МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра автоматизированных систем управления

Лабораторная работа №9

по дисциплине: «Современные технологии разработки программного обеспечения»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Выполнил:** |  | **Проверил:** |  | |
| Студент | *Чибисов И.А.* | Преподаватель | *Эстрайх И.В.*  *ст. преподаватель* | |
| Факультет | *АВТФ* |  |  | |
| Направление (специальность) подготовки | *09.04.03 – Прикладная информатика* | Балл: \_\_\_\_\_\_\_\_\_ | |  |
| Группа | *АПМ-22* | Оценка *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | | |
| Шифр | *010594214* |  |  | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| подпись | | подпись | | |
| Дата сдачи: | «\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г. | Дата защиты: | «\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г | |

Новосибирск

2023

Цель работы: научиться вызывать из своей программы стандартные приложения и передавать им данные, а также разработать свой браузер для замены стандартного

Задание на лабораторную работу

1. Создайте приложение, в котором проверяется наличие соединения с Интернетом. Интерфейс можно сделать простым – изображениекнопка в центре экрана. Подготовьте две картинки: если подключение отсутствует, то при нажатии на кнопку в качестве фона будет устанавливаться одна картинка (например, дверь с замком), а при наличии работающего Интернета - другая (например, открытая дверь). Кроме того, в зависимости от наличия или отсутствия соединения, должно выводиться соответствующее всплывающее сообщение.

2. Создайте приложение, в котором пользователю предлагается несколько телефонных номеров и выбранный им номер передается приложению Телефон (после чего пользователю нужно будет нажать только кнопку вызова для звонка).

3. Создайте приложение, в котором пользователь заполняет поля

Добавьте на форму кнопки «Очистить» и «Отправить». Первая из них стирает всю набранную пользователем информацию, по нажатию второй пользователю предлагается выбрать почтовый клиент, и затем выводится e-mail сообщение с заполненными полями.

4. Создайте приложение, в котором пользователю предлагается несколько ссылок на веб-страницы, и выбранная им ссылка открывается во второй активности, где реализован собственный браузер на основе WebView, позволяющий переходить в нем же по ссылкам и возвращаться на предыдущую страницу (до открытия собственного браузера пользователь должен выбрать, с помощью чего открыть ссылку).

5. Включите в отчет цель работы, коды программ, скриншоты с результатами и вывод.

Приложение 1 Проверка интернета

Напишем метод проверки интернет-соединения, так же выведем тост, запустим приложение смотрим результат на рисунке 1 с включённым и выключенным интернетом.

protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState)**;** setContentView(R.layout.*activity\_main*)**;** imageView = findViewById(R.id.*imageView*)**;** connectivityManager = (ConnectivityManager) getSystemService(Context.*CONNECTIVITY\_SERVICE*)**;** updateConnectionStatus()**;**}  
private void updateConnectionStatus() {  
 NetworkInfo activeNetworkInfo = connectivityManager.getActiveNetworkInfo()**;** if (activeNetworkInfo != null && activeNetworkInfo.isConnected()) {  
 imageView.setImageResource(R.drawable.*yes*)**;** showToast("Есть подключение к Интернету")**;** } else {  
 imageView.setImageResource(R.drawable.*no*)**;** showToast("Нет подключения к Интернету")**;** }  
}  
private void showToast(String message) {  
 Toast.*makeText*(this**,** "Статус подключения: " + message**,** Toast.*LENGTH\_SHORT*).show()**;**}

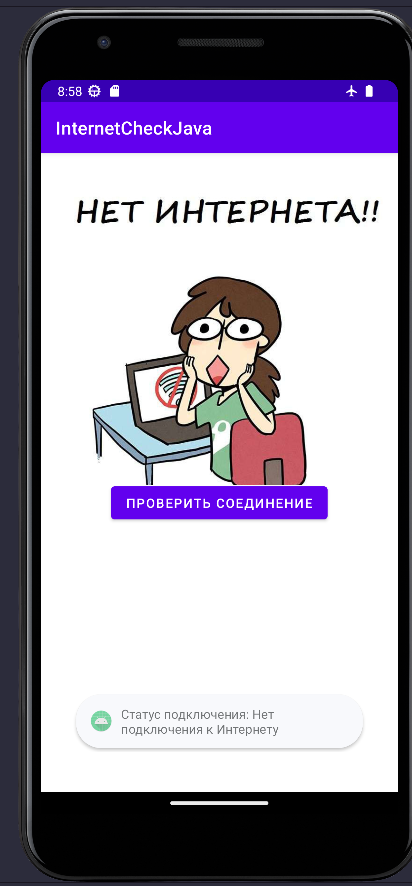
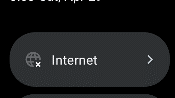
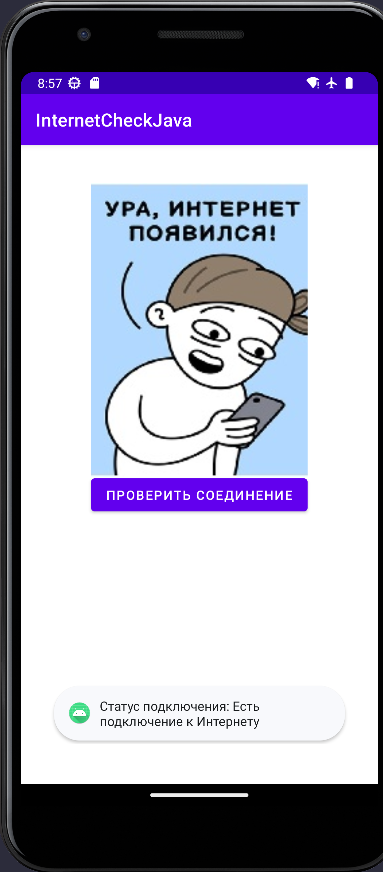


Рисунок 1 – приложение проверка интернета

Приложение 2 контакты

Создадим пустой проект добавим в макет пару TextView с номерами  
 Напишем метод по клику на номер будет открывается телефон с этим номером. Результат на рисунке 2

protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState)**;** setContentView(R.layout.*activity\_main*)**;** TextView number1 = findViewById(R.id.*number1*)**;** TextView number2 = findViewById(R.id.*number2*)**;** TextView number3 = findViewById(R.id.*number3*)**;** TextView number4 = findViewById(R.id.*number4*)**;** setTextViewClickListener(number1**,** R.string.*number79963807895*)**;** setTextViewClickListener(number2**,** R.string.*number79963807896*)**;** setTextViewClickListener(number3**,** R.string.*number79963807897*)**;** setTextViewClickListener(number4**,** R.string.*number79963807898*)**;**}  
private void setTextViewClickListener(TextView textView**,** final int numberResId) {  
 textView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String number = getString(numberResId)**;** Intent intent = new Intent(Intent.*ACTION\_DIAL***,** Uri.*fromParts*("tel"**,** number**,** null))**;** startActivity(intent)**;** }  
 })**;**

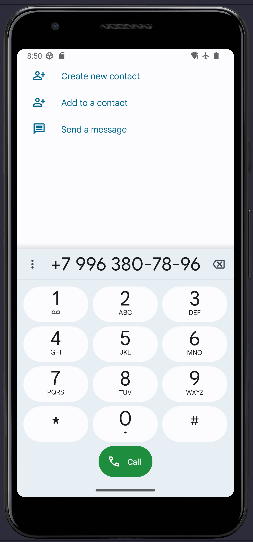
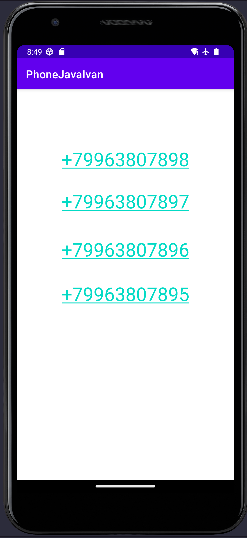


Рисунок 2 – приложение контакты

Приложение 3 форма для письма

Создадим простую форму с полями для ввода и кнопками. Напишем пару методов для отправки в почтовый клиент и отчески, так же неободимо войти в gmail что бы можно было продемонстрировать открытие приложения почты. Код представлен ниже, а работа приложения на рисунке 3.

mToEditText = findViewById(R.id.*mail*)**;** mSubjectEditText = findViewById(R.id.*theme*)**;** mMessageEditText = findViewById(R.id.*message*)**;** Button sendButton = findViewById(R.id.*send*)**;** sendButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 sendEmail()**;** }  
 })**;** Button clearButton = findViewById(R.id.*clear*)**;** clearButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 clearFields()**;** }  
 })**;**}  
  
private void sendEmail() {  
 String to = mToEditText.getText().toString()**;** String subject = mSubjectEditText.getText().toString()**;** String message = mMessageEditText.getText().toString()**;** Intent emailIntent = new Intent(Intent.*ACTION\_SENDTO***,** Uri.*parse*(  
 "mailto:" + to +  
 "?subject=" + Uri.*encode*(subject) +  
 "&body=" + Uri.*encode*(message)))**;** emailIntent.putExtra(Intent.*EXTRA\_SUBJECT***,** subject)**;** emailIntent.putExtra(Intent.*EXTRA\_TEXT***,** message)**;** startActivity(Intent.*createChooser*(emailIntent**,** "Send Email"))**;**}  
  
  
private void clearFields() {  
 mToEditText.setText("")**;** mSubjectEditText.setText("")**;** mMessageEditText.setText("")**;**}

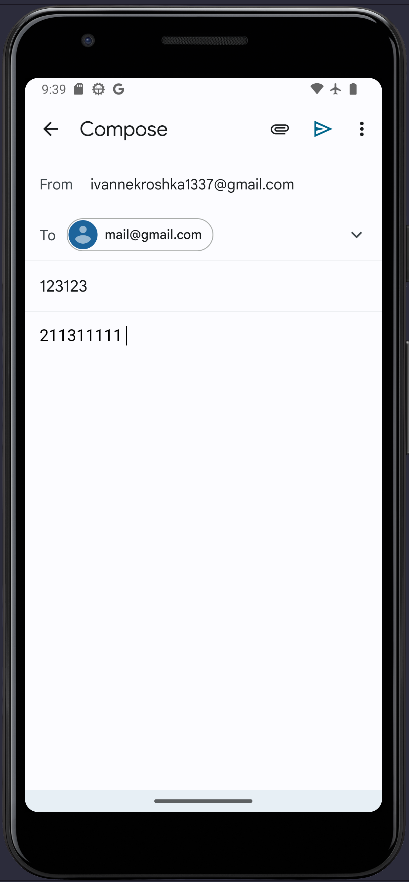
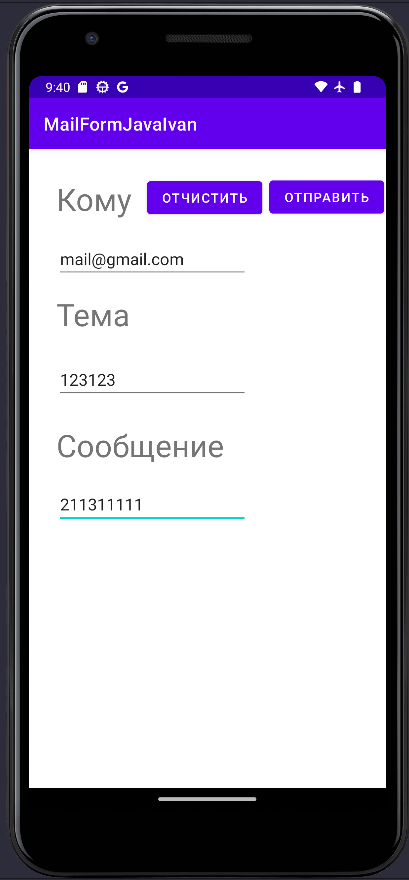
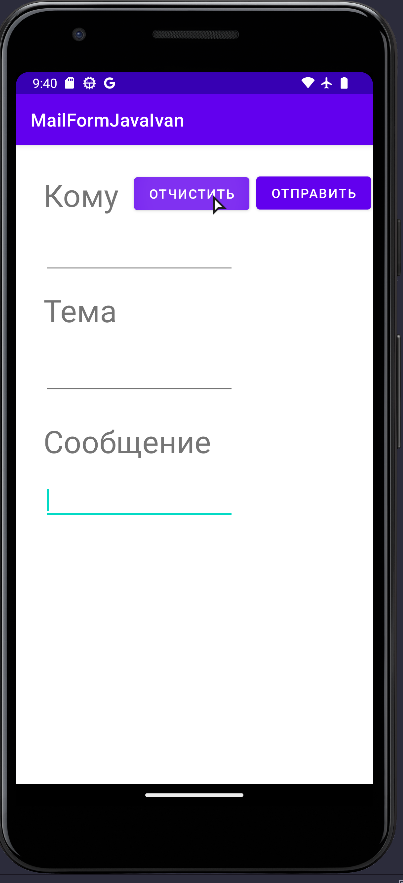
 

Рисунок 3 – форма для письма

Приложение 4 Браузер

Создадим пустой проект в MainActivity добавили пару кликабельных ссылок в activity\_main.xml.

MainActivity выглядит так

public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState)**;** setContentView(R.layout.*activity\_main*)**;** TextView link1 = findViewById(R.id.*link1*)**;** TextView link2 = findViewById(R.id.*link2*)**;** TextView link3 = findViewById(R.id.*link3*)**;** link1.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 openLinkDialog(getString(R.string.*link1*))**;** }  
 })**;** link2.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 openLinkDialog(getString(R.string.*link2*))**;** }  
 })**;** link3.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 openLinkDialog(getString(R.string.*link3*))**;** }  
 })**;** }  
  
 private void openLinkDialog(final String url) {  
 AlertDialog.Builder builder = new AlertDialog.Builder(this)**;** builder.setTitle("Выберите действие")  
 .setMessage("Открыть ссылку в браузере или в WebViewActivity?")  
 .setPositiveButton("Браузер"**,** new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog**,** int which) {  
 openLinkInBrowser(url)**;** }  
 })  
 .setNegativeButton("WebViewActivity"**,** new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog**,** int which) {  
 openLinkInWebView(url)**;** }  
 })  
 .show()**;** }  
  
 private void openLinkInBrowser(String url) {  
 Intent intent = new Intent(Intent.*ACTION\_VIEW*)**;** intent.setData(Uri.*parse*(url))**;** startActivity(intent)**;** }  
  
 private void openLinkInWebView(String url) {  
 Intent intent = new Intent(MainActivity.this**,** WebViewActivity.class)**;** intent.putExtra("url"**,** url)**;** startActivity(intent)**;** }

WebViewActivity визуально это кнопка назад и webView на весь экран.

WebViewActivity код:

public class WebViewActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState)**;** setContentView(R.layout.*activity\_web\_view*)**;** WebView mWebView = findViewById(R.id.*webView*)**;** Button mBackButton = findViewById(R.id.*backButton*)**;** // Получаем ссылку на открываемую страницу из MainActivity  
 String url = getIntent().getStringExtra("url")**;** // Настраиваем WebView  
 mWebView.getSettings().setJavaScriptEnabled(true)**;** mWebView.setWebViewClient(new WebViewClient())**;** mWebView.loadUrl(url)**;** // Обработчик нажатия на кнопку "Назад"  
 mBackButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 onBackPressed()**;** }  
 })**;** }  
}

Демонстрация приложения представлена на рисунке 4-5 с выбором, где открыть ссылку.

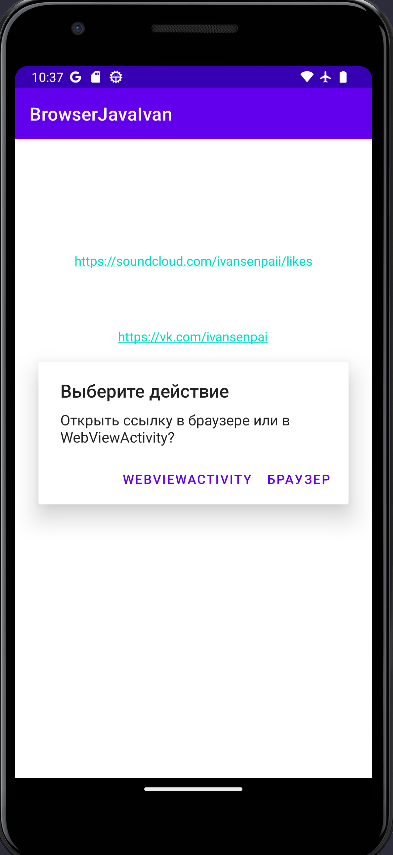
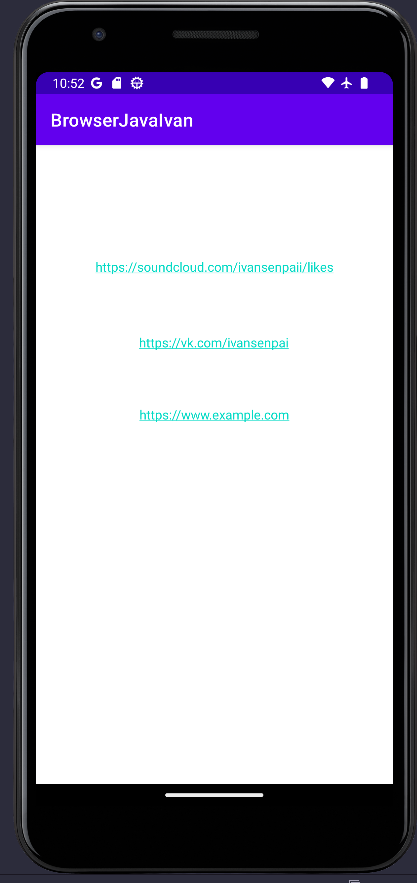


Рисунок 4 – откртытие ссылок в браузере и в WebView

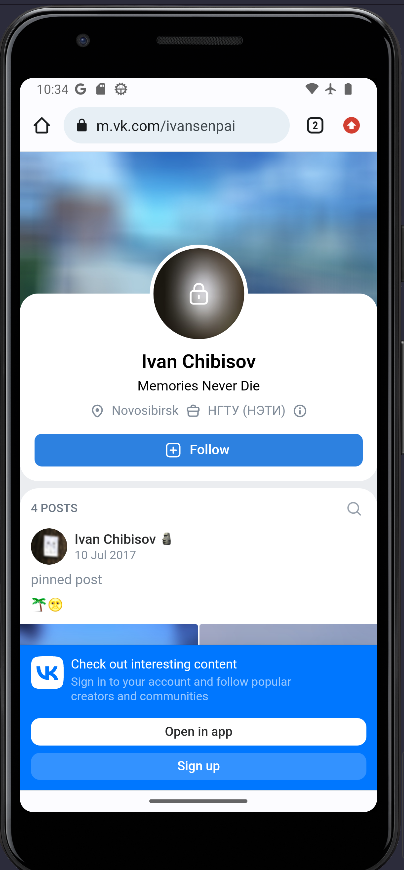
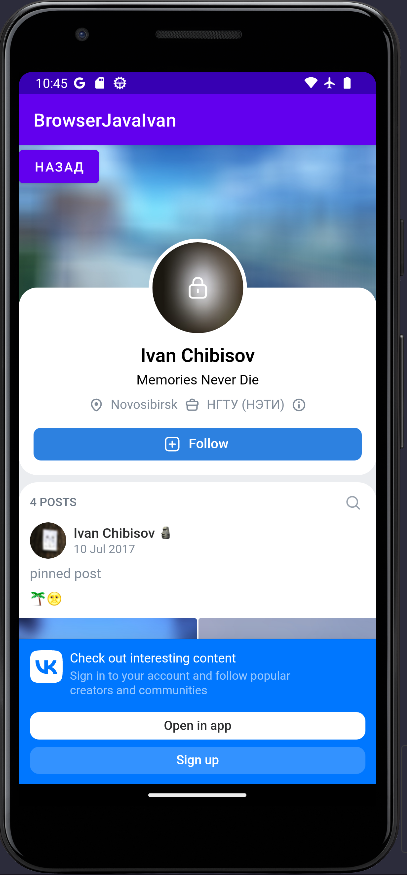
 

Рисунок 4 – откртытие ссылок в браузере и в WebView

Вывод: мы научились вызывать из своей программы стандартные приложения и передавать им данные, а также разработать свой браузер для замены стандартного.