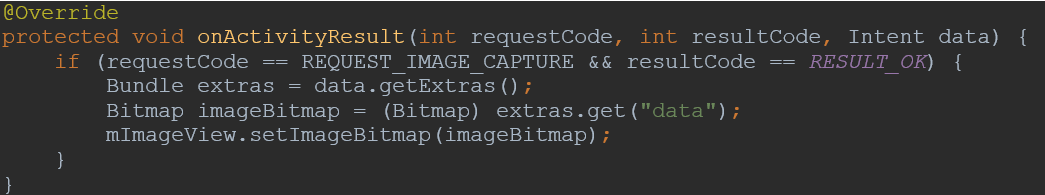
Android просит другие приложения что-либо сделать с помощью намерений. Процесс включает в себя три составляющие: само намерение, запуск внешнего явления и выполнение некоторого кода для обработки полученного видео.



**Рисунок 22 –** Реализовать запись видео с помощью других приложений

4.3 Реализовать просмотр видео

Приложение Камера возвращает ссылку на видео в виде Uri, с помощью намерения, которое можно получить в методе onActivityResult(). Следующий код служит для получения видео и отображения его в компоненте VideoView:

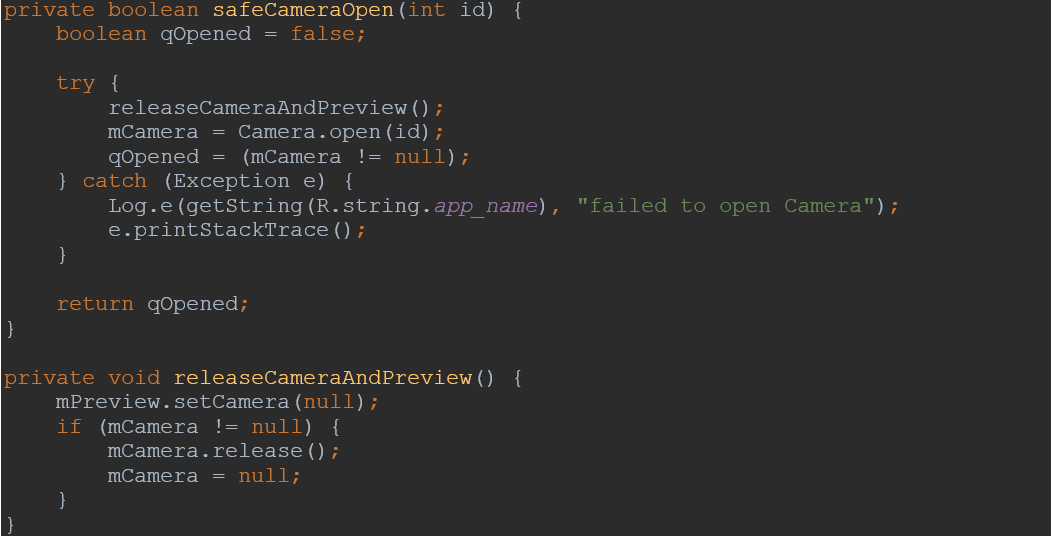


**Рисунок 23 – Код служит для получения видео и отображения его в компоненте VideoView**

**5. Управление камерой(прямая работа)**

**5.1 Создать объект камеры**

Первым делом нужно получить экземпляр класса Camera. Рекомендуем работать с камерой в отдельном потоке (его можно создать в методе onCreate()) , как это делает стандартное приложение Камера. Это хорошая идея, поскольку запуск камеры может утянуть на дно весь поток с пользовательским интерфейсом. В отдельных случаях можно запускать камеру в методе onResume(), чтобы упростить повторное использование кода и сделать управление проще.

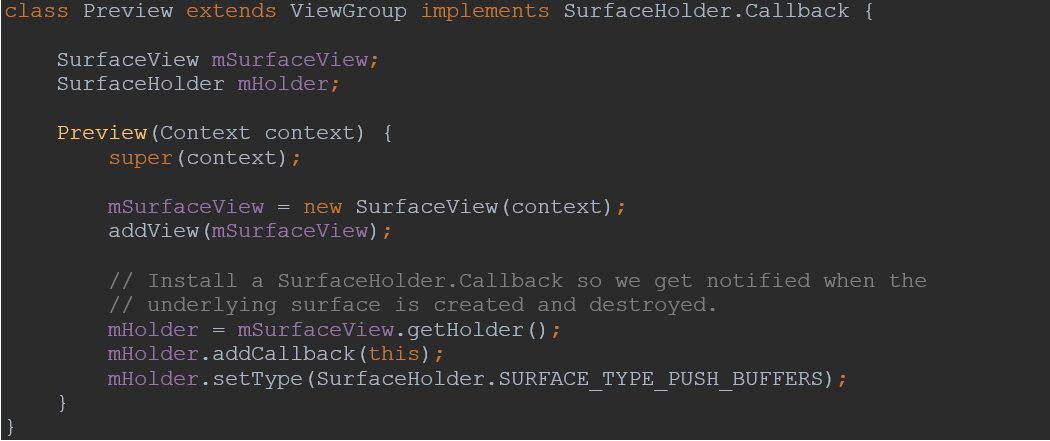


**Рисунок 24 – Создать объект камеры**

**5.2 Создать предпросмотр с камеры. Реализовать выбор и запуск предпросмотра.**

Создаем класс предпросмотра

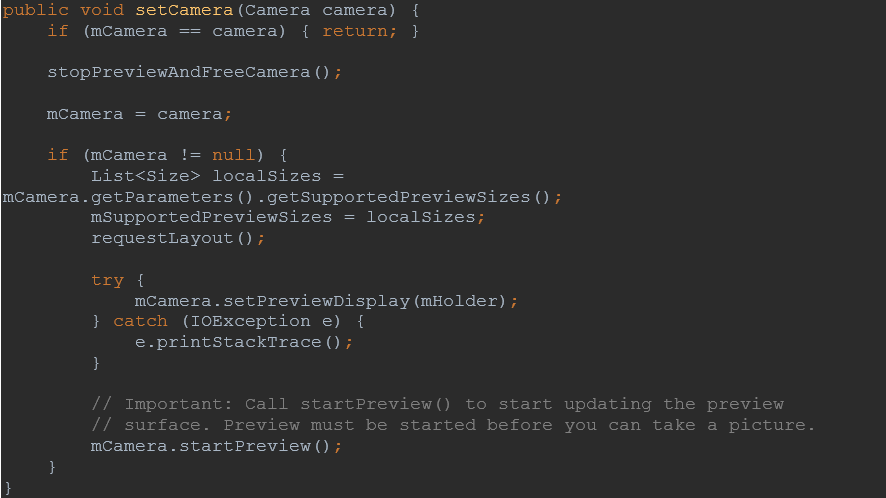
Чтобы запустить предпросмотр, сначала надо создать класс для предпросмотра. Предпросмотр требует реализации интерфейса android.view.SurfaceHolder.Callback, который используется для передачи картинки с датчика камеры в приложение:



**Рисунок 25 – Создаем класс предпросмотра**

**Выбор и запуск предпросмотра**

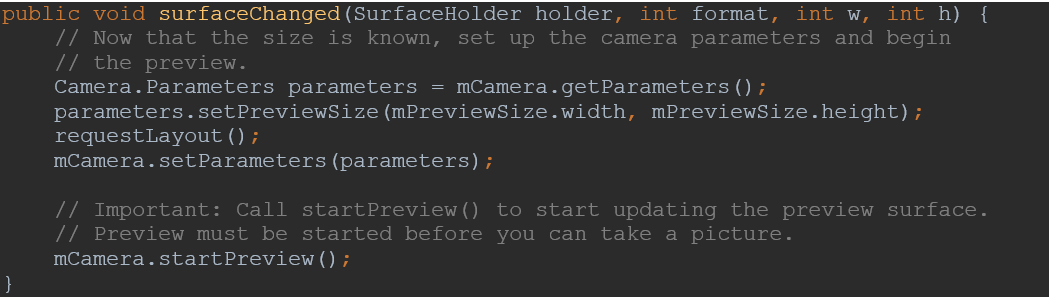
Экземпляр камеры и связанный с ним экземпляр предпросмотра должны создаваться в определенном порядке – объект камеры должен создаваться первым. В приведенном ниже примере, процесс инициализации камеры сделан таким образом, что метод Camera.startPreview() вызывается в методе setCamera() всякий раз, когда пользователь меняет камеру. Объект предпросмотра должен также перезапускаться в методе обратного вызова surfaceChanged() класса предпросмотра.



**Рисунок 26 – Выбор и запуск предпросмотра**

**5.3 Реализовать изменение настройки камеры**

Камера позволяет менять свои настройки, начиная зумом и заканчивая компенсацией экспозиметра. В данном примере показано только изменение размера предпросмотра. Чтобы узнать больше, смотрите исходники приложения Камера.



**Рисунок 27 – Изменение настроек камеры**

**5.4 Реализовать выбор ориентации предпросмотра**

Большинство приложений блокируют ориентацию дисплея в альбомном режиме, поскольку это естественная ориентация сенсора камеры. Этот параметр не мешает вам делать снимки в портретном режиме, поскольку ориентация сенсора записывается в заголовки EXIF. Метод setCameraDisplayOrientation() позволяет изменить ориентацию предпросмотра, не затрагивая информацию, которая будет записана.

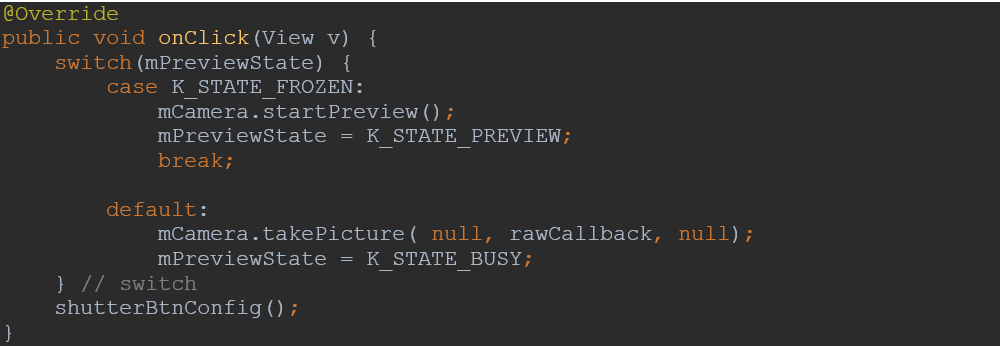
**5.5 Реализовать получение картинки с камеры**

Используйте метод Camera.takePicture(), чтобы получить изображение с камеры, после того, как предпросмотр был запущен. Вы можете создать объекты Camera.PictureCallback и Camera.ShutterCallback и передать их в метод Camera.takePicture().

Если вы хотите захватывать изображения непрерывно, вы можете создать объект Camera.PreviewCallback, который содержит метод onPreviewFrame(). Вы можете захватить кадры только из выбранного объекта предпросмотра, или настроить задержку для вызова takePicture().

**5.6 Реализовать перезапуск предпросмотра**

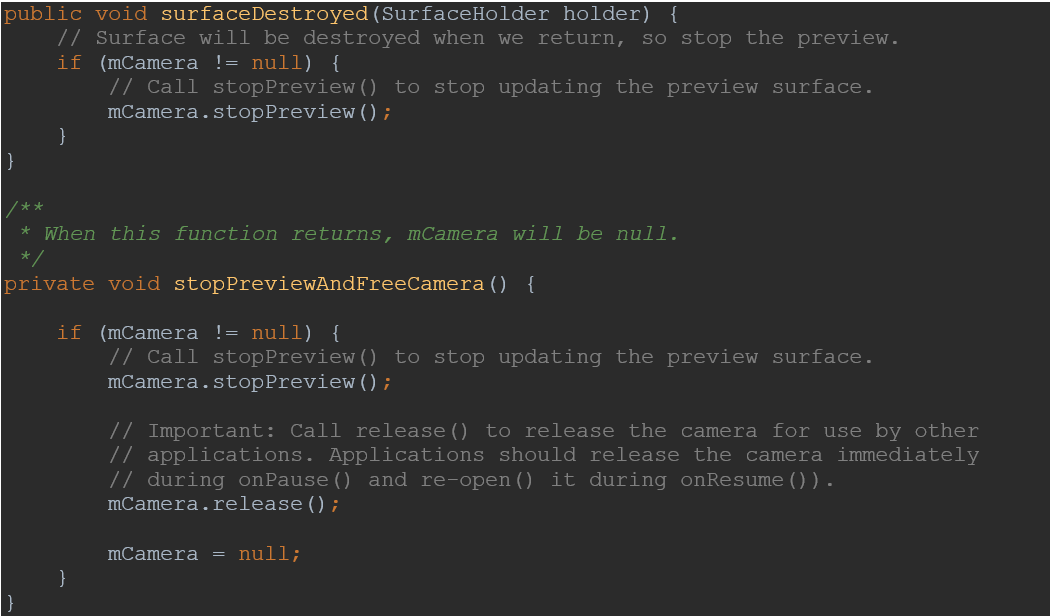
После спуска затвора, вы должны перезапустить предпросмотр, прежде чем пользователь сможет сделать еще одну фотографию.



**Рисунок 28 – Перезапуск сделан с помощью переопределения метода нажатия на кнопку затвора**

**5.7 Реализовать остановку предпросмотра и освобождение камеры**

Вы должны освободить объект Camera, как только камера становится не нужна. Иначе вы рискуете получить аварийное завершение других приложений, или даже новых экземпляров вашего приложения.



**Рисунок 29 – Остановка предварительного просмотра и освобождение камеры**

Ранее в этом уроке, эта процедура была также частью метода setCamera(), поскольку инициализация камеры всегда начинается с остановки предварительного просмотра.

# ВЫВОД

В ходе выполнения практической работы были изучены:

* Методические указания Работа с мультимедиа

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Лекции по дисциплине «Разработка мобильных приложений» / И. В. Синицын, МИРЭА — Российский технологический университет, 2022.