

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені
ТАРАСА ШЕВЧЕНКА**



ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра прикладних інформаційних систем

Звіт до лабораторної роботи №8

з курсу

«Архітектура мобільних операційних систем і пристроїв»

*Студентки 4 курсу
групи ПП-41*

спеціальності 122 «Комп'ютерні науки»

ОП «Прикладне програмування»

Штось Софії Максимівни

Викладач:

Краснощок В.М.

Київ – 2023

Тема роботи: Робота із контент-провайдерами.

Мета роботи: Отримати практичні навички створення власних контент-провайдерів.

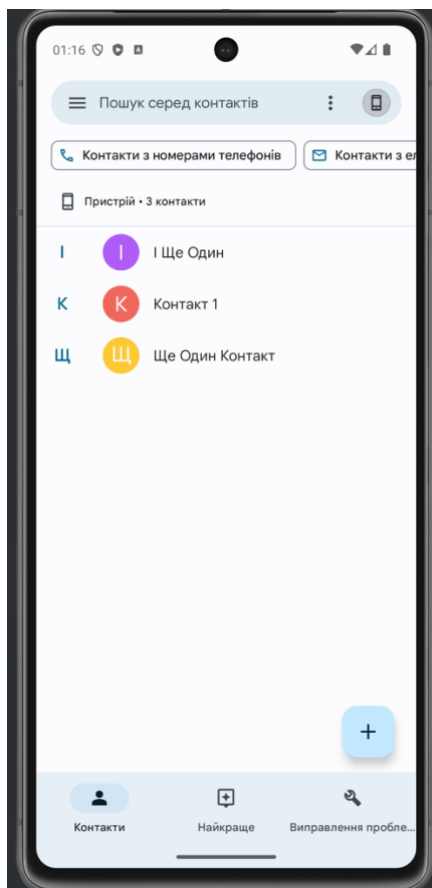
Завдання для виконання

Виконайте завдань до виконання у відповідності до теоретичних відомостей поданих вище.

Хід роботи

Завдання 1.

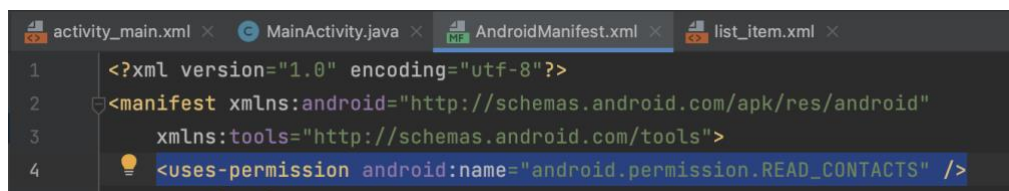
Створення контактів:



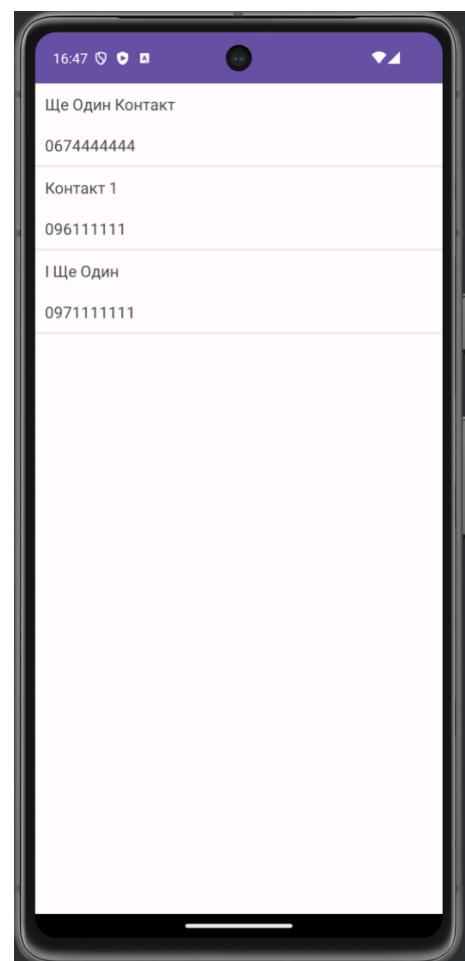
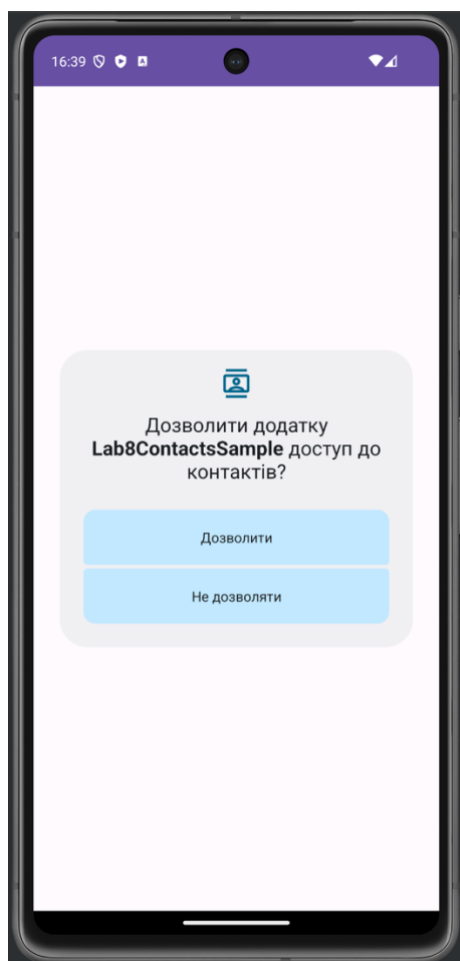
```
public class MainActivity extends ListActivity {
    ListView lv;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        lv = getListView();
        if (checkSelfPermission(android.Manifest.permission.READ_CONTACTS) ==
PackageManager.PERMISSION_GRANTED) {
            get(getListView());
        } else {
            requestPermissions(new
```

```
String[] {android.Manifest.permission.READ_CONTACTS}, 1);
    }

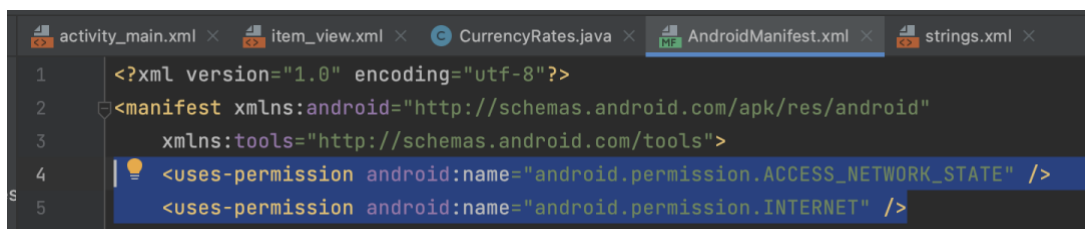
    public void get(View v){
        Cursor cursor =
getContentResolver().query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
        null, null, null, null);
        startManagingCursor(cursor);
        String[] from = {ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME,
ContactsContract.CommonDataKinds.Phone.NUMBER,
        ContactsContract.CommonDataKinds.Phone._ID};
        int[] to = {R.id.contact_name, R.id.contact_number};
        SimpleCursorAdapter simpleCursorAdapter = new SimpleCursorAdapter(this,
R.layout.list_item, cursor, from, to);
        lv.setAdapter(simpleCursorAdapter);
        lv.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);
    }
}
```



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.READ_CONTACTS" />
```



Завдання 2. «Робота із віддаленою БД»



Item_view.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal" >
    <TextView
        android:id="@+id/charCodeView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#FF8"
        android:minWidth="45sp"
        android:padding="4dp"
        android:textColor="#00F"
        android:textStyle="bold"
        android:gravity="center"
        android:shadowDx="8"
        android:shadowDy="8"
        android:shadowColor="#000"
        android:shadowRadius="8" />
    <TextView
        android:id="@+id/valueView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#008"
        android:background="#FFE"
        android:minEms="3"
        android:padding="3dp" />
    <TextView
        android:id="@+id/nominalView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="3dp" />
    <TextView
        android:id="@+id/nameView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ellipsize="marquee"
        android:singleLine="true" />
</LinearLayout>
```

CurrencyRates.java:

```
public class CurrencyRates extends ListActivity {

    private final static String KEYCHARCODE = "targetCurrency";
    private final static String KEYVALUE = "exchangeRate";
    private final static String KEYNAME = "targetName";
    ArrayList<Map<String, String>> data;
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    Thread thread = new Thread(new Runnable() {
        @Override
        public void run() {
            try {
                data = getData();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
    thread.start();
    try {
        thread.join();
    } catch (InterruptedException e) {
        System.out.println("got interrupted!");
    }
    //Log.d("CUSTOM_TAG", String.valueOf(data.size()));
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            String[] from = {KEYCHARCODE, KEYVALUE, KEYNAME };
            int[] to = {R.id.charCodeView, R.id.valueView, R.id.nameView };
            SimpleAdapter sa = new SimpleAdapter(getApplicationContext(), data,
                R.layout.item_view,
                from, to);
            setListAdapter(sa);
            // Stuff that updates the UI

        }
    });
}

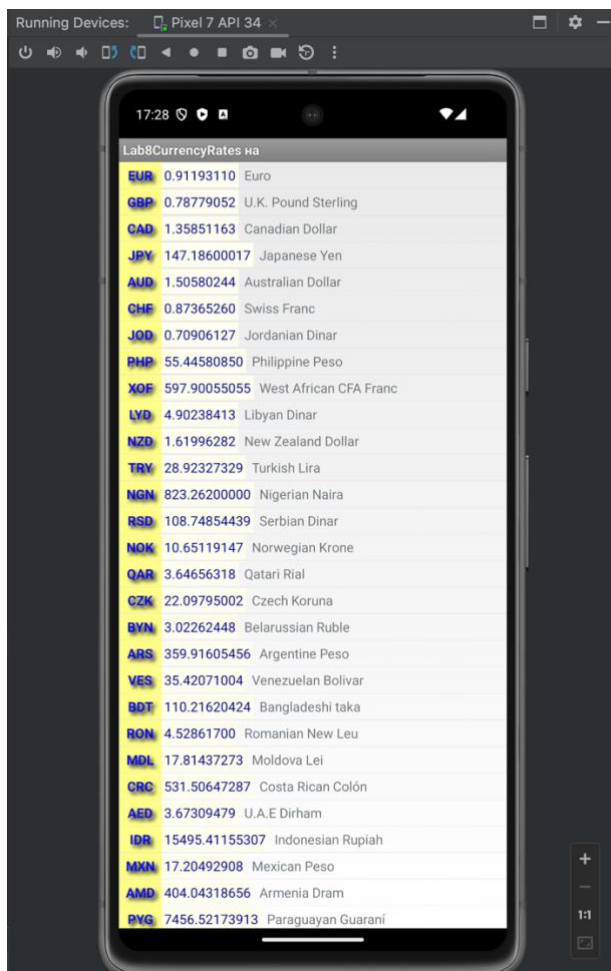
private ArrayList<Map<String, String>> getData() {
    ArrayList<Map<String, String>> list = new ArrayList<Map<String, String>>();
    Map<String, String> m;
    try{
        URL url = new URL(getString(R.string.rates_url));
        HttpURLConnection httpConnection =
            (HttpURLConnection) url.openConnection();
        int responseCode = httpConnection.getResponseCode();
        if (responseCode == HttpURLConnection.HTTP_OK) {
            InputStream in = httpConnection.getInputStream();
            DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
            DocumentBuilder db = dbf.newDocumentBuilder();
            Document dom = db.parse(in);
            Element docElement = dom.getDocumentElement();
            String date = docElement.getAttribute("Currency Name");
            setTitle(getTitle() + " на " + date);
            NodeList nodeList = docElement.getElementsByTagName("item");
            int count = nodeList.getLength();
            if (nodeList != null && count > 0) {
                for (int i = 0; i < count; i++) {
                    Element entry = (Element) nodeList.item(i);
                    m = new HashMap<String, String>();
                    String charCode =
entry.getElementsByTagName(KEYCHARCODE).item(0).getFirstChild().getNodeValue();
                    String value =
entry.getElementsByTagName(KEYVALUE).item(0).getFirstChild().getNodeValue();
                    String name =

```

```

entry.getElementsByTagName(KEYNAME).item(0).getFirstChild().getNodeValue();
        m.put(KEYCHARCODE, charCode);
        m.put(KEYVALUE, value);
        m.put(KEYNAME, name);
        list.add(m);
    }
}
else {
}
}
catch (MalformedURLException e) {
    e.printStackTrace();
}
catch (IOException e) {
    e.printStackTrace();
}
catch (ParserConfigurationException e) {
    e.printStackTrace();
}
catch (SAXException e) {
    e.printStackTrace();
}
return list;
}
}

```



Висновок: Отже, у ході цієї лабораторної роботи було проведено роботу з контент-провайдерами. Було здобуто навички створення власних контент-провайдерів.

Контрольні запитання

1. *Для чого потрібен Content Provider?* Content Provider служить для обміну даними між різними додатками в Android, забезпечуючи стандартизований інтерфейс доступу до даних.
2. *Чи кожен Content Provider надає відкритий URI?* Ні, не кожен Content Provider надає відкритий URI; доступ до даних контент-провайдера може бути обмеженим або захищеним.
3. *Як задекларувати Content Provider у файлі маніфесту додатку?* Для задекларування Content Provider у файлі маніфесту слід додати елемент <provider> з відповідними атрибутами, такими як authorities і name.
4. *Які параметри необхідні для того, щоб зробити запити до Content Provider?* Для зроблення запитів до Content Provider необхідно вказати URI, який вказує на дані, а також визначити необхідні параметри, такі як стовпці або умови вибору даних