|  |
| --- |
| 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 №11**

**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. 3](#_Toc100450779)

[1. Печать документов 3](#_Toc100450780)

[1.1 Реализовать печать изображений 3](#_Toc100450781)

[2 Печать HTML документов 4](#_Toc100450782)

[2.1 Реализовать загрузку HTML документа 4](#_Toc100450783)

[2.2 Реализовать создание задания на печать 5](#_Toc100450784)

[3. Печать собственных документов 6](#_Toc100450785)

[3.1 Реализовать присоединение к менеджеру печати 6](#_Toc100450786)

[3.2 Реализовать создание адаптера печати 7](#_Toc100450787)

[3.3 Реализовать вычисление информации о документе печати 8](#_Toc100450788)

[3.4 Реализовать запись в файл печати 9](#_Toc100450789)

[3.5 Реализовать отрисовку PDF страницы 10](#_Toc100450790)

[ВЫВОД 12](#_Toc100450791)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 12](#_Toc100450792)

# ОТЧЕТ ВЫПОЛНЕНИЯ ПРАКТИЧЕСКОЙ РАБОТЫ.

**1.** Печать документов

Пользователи Android зачастую просматривают информацию исключительно на их устройствах, но бывают случаи, когда показывать экран не является адекватным способом для обмена информации. Возможность распечатать информацию Android устройств, является отличным способом показать большую версию данных из вашего приложения или поделиться ей с теми, кто не использует ваше приложение. Распечатка также позволяет создавать отпечатки информации, которые не зависят от наличия устройства, заряда батареи или наличия беспроводного соединения.

Печать фотографий

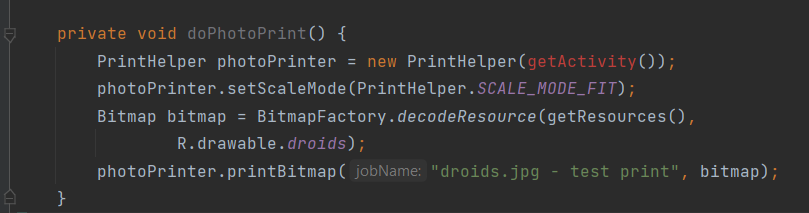
Фотосъемка и обмен фотографиями, это наиболее популярное использование мобильных устройств. Если ваше приложение позволяет сделать фотографию, показать ее на экране или поделиться ей с другими пользователями, рассмотрите возможность разрешить распечатывать фотографии из вашего приложения. Библиотека поддержки Android предоставляет удобные функции для поддержки печати, используя минимум кода и простой набор вариантов компоновки печати.

**1.1** Реализовать печать изображений

Класс PrintHelper библиотеки поддержки предоставляет простой способ печати изображений. Класс имеет единственную опцию макета, setScaleMode(), которая позволяет выводить изображения на печать в одном из двух режимов:

SCALE\_MODE\_FIT – опция меняет размер изображения так, чтобы все изображение находилось в пределах области печати страницы.

SCALE\_MODE\_FILL – опция масштабирует изображение таким образом, чтобы оно заполнило всю печатную площадь страницы. Другими словами, у изображения при необходимости обрезается одна или несколько сторон. Эта опция используется по умолчанию, если вы не выбрали режим печати.



**Рисунок 1 – Запустить процесс печати**

Никаких действий, после вызова метода printBitmap() не требуется. Далее будет выведен стандартный интерфейс Android для выбора принтера и опций печати. После выбора принтера на него будет отправлено задание на печать, а в системной панели появится уведомление о процессе печати.

**2** Печать HTML документов

Печать содержимого, выходящего за рамки простой фотографии, требует объединить текст и графику в один документ для печати. Android фреймворк предоставляет способ использования HTML для сборки таких документов и вывода их на печать, используя минимум кода.

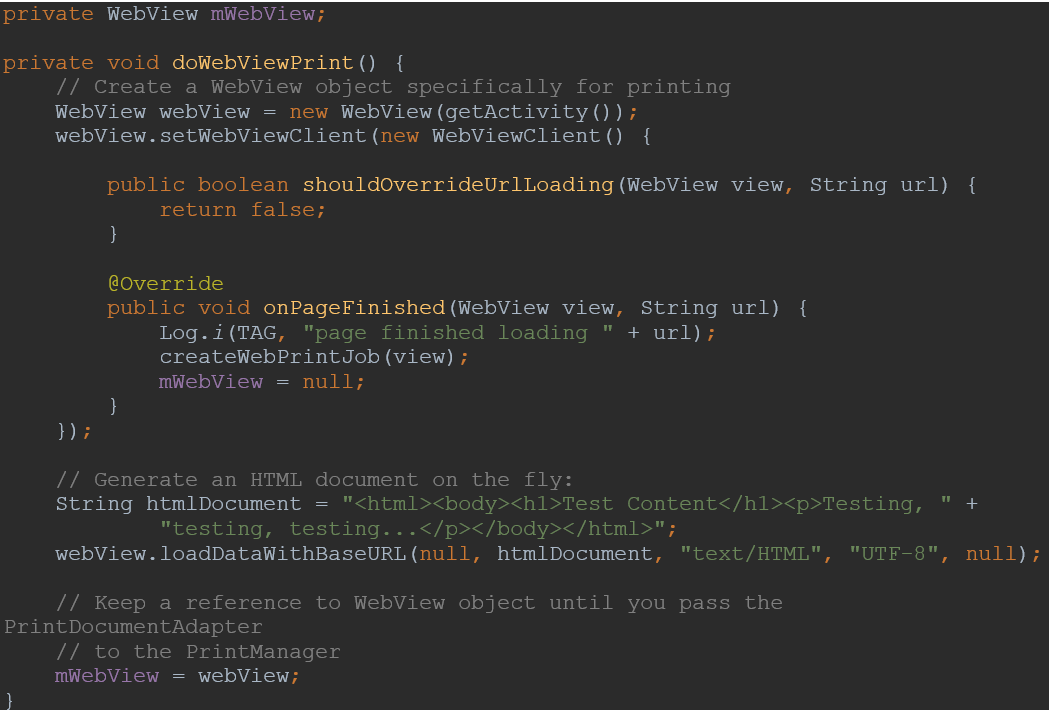
2.1 Реализовать загрузку HTML документа

Печать HTML документа с помощью WebView предполагает загрузку HTML ресурса или создание HTML документа как строки. В этом разделе описывается как создать HTML строку и загрузить ее в WebView для печати.

Данный объект обычно является частью разметки явления. Однако, если ваше приложение не использует WebView, вы должны создать экземпляр класса специально для печати. Главные шаги создания компонента печати:

1. Создание объекта типа WebViewClient, который запустит печать после загрузки HTML ресурса.

2. Загрузка HTML ресурса в объект WebView



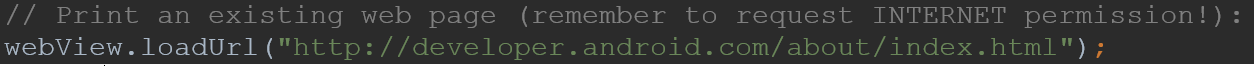
**Рисунок 2 – создать экземпляр WebViewClient и загрузить HTML документ**

Если вы хотите добавить графику на страницу, поместите графические файлыв директорию assets/ вашего проекта и укажите базовый URL в качестве первого параметра метода loadDataWithBaseURL(), как показано ниже:



**Рисунок 3 –** добавить графику на страницу

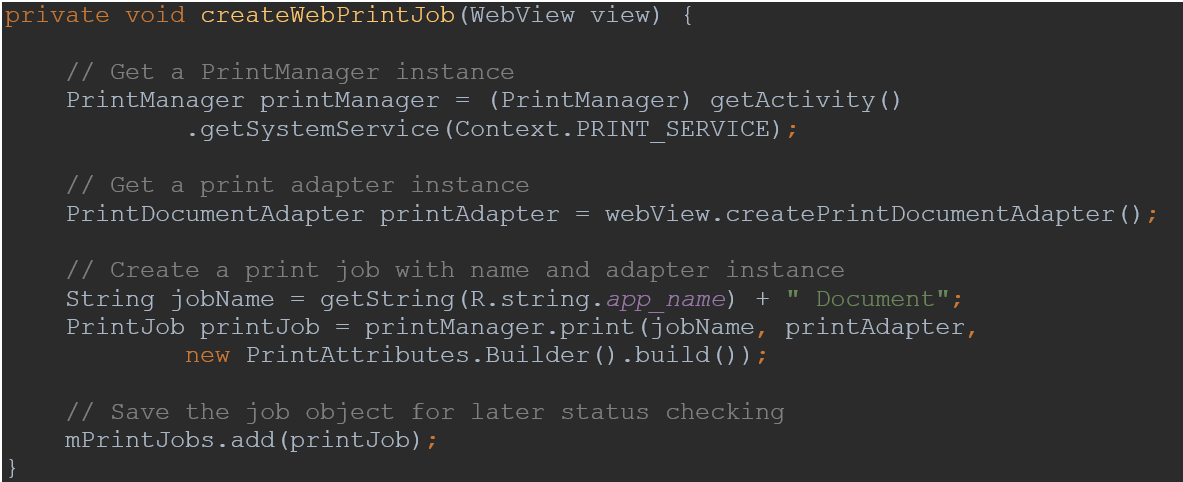
Вы можете также загрузить web-страницу для печати, заменив метод loadDataWithBaseURL() на метод loadUrl(), как показано ниже:



**Рисунок 4 – Загрузить web-страницу для печати**

2.2 Реализовать создание задания на печать

После создания объекта WebView и загрузки HTML документа, ваше приложение почти достигло процесса печати. Осталось получит доступ кменеджеру печати PrintManager, создать адаптер печати и, наконец, создать задание на печать.



**Рисунок 5 –** Создание задания на печать

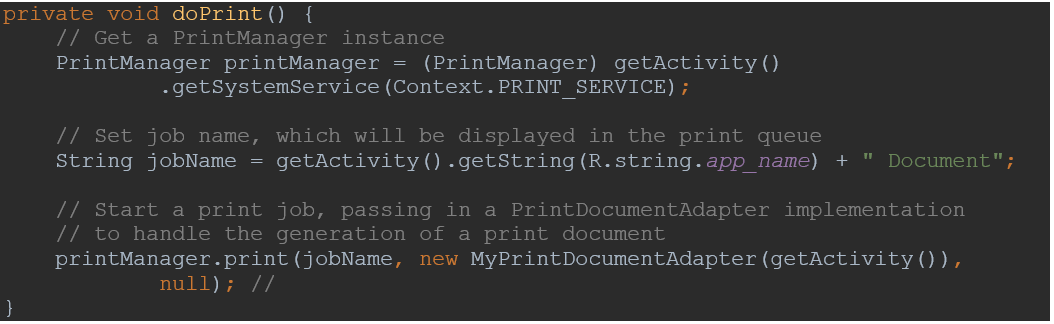
3. Печать собственных документов

Для некоторых приложений, например графических редакторов, редакторов макетов и других приложений, в которых функции сосредоточены на работе с графикой, создание красивой распечатки – это ключевая функция. В данном случае недостаточно печати изображений или HTML документов. Для вывода на печать в таких приложениях требуется прецизионный контроль во всем, что попадает на страницу печати, включая шрифты, текстовая разметка, разрывы страниц, заголовки, колонтитулы и графические элементы.

Такой персональных подход к печати требует гораздо больше кода, чем мы обсуждали ранее. Вы должны создать компоненты, которые взаимодействуют с интерфейсом печати, приспособиться к настройкампринтера, нарисовать элементы страниц и управлять печать для множества страниц.

3.1 Реализовать присоединение к менеджеру печати

Если ваше приложение управляет процессом печати, первым делом после получения запроса на печать от пользователя, неоходимо присоединиться к интерфейсу печати Android и получить экземпляр класса PrintManager. Этот класс позволяет инициализировать задание на печать и запускает жизненный цикл печати. Следующий пример показывает как получить менеджер печати и запустить процесс печати:



**Рисунок 6 – получить менеджер печати и запустить процесс печати**

3.2 Реализовать создание адаптера печати

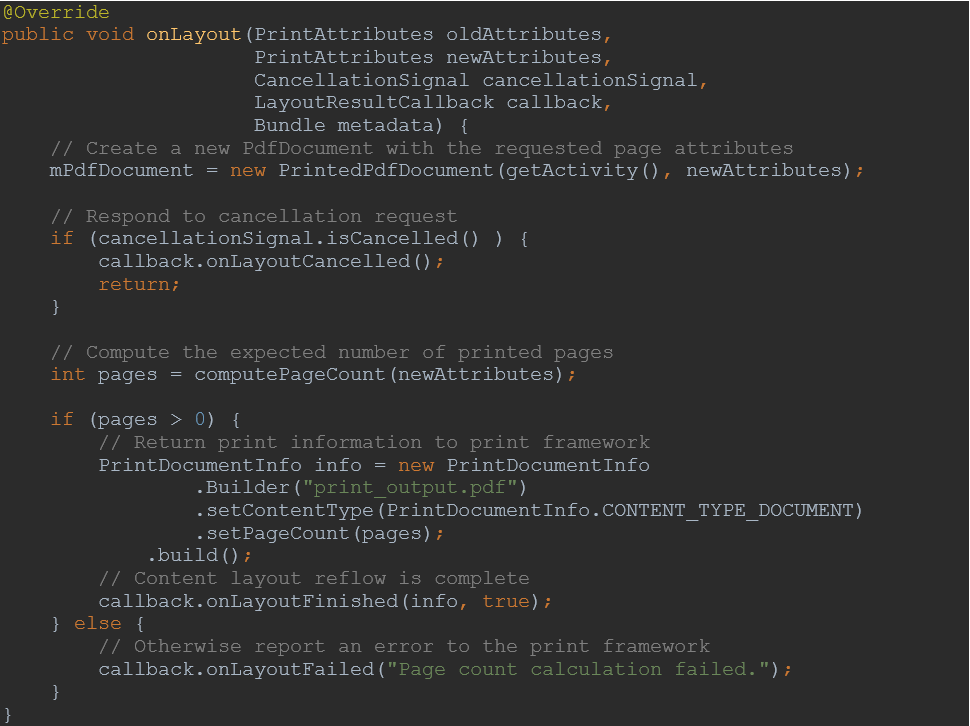
Адаптер печати взаимодействует с интерфейсом печати Android и выполняет пошагово процесс печати. Этот процесс требует от пользователя выбора принтера и опций печати, прежде чем документ отправится на печать. Этот выбор может повлиять на конечный результат, так как пользователь может выбирать принтеры с разными характеристиками вывода, например с разным размером страниц или разной ориентацией страниц. После выбора принтера, интерфейс печати запрашивает у адаптера формирование документа и подготовку к печати. После нажатия кнопки “печать”, интерфейс печати берет сформированный документ и передает его поставщику печати на вывод. Во время процесса печати пользователь может отменить печать, поэтому адаптер печати должен прослушивать статус и соответствующим образом на него реагировать.

Абстрактный класс PrintDocumentAdapter создан для управления жизненным циклом процесса печати, который имеет четыре метода обратного вызова. Вы должны реализовать эти методы в вашем адаптере печати, чтобы правильно взаимодействовать с интерфейсом печати:

* onStart() – вызывается как только начинается процесс печати. Если ваше приложение несколько подготовительных задач, например получение отпечатка данных, которые должны быть распечатаны, выполняйте их здесь. Реализация данного метода не обязательна.
* onLayout() – вызывается каждый раз, когда пользователь меняет настройки печати, которые влияют на вывод, например размер страницы, ориентация, заставляя приложение пересчитывать компоновку страниц для печати. Как минимум этот метод должен возвращать количество страниц, которое ожидается в документе.
* onWrite() – вызывается для передачи печатных страниц в файл для печати. Этот метод может вызываться один или несколько раз, после каждого вызова метода onLayout().
* onFinish() – вызывается однократно, в конце процесса печати. Если ваше приложение имеет несколько задач на освобождение ресурсов, выполните их здесь. Метод не является обязательным. В следующих разделах описано, как реализовать разметку и методы записи, которые критичны для функционирования адаптера печати.

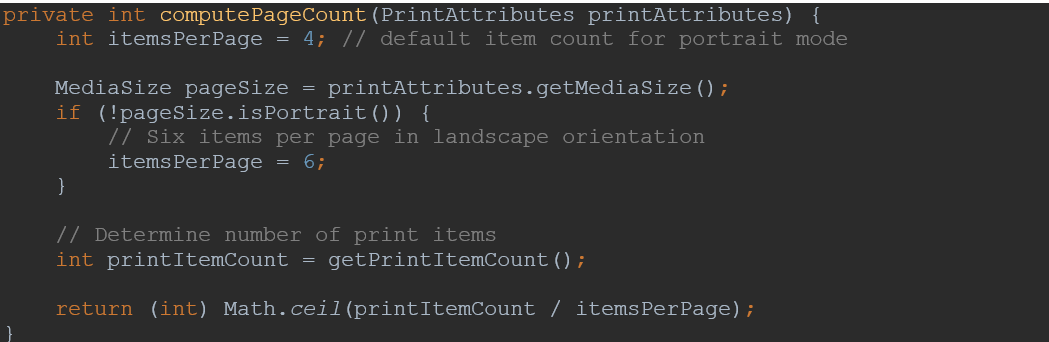
**3.3 Реализовать вычисление информации о документе печати**

Вместе с реализацией класса PrintDocumentAdapter, ваше приложение должно указать тип документа и подсчитать общее число страниц для печати, информацию о размере страниц. Реализация метода onLayout в адаптере производит эти вычисления и предоставляет информацию о том, что ожидается на выходе в класс PrintDocumentInfo, включая количество страниц и тип данных. Следующий пример показывает базовую реализацию метода onLayout для объекта PrintDocumentAdapter:



**Рисунок 7 – базовая реализация метода onLayout для объекта PrintDocumentAdapter**

Следующий код показывает реализацию, где количество страниц определяется ориентацией печати:



**Рисунок 8 – количество страниц определяется ориентацией печати**

3.4 Реализовать запись в файл печати

Когда приходит время записывать данные в файл для печати, система вызывает метод onWrite() класса PrintDocumentAdapter. Параметры метода указывают, какие страницы должны быть записаны. Ваша реализация метода должна поместить каждую запрошенную страницу в многостраничный PDF документ. При завершении этого процесса вы вызываете метод onWriteFinished().

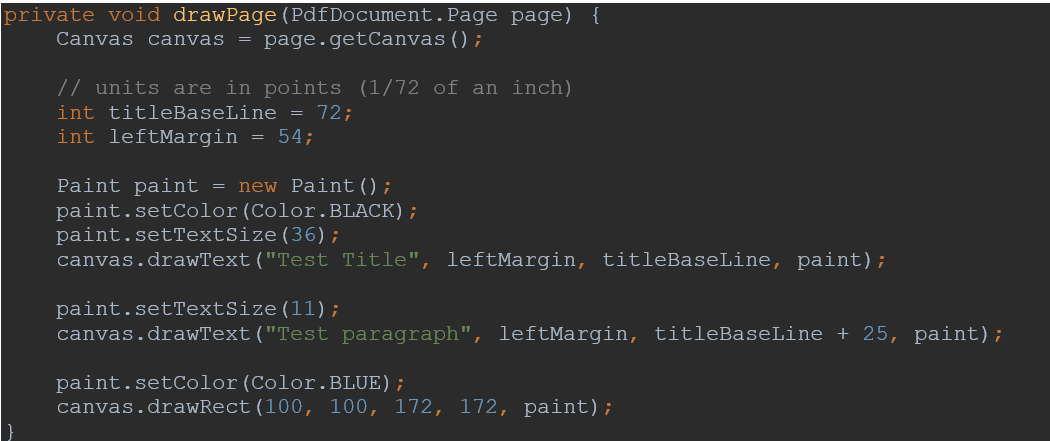
****

**Рисунок 8 – Класс PrintedPdfDocument для создания PDF файла**

**3.5 Реализовать отрисовку PDF страницы**

Ваше приложение должно генерировать PDF документ и передавать его в интерфейс печати Android. Вы можете использовать для этих целей библиотеку генерации PDF. В этом уроке мы расскажем, как использовать класс PrintedPdfDocument для генерации PDF из ваших данных.

Класс PrintedPdfDocument использует объект типа Canvas для отрисовки элементов PDF страниц, также как при рисовании на разметке явления. Вы можете нарисовать элементы на странице для печати используя методы класса Canvas.



**Рисунок 9 – нарисовать некоторые простые элементы на PDF странице**

# ВЫВОД

В ходе выполнения практической работы были изучены:

* Методические указания печать документов, печать HTML документов, печать собственных документов

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

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