МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ

КАЛИНИНГРАДСКОЙ ОБЛАСТИ

ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ

«КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СТРОИТЕЛЬСТВА»

(ГБУ КО ПОО «КИТиС»)

|  |  |  |
| --- | --- | --- |
| Курсовой проект  допущен к защите  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата)  Зам.директора по УМР  \_\_\_\_\_\_\_\_\_\_\_\_Павленко Г.Я.  (подпись) |  | Курсовой проект  защищен с оценкой  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка)  Руководитель работы  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |

Пояснительная записка к курсовому проекту

по дисциплине: МДК 01.01 Разработка программных модулей

Тема: Разработка мобильного Web-браузера под Android

Специальность 09.02.07 «Информационные системы и программирование»

|  |  |
| --- | --- |
| Выполнил: | студент 3 курса,  группы ИСп 19-2к  Макеев Дмитрий Константинович  \_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
| Руководитель: | Большакова-Стрекалова Анна Викторовна, преподаватель ГБУ КО ПОО КИТиС  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Калининград

2022

Содержание

[Введение 1](#_Toc102731586)

[1 Описание предметной области 3](#_Toc102731587)

[1.1 Аналоги разрабатываемого приложения 3](#_Toc102731588)

[1.2 Техническое задание 4](#_Toc102731589)

[1.3 Описание структуры приложения 5](#_Toc102731590)

[2 Описание Разработки Приложения 7](#_Toc102731591)

[2.1 Обоснование средств разработки 8](#_Toc102731592)

[2.2 Разработка интерфейса 11](#_Toc102731593)

[2.3 Разработка логики работы приложения 11](#_Toc102731594)

[Схема взаимодействия компонентов проекта 12](#_Toc102731595)

[2.4 Описание переменных, компонентов, классов и подпрограмм 13](#_Toc102731596)

[3 Тестирование и установка приложения 16](#_Toc102731597)

[Заключение 17](#_Toc102731598)

[Список используемой литературы 18](#_Toc102731599)

[Приложение А. Код программы 19](#_Toc102731600)

## Введение

Понятие мобильного веб-приложения можно трактовать по-разному. Для одних — это приложение, работающее в Web и спроектированное так, чтобы корректно отображаться на мобильном устройстве, а для других — это приложение, созданное специально для конкретной мобильной ОС, которое соединяется с Web для отправки и приема данных. Чтобы разграничить эти позиции, можно воспользоваться «шкалой» от стандартных до нативных веб-приложений. Если нативные работают со скоростью «железа», то гибридные и мобильные исполняются поверх дополнительных уровней, которые расходуют вычислительные ресурсы, снижая быстродействие устройства.

Введем понятия стандартных и быстрореагирующих приложений, а также «мобильного Web». Соответствующие технологии, а также их преимущества и недостатки весьма схожи, однако здесь перечислим их по отдельности, подчеркнув, что, в сущности, это разные решения. Некоторым пользователям и разработчикам нравится адаптивный интерфейс, тогда как другие отдают предпочтение специализированным приложениям для конкретных устройств. Мобильно-ориентированные веб-сайты (например, соответствующие варианты Facebook или Google Docs) обычно предлагают пользователю возможность доступа и к стандартной веб-версии. Стандартная программа на смартфоне, вероятнее всего, будет работать медленнее, но некоторые пользователи выберут именно ее, чтобы иметь доступ ко всем возможностям сайта.

Стандартные веб-приложения

Термином «веб-приложения» обозначаются приложения, рассчитанные на исполнение в браузерах для настольных компьютеров. Они также смогут работать на мобильных устройствах, если не полагаются на специфические технологии, отсутствующие на многих мобильных устройствах (например, Adobe Flash).

Адаптивный веб-интерфейс

Приложения с адаптивным веб-интерфейсом автоматически меняют его внешний вид в зависимости от размера устройства — чаще всего при этом используется технология CSS (cascading style sheets). Дизайн может быть выбран сервером при доставке приложения, изменен на уровне клиента либо обоими способами. Смысл в том, чтобы контент из одного и того же источника отображался по-разному в зависимости от особенностей конкретного устройства. Данный вариант применим как для мобильных веб-приложений, так и для исполняемых на других видах устройств — например, на игровых консолях и телевизорах.

## 1 Описание предметной области

## Аналоги разрабатываемого приложения

Функциональные возможности браузеров постоянно расширяются и улучшаются благодаря конкуренции между их разработчиками и высоким темпом развития и внедрения информационных технологий. Несмотря на то, что браузеры разных изготовителей базируются на разных технологических решениях, большинство современных браузеров придерживается международных стандартов и рекомендаций W3C в области обработки и отображения данных. Стандартизация позволяет добиться предсказуемости в визуальном представлении информации конечному пользователю независимо от технологии, которая использована для ее отображения в браузере. В последние годы многие разработчики браузеров сосредоточили свои усилия на повышении удобства пользовательского интерфейса браузеров для их использования в аппаратных устройствах, в которых применяются сенсорные экраны.

Браузеры распространяются, как правило, бесплатно. Потребителям браузер может быть поставлен в форме самостоятельного (автономного) приложения или в составе комплектного программного обеспечения

Разрабатываемое приложения является аналогом таких браузеров как:

1) Яндекс браузер - браузер, созданный компанией «Яндекс» на основе движка Blink, используемого в открытом браузере Chromium. Впервые был представлен 1 октября 2012 года на технологической конференции Yet another Conference (Рисунок 1)



Рисунок 1 – Яндекс Браузер

2) Google Chrome — браузер, разрабатываемый компанией Google на основе свободного браузера Chromium и движка Blink (до апреля 2013 года использовался WebKit). (Рисунок 2)



Рисунок 2 – Google Chrome

Вывод: Яндекс - более информационный поисковик и обрабатывает лучше информационные запросы.

Гугл - обрабатывает лучше ваши навигационные запросы. Если вы захотите найти сайт какой-то компании, то Гугл с большей вероятностью выдаст вам его на первое место в поиске.

Разработанное приложение предоставляет большинство функций предоставляемых описанными выше.

## Техническое задание

1) Главная идея приложения.

Главной идеей является создание быстрого, «легкого» и понятного в оформлении мобильного web-браузера. Он будет отличаться небольшим размером, удобством в работе, хорошей скоростью работы, прекрасно подходит для работы на машинах с небольшим объемом ОЗУ и маломощным CPU.

2) Основные задачи:

1. Обеспечить удобность при использовании мобильного Web-браузера.

2. Автоматизация получения доступа к данным.

3. Создание простого, понятного и удобного интерфейса.

4. Обеспечить высокую скорость работы приложения.

3) Основные пользователи приложения.

Основные пользователи приложения – Разработчики и непосредственно пользователи

Разработчики осуществляют непосредственно создание приложения, его загрузку в сеть, поддержку и развитие

Пользователями приложения выступают люди всех возрастов, являющиеся пользователями Android-смартфонов, все кто используют интернет для получения необходимой информации.

Для разработки UML – диаграммы можно воспользоваться онлайн редактором (Рисунок 3).

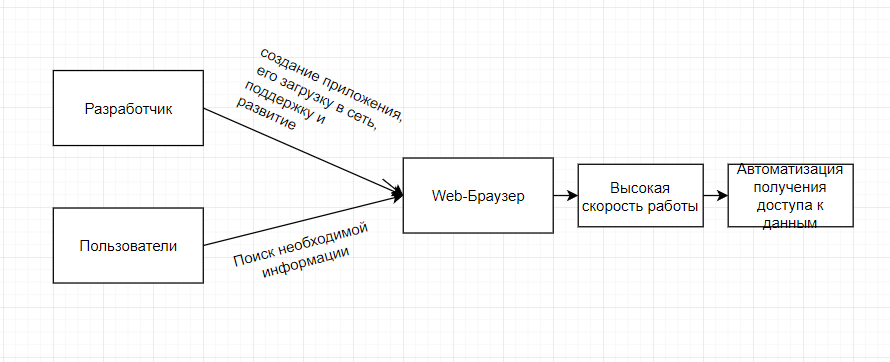


Рисунок 3 – UML-диаграмма

## 1.3 Описание структуры приложения

Структурная схема будущего приложения (Рисунок 4).

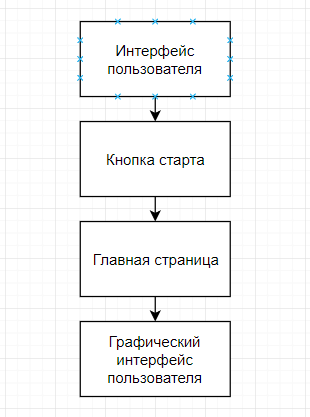


Рисунок 4 – Структурная схема будущего приложения

Функциональные требования:

1) отображение содержания web-сайтов;

2) высокая скорость загрузки страниц;

3) осуществление перехода от одной страницы к другой.

## 2 Описание Разработки Приложения

Вводим название приложения. Выбор минимальной версии оставляем по умолчанию. В окне выбора шаблона выбираем Empty Activity. Остальные настройки оставляем по умолчанию.

Изменяем макет главного экрана.

В файле манифеста добавить разрешение на доступ к интернету.

<uses-permission android:name="android.permission.INTERNET" />

Теперь создадим класс для веб-клиента.

Класс myWebClient унаследован от WebViewClient. Мы определяем собственный веб-клиент для того, чтобы ссылки открывались в нем, а не в системном WebView.

Здесь переопределяем метод shouldOverrideUrlLoading, в котором вызываем метод загрузки loadUrl(url).

Сначала определяем переменные для прогрессбара, поля ввода, вебвью и кнопок.

Затем в методе onCreate инициализируем все эти объекты.

Методом setWebViewClient устанавливаем наш веб-клиент для вебвью.

Затем в методе setWebChromeClient устанавливаем прогрессбар, который будет отображать процесс загрузки страницы.

Затем включаем поддержку JavaScript и загружаем url по умолчанию — это будет страница Google.

сталось присвоить кнопкам слушатели и определить действия по нажатию каждой кнопки.

При нажатии кнопки Отправки запроса будем считывать значение текстового поля, преобразовывать его в строку и присваивать переменной url. Затем проверяем переменную url на наличие префикса http:// и добавляем его, если он отсутствует. Передаем url браузеру WebView.

После создаем экземпляр класса InputMethodManager и вызываем его метод hideSoftInputFromWindow для скрытия клавиатуры после нажатия кнопки отправки адреса.

Следующая кнопка — Вперед, которая загружает страницу, покинутую в результе возврата по кнопке Назад (имеется в виду кнопка Назад нашего приложения, а не системная). Здесь через проверку возможности перехода вперед по истории методом WebView canGoForward() вызываем метод WebView goForward() для перехода.

Далее обрабатываем кнопку Назад, где по нажатию проверяем методом WebView canGoBack() возможность возврата по истории просмотров назад и вызываем метод WebView goBack().

Кнопка Обновить по нажатию вызывает метод WebView reload() для обновления страницы.

Запускаем приложение на устройстве.

## 2.1 Обоснование средств разработки

Выбор средств разработки:

1) Java (Рисунок 5);

Это официальный язык для разработки под Android, который поддерживается Android Studio и является основой для изучения Kotlin.

Основные преимущества языка:

- больше материалов и легче найти информацию;

- большое комьюнити;

- среда Android Studio изначально заточена под работу с Java;



Рисунок 5 – Java

2) C/C++ (Рисунок 6);

C-языки позиционируются как высокопроизводительные, и это важно, особенно если речь идет о чём-то тяжеловесном вроде мобильной 3D-игры.

Но стоит сразу оговориться, что прибегнуть к Java всё-таки придётся: без этого языка невозможно создать полноценное мобильное приложение. Android NDK позволит написать на C/C++ только часть приложения, например библиотеку, которую через Java можно подключить к программе.

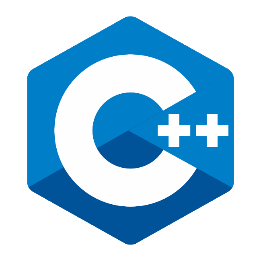


Рисунок 6 – C/C++

3) Python (Рисунок 7);

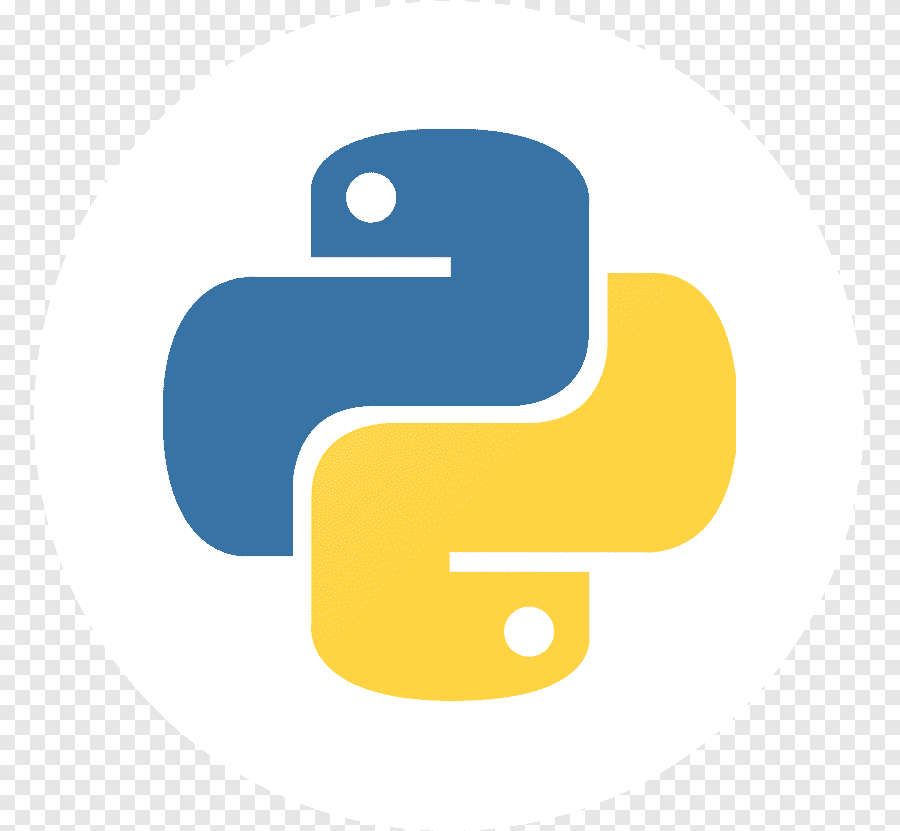


Рисунок 7 – Python

Язык программирования приспособили к разработке мобильных приложений с помощью Kivy и BeeWare:

Kivy — это open source библиотека для создания кроссплатформенных приложений, в том числе под Android и iOS (Рисунок 8).



Рисунок 8 – Kivy

BeeWare — это набор UI-инструментов для создания нативных Android-приложений (Рисунок 9).



Рисунок 9 – BeeWare

Программное средство было реализовано на языке Java. Создание программы будет предполагаться по заранее заданным критериям.

Браузер - это прикладное программное обеспечение для просмотра веб-страниц; содержания веб-документов, компьютерных файлов и их каталогов; управления веб-приложениями; а также для решения других задач. В глобальной сети браузеры используют для запроса, обработки, манипулирования и отображения содержания веб-сайтов. Многие современные браузеры также могут использоваться для обмена файлами с серверами ftp, а также для непосредственного просмотра содержания файлов многих графических форматов (gif, jpeg, png, svg), текстовых форматов (pdf, djvu) и других файлов.

Пользовательский интерфейс должен быть удобен в использовании, прост в освоении, понятен.

Предполагается создание графического пользовательского интерфейса. Консольный интерфейс не предусмотрен.

К входным данным относится: ввод адреса сайта.

На выход должны выводиться: желаемый веб-ресурс, на который хочет перейти пользователь, либо страница об ошибке, если страница не была найдена.

Программное средство должно быть простым в сопровождении, легко расширяемым, модифицируемым.

## 2.2 Разработка интерфейса

Интерфейс приложения должен быть понятен, удобен и прост для пользователей всех возрастов.

В центре экрана расположена строка ввода. Она позволяет вводить текст и ссылки для поиска информации в сети Интернет.

Рядом со строкой поиска расположена кнопка «Найти». Она позволяет выполнить запрос введенный пользователем  
На главной странице под строкой поиска расположена информация о популярных запросах и лента новостей  
Подобный интерфейс является простым и удобным для пользователей. Главная страница содержит актуальную информацию, которая может быть полезна и интересна пользователю.

## 2.3 Разработка логики работы приложения

## Схема взаимодействия компонентов проекта

1) Activity(деятельность) – Визуальный интерфейс пользователя заполняет весь экран портативного устройства.

Приложение имеет несколько деятельностей:

1) Окно ввода запроса

Пользователь вводит запрос для поиска необходимой информации

2) Окно вывода результатов поиска

После ввода запроса приложение выводит пользователю информацию, соответствующую критериям поиска, ссылки на различные web-ресурсы: web-страницы, файлы, документы (Рисунок 10).

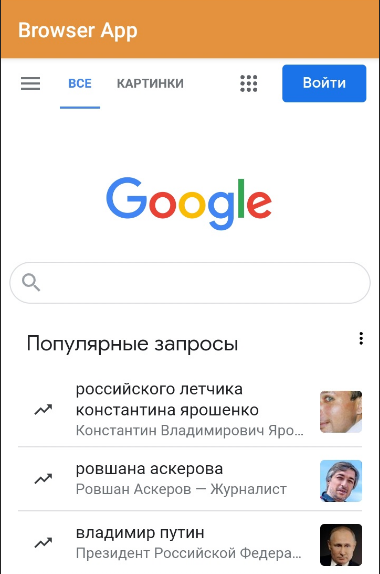


Рисунок 10 – Интерфейс браузера

2) Service (служба)

Разрабатываемому приложению (Мобильный Web-браузер) необходим доступ в интернет для просмотра веб-страниц, содержания веб-документов, компьютерных файлов и их каталогов. а также для решения других задач.

3)Broadcast Receiver

Во время работы мобильный Web-браузер будет обращаться к сторонним Web-ресурсам: страницам документам, файлам и т.д. для поиска необходимой информации и предоставления ее пользователю.

## 2.4 Описание переменных, компонентов, классов и подпрограмм

Используемые классы:

1. android.content.Intent - это асинхронные сообщения, позволяющие компонентам приложения запрашивать функциональность от других компонентов Android. Intents позволяют взаимодействовать с другими компонента из тех же приложений, так же, как и с компонентами созданные другими приложениями.
2. android.os.Bundle - представляет собой враппер (оболочку) над коллекцией ArrayMap для создания более комфортного в работе контейнера для элементов разных типов, в котором можно разместить любые Parcelabe-объекты и примитивные типы. Класс является потокобезопасным и может использоваться для передачи значений между разными потоками. Доступ к элементам этой коллекции осуществляется, как и в ArrayMap, по парам ключ-значение. Дополнительно класс предоставляет методы по размещению и извлечению типизированных данных и некоторые другие возможности.
3. android.view.View - View представляют собой элементы управления и прочие виджеты, например, кнопки, текстовые поля и т.д., через которые пользователь взаимодействует с программой
4. androidx.appcompat.app.AppCompatActivity - это базовый класс для действий, которые используют функциональные возможности панели поддержки библиотеки. Применяется для обратной совместимости в плане дизайна.

Методы:

1. Метод setContentView(int) – устанавливает содержимое Activity из layout-файла. Но в качестве аргумента мы указываем не путь к layout-файлу (res/layout/activity\_main.xml), а константу, которая является ID файла.
2. super.onCreate(savedInstanceState) - onCreate используется для запуска действия. super используется для вызова конструктора родительского класса.

<uses-permission android:name="android.permission.INTERNET"/> - Разрешение, которое открывает доступ в интернет.

<intent-filter> - Фильтр Intent представляет собой секцию в файле манифеста приложения, описывающее типы объектов Intent, которые компонент мог бы выполнить.

Таким образом, наличие фильтра Intent в описании активности в манифесте позволяет другим приложениям напрямую запускать данную операцию с помощью некоторого объекта Intent.

Классы:

1) android.os.Bundle; - android.os.Bundle - представляет собой враппер (оболочку) над коллекцией ArrayMap для создания более комфортного в работе контейнера для элементов разных типов, в котором можно разместить любые Parcelabe-объекты и примитивные типы. Класс является потокобезопасным и может использоваться для передачи значений между разными потоками. Доступ к элементам этой коллекции осуществляется, как и в ArrayMap, по парам ключ-значение. Дополнительно класс предоставляет методы по размещению и извлечению типизированных данных и некоторые другие возможности.

2) WebView — это компонент платформы Android, который позволяет встраивать web-страницы в Android-приложения.

3) AppCompatActivity была представлена в Android-SDK с момента выпуска библиотеки appcompat поддержки Android.

AppCompatActivity – это прямой дочерний класс FragmentActivity поддержки v4 и прямой родительский класс ActionBarActivity. Это базовый класс для действий, которые используют функциональные возможности панели поддержки библиотеки, применяется для обратной совместимости в плане дизайна

Методы:

1. OnCreate() - первый метод, с которого начинается выполнение активности. В этом методе активность переходит в состояние Created. Он вызывается непосредственно после создания активности.
2. onBackPressed() – переопределяет поведение кнопки.

## Тестирование и установка приложения

Разработанный мобильный web-браузер запускается на устройстве на полный экран. Проблемы с отображением элементов отсутствуют. Приложение имеет доступ к интернету, что позволяет ему отображать для пользователя необходимую и актуальную информацию. Мобильный Web-браузер работает стабильно. Скорость работы высокая. Информация, предоставляемая приложением, соответствует критериям поиска, которые указал пользователь.

При запуске приложения возникала проблема с доступом в интернет. У приложения отсутствовали необходимые разрешения.

Проблема была устранена.

Параметры скомпилированного файла

Приложение автоматически подстраивается под размер экрана устройства, что позволяет корректно отображать все графические элементы.

Размер скомпилированного приложения - 11 Мб.

## Заключение

В процессе выполнения курсового проекта приобрёл практический опыт работы с языком программирования Java и создании собственного мобильного приложения. Реализовал свои умения в работе с современными case – средствами, формировании и настройке работы в выбранной среде разработки. Применил на практике свои знания об основных положениях теории объектно-ориентированного программирования, основных принципах построения концептуальной, логической и физической модели приложения, проработке удобного интерфейса, основных типах сетевых топологий, приемах работы в компьютерных сетях. Разработал собственный мобильный web-браузер для android-смартфонов с удобным интерфейсом и понятным интерфейсом

## Список используемой литературы

1. Голощапов А.Л. Android. Создание приложений для смартфонов и планшетных ПК, СПб: БХВ-Петербург, 2016 г.

2. Дэрси, Л. Разработка приложений для Android-устройств. Т. 1: Базовые принципы / Л. Дэрси, Ш. Кондер. - М.: Лори, 2018. - 402 c.

3. Колисниченко, Денис Программирование для Android / Денис Колисниченко. - М.: БХВ-Петербург, 2015. - 706 c.

4. web-ресурс для построения UML-диаграмм diagrameditor - <https://www.diagrameditor.com>

5. Уроки по созданию приложения <http://www.fandroid.info/urok-po-sozdaniyu-prostogo-brauzera-v-android-studio/>

6. Разработка приложений на Android <https://proglib.io/p/razrabotka-prilozheniy-na-android-s-nulya-ustanovka-sredy-razrabotki-2020-12-31>

7. Как начать разрабатывать под Android <https://tproger.ru/translations/how-to-start-android/>?

8. Как создать мобильное приложение в Android Studio <https://wnfx.ru/kak-sozdat-prilozhenie-v-android-studio/?ysclid=l2srq1k8m6>

9. Создание приложений для Android <https://www.internet-technologies.ru/articles/sozdanie-pervogo-android-prilozheniya-vse-chto-vam-nuzhno-znat.html>

10. Как написать простое Android приложение на Java <https://temofeev.ru/info/articles/kak-napisat-prostoe-android-todo-prilozhenie-na-java/>?

## Приложение А. Код программы

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="1" android:versionName="1.0" android:compileSdkVersion="29" android:compileSdkVersionCodename="10" package="com.dude.broser" platformBuildVersionCode="29" platformBuildVersionName="10">

<uses-sdk android:minSdkVersion="20" android:targetSdkVersion="29"/>

<uses-permission android:name="android.permission.INTERNET"/>

<application android:theme="@style/AppTheme" android:label="@string/app\_name" android:icon="@mipmap/ic\_launcher" android:debuggable="true" android:allowBackup="true" android:supportsRtl="true" android:roundIcon="@mipmap/ic\_launcher\_round" android:appComponentFactory="androidx.core.app.CoreComponentFactory">

<activity android:name="com.dude.broser.WebActivity"/>

<activity android:name="com.dude.broser.MainActivity">

<intent-filter>

<action android:name="android.intent.action.MAIN"/>

<category android:name="android.intent.category.LAUNCHER"/>

</intent-filter>

</activity>

</application>

</manifest>

Файл WebActivity.java

Листинг кода:

package com.dude.broser;

import android.os.Bundle;

import android.webkit.WebView;

import android.webkit.WebViewClient;

import androidx.appcompat.app.AppCompatActivity;

public class WebActivity extends AppCompatActivity {

private WebView webView;

/\* access modifiers changed from: protected \*/

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView((int) C0262R.layout.activity\_web);

this.webView = (WebView) findViewById(C0262R.C0264id.webview);

this.webView.setWebViewClient(new WebViewClient());

this.webView.loadUrl("https://www.google.com/");

this.webView.getSettings().setJavaScriptEnabled(true);

}

public void onBackPressed() {

if (this.webView.canGoBack()) {

this.webView.goBack();

} else {

super.onBackPressed();

}

}

}

Файл MainActivity.java

package com.dude.broser;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

/\* access modifiers changed from: protected \*/

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView((int) C0262R.layout.activity\_main);

}

public void LoadWebPage(View v) {

startActivity(new Intent(this, WebActivity.class));

}

}