

ĐỀ CƯƠNG BÀI GIẢNG

BÀI 8. THỰC HÀNH LƯU TRỮ DỮ LIỆU VỚI FILE

Nội dung bài học trước khi lên lớp (trang 1 đến trang 6):

- Thực hành với Shared Preferences: Phiếu bài tập 8.1

Nội dung bài học thực hiện lên lớp (trang 7 đến hết):

- Thực hành lưu trữ dữ liệu trong thiết bị: Phiếu bài tập: 8.2, 8.3
- Thực hành lưu trữ dữ liệu với thẻ nhớ : Phiếu bài tập: 8.4

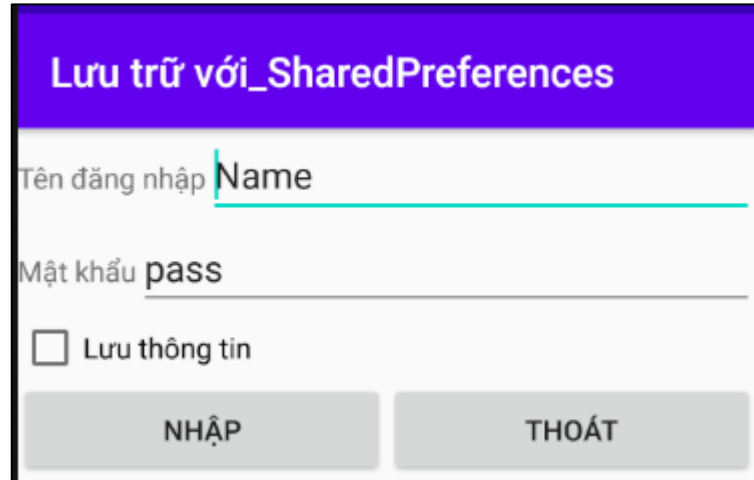
Nội dung bài học sau khi lên lớp: Phiếu bài tập: 8.5

NỘI DUNG BÀI HỌC

1. Thực hành với Shared Preferences..... 2
2. Thực hành lưu trữ dữ liệu trong thiết bị- Internal storage 9
3. Thực hành lưu trữ dữ liệu sử dụng bộ nhớ cache 15
4. Thực hành lưu trữ dữ liệu với thẻ nhớ - External storage..... 18

1. Thực hành với Shared Preferences

Cho màn hình giao diện sau



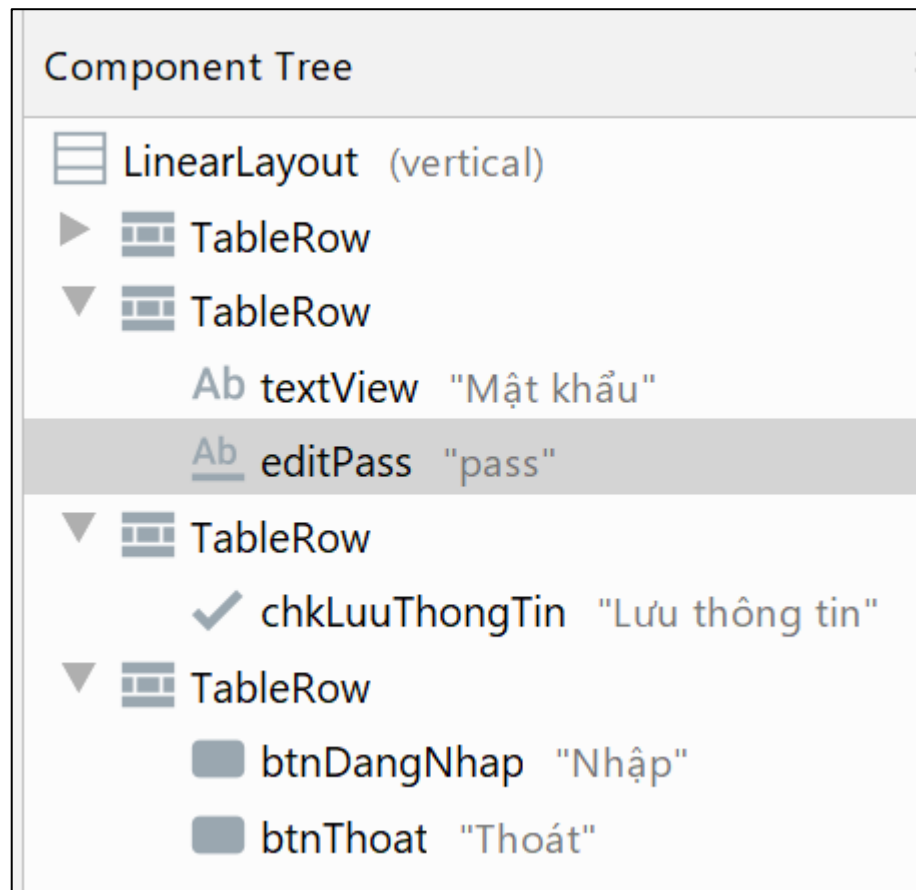
The screenshot shows a mobile application interface with a purple header bar containing the text "Lưu trữ với_SharedPreferences". Below the header, there are two text input fields. The first field is labeled "Tên đăng nhập" (Username) and contains the text "Name". The second field is labeled "Mật khẩu" (Password) and contains the text "pass". Below these fields is a checkbox labeled "Lưu thông tin" (Save information). At the bottom of the form, there are two buttons: "NHẬP" (Login) and "THOÁT" (Exit).

Màn hình gồm:

- 2 ô EditText nhập liệu tên đăng nhập và mật khẩu.
- 1 Checkbox lưu thông tin
- 2 Button nhập và thoát

Thực hiện nhập thông tin tên đăng nhập mật khẩu, chọn lưu thông tin thì lần trải nghiệm sau thông tin sẽ được nhớ và hiển thị.

Thiết kế giao diện ứng dụng: Cho cấu trúc cây thiết kế



Đặt lại tiêu đề hiển thị giao diện ứng dụng

strings.xml

```
<resources>
    <string name="app_name">Lưu trữ
với_SharedPreferences</string>
</resources>
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".MainActivity">
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

```
        android:text="Tên đăng nhập" />

        <EditText
            android:id="@+id/editTen"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="textPersonName"
            android:text="Name" />

    </TableRow>
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Mật khẩu" />

        <EditText
            android:id="@+id/editPass"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="textPersonName"
            android:layout_weight="1"
            android:text="pass" />
    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <CheckBox
            android:id="@+id/chkLuuThongTin"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Lưu thông tin" />
    </TableRow>
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:weightSum="2"
    >
```

```
<Button
    android:id="@+id/btnDangNhap"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Nhập" />

<Button
    android:id="@+id/btnThoat"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Thoát" />

</TableRow>
</LinearLayout>
```

MainActivity.java

```
import androidx.appcompat.app.AppCompatActivity;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
    EditText editTen, editPass;
    Button btnDangNhap, btnThoat;
    CheckBox chkLuuThongTin;
    String tenThongTinDangNhap="login";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editTen = (EditText) findViewById(R.id.editTen);
        editPass = (EditText) findViewById(R.id.editPass);
        btnDangNhap = (Button)
findViewById(R.id.btnDangNhap);
        btnThoat = (Button) findViewById(R.id.btnThoat);
        chkLuuThongTin = (CheckBox)
findViewById(R.id.chkLuuThongTin);
        btnDangNhap.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                saveLoginState();
            }
        });
    }
}
```

```
        }
    }
    );
    btnThoat.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        finish();
    }
});
}
@Override
protected void onPause() {
    super.onPause();
    saveLoginState();
}
public void saveLoginState(){
    SharedPreferences preferences=getSharedPreferences(
        tenThongTinDangNhap,MODE_PRIVATE);
    SharedPreferences.Editor editor=preferences.edit();
    editor.putString("UserName",
        editTen.getText().toString());
    editor.putString("PassWord",
        editPass.getText().toString());
    editor.putBoolean("Save",
        chkLuuThongTin.isChecked());
    editor.commit();
}
@Override
protected void onResume() {
    super.onResume();
    SharedPreferences preferences=
        getSharedPreferences(
            tenThongTinDangNhap,MODE_PRIVATE);
    String userName=
        preferences.getString("UserName", "");
    String pass=preferences.getString("PassWord", "");
    boolean save=preferences.getBoolean("Save", false);
    if (save) {
        editTen.setText(userName);
        editPass.setText(pass);
        chkLuuThongTin.setChecked(save);
    }
}
}
```

Kết quả lưu trong thiết bị

Data/data/<package ID>/shared_prefs/<file.xml?

Device File Explorer	
Emulator Pixel_XL_API_29 Android 8.1.0, API 27	
Name	Date
> com.google.android.youtube	2022-05-27 16:31
> com.google.ar.core	2022-05-27 16:30
> com.ustwo.lwp	2022-05-27 16:30
> jp.co.omronsoft.openwnn	2022-05-27 16:30
> org.chromium.webview_shell	2022-05-27 16:30
> vuduong.com	2022-05-27 16:37
> vuthiduong.m.bai1_hello_2022	2022-05-27 18:01
> vuthiduong.m.bai4_12_todolist	2022-05-27 23:36
> vuthiduong.m.bai4_spinner_listview_v2	2022-05-27 22:55
✓ vuthiduong.m.bai8_1_sharedpreference	2022-05-31 09:17
cache	2022-05-31 09:17
code_cache	2022-05-31 09:17
shared_prefs	2022-05-31 09:17
login.xml	2022-05-31 09:29

SharedPreferences pref=getSharedPreferences(“PREP”,MODE_PRIVATE)

nhận lấy thể hiện

Một số chế độ lưu trữ (MODE) được sử dụng

MODE_PRIVATE	Lưu trữ riêng tư không cho phép các ứng dụng khác có thể truy cập vào tập tin SharedPreferences trong ứng dụng.
MODE_WORLD_READABLE	Cho phép các ứng dụng khác chỉ có thể đọc dữ liệu từ tập tin SharedPreferences trong ứng dụng nhưng không được phép ghi dữ liệu.
MODE_WORLD_WRITEABLE	Cho phép các ứng dụng khác có thể ghi dữ liệu vào trong tập tin SharedPreferences trong ứng dụng.

Các phương thức lưu trữ dữ liệu của Editor

Các phương thức của Editor	Mô tả
putBoolean(Key, value)	Phương thức lưu giá trị Boolean
putFloat(Key, value)	Phương thức lưu giá trị Float
putInt(Key, value)	Phương thức lưu giá trị Integer
putLong(Key, value)	Phương thức lưu giá trị Long
putString(Key, value)	Phương thức lưu giá trị String
putSet(Key, value)	Phương thức lưu giá trị mảng giá trị String
Commit()	Xác thực trạng thái lưu vào XML

Các phương thức đưa dữ liệu vào SharedPreferences

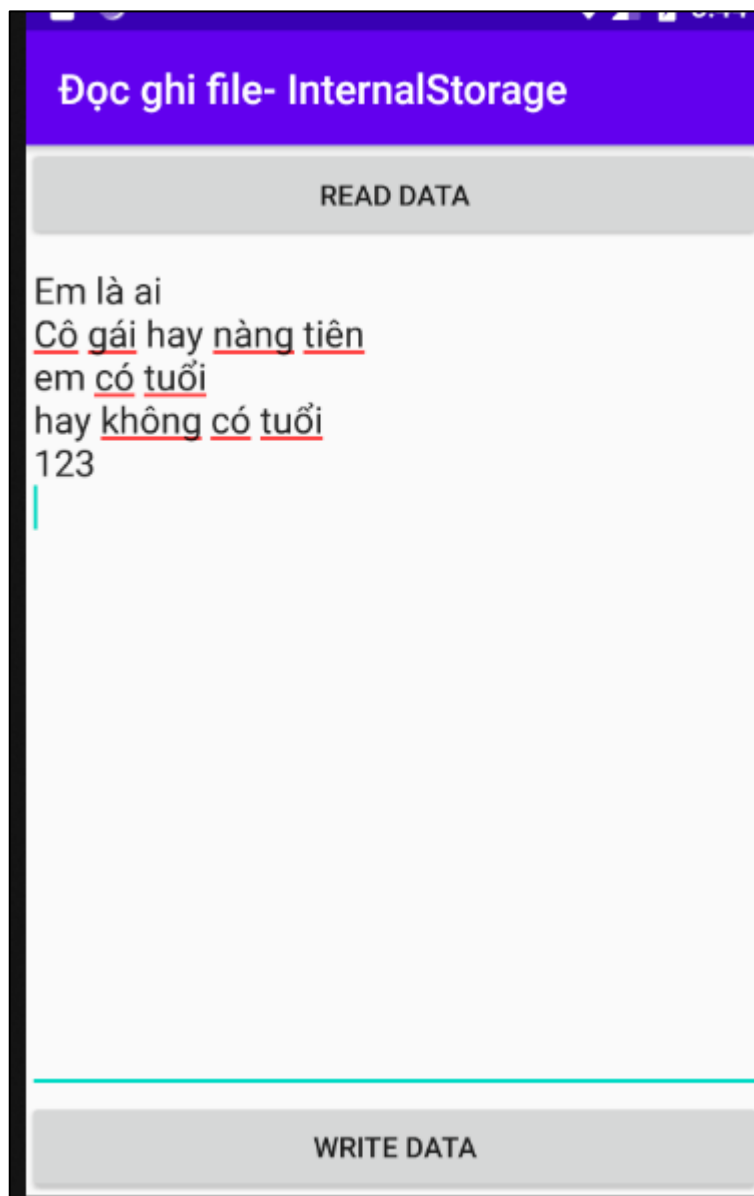
Các phương thức của Editor	Mô tả
apply()	Cập nhật dữ liệu mà không cần trả về kết quả thực thi lệnh thành công hay thất bại
commit()	Cập nhật dữ liệu và trả về kết quả là true nếu thực thi lệnh thành công và trả về false nếu thất bại
clear()	Xóa toàn bộ dữ liệu trong Shared Preference
remove(String key)	Xóa dữ liệu trong Shared Preference dựa vào “key” tương ứng.

Ghi chú:

- Lưu giá trị thông qua apply() vs commit ()
- commit() lưu trữ dữ liệu đồng bộ (synchronously)
- apply() là không đồng bộ (asynchronously).
- Sử dụng phương thức apply() sẽ thực thi lệnh nhanh hơn so với commit().

2. Thực hành lưu trữ dữ liệu trong thiết bị- Internal storage

Màn hình giao diện kết quả gọi ý sau.



Màn hình gồm:

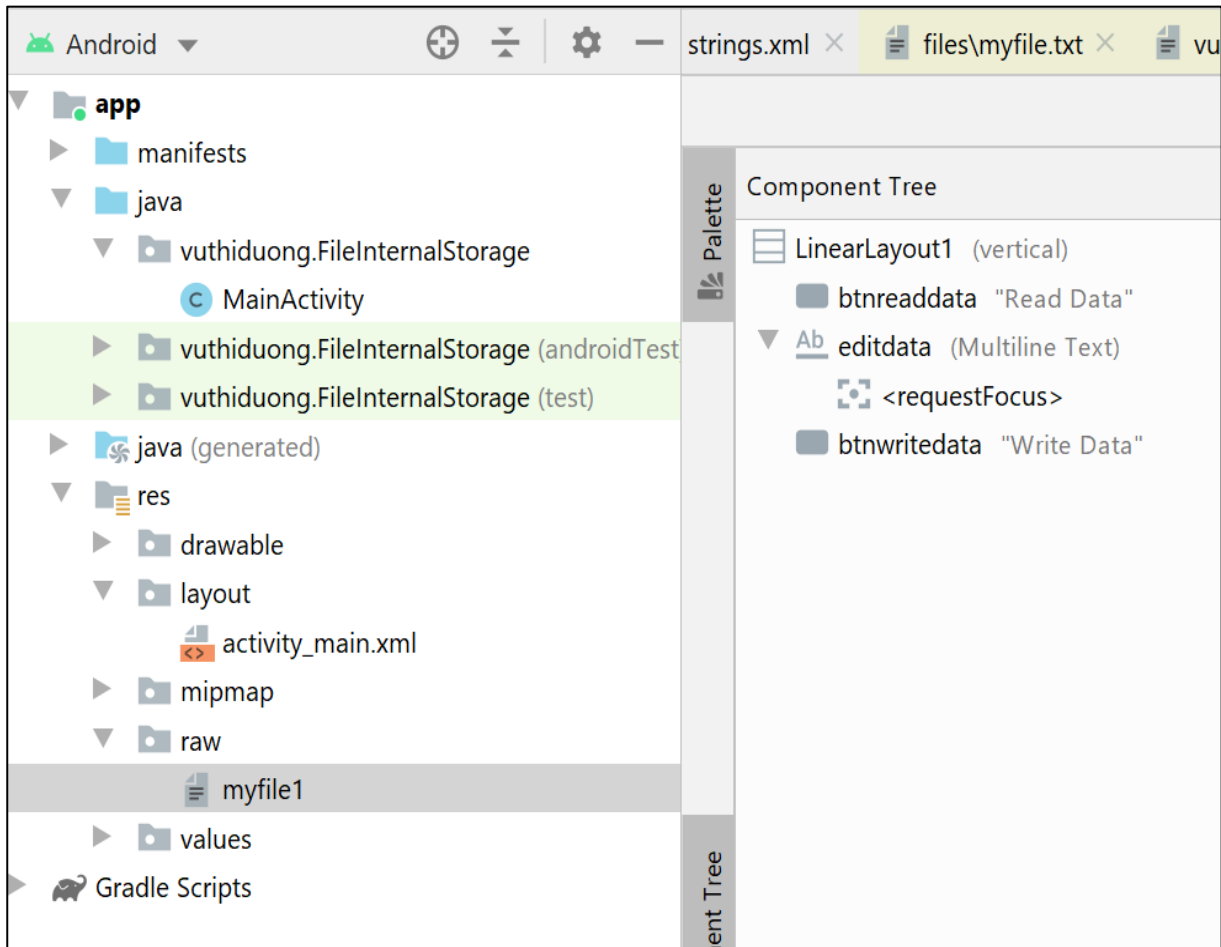
- 1 ô EditText nhập liệu
- 2 Button ReadData và WriteData

Thiết kế màn hình có thể điều chỉnh về kích thước và chiều cao cũng như hiển thị phù hợp với thiết bị cài đặt ứng dụng

Thực hiện: Tạo 1 file .txt trong cấu trúc thư mục ứng dụng. Khi chọn ReadData, ứng dụng thực hiện đọc file và hiển thị lên Textbox. Bổ sung thông tin soạn thảo, chọn WriteData, thông tin file lưu vào bộ nhớ.

Hướng dẫn cài đặt

1. Cấu trúc prj và cây thiết kế giao diện ứng dụng



Activity_main.xml

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/LinearLayout1"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context="vuthiduong.FileInternalStorage.MainActivity"
>
    <Button
        android:id="@+id/btnreaddata"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Read Data" />

    <EditText
        android:id="@+id/editdata"
        android:layout_width="match_parent"
        android:layout_height="0dip"
        android:layout_weight="2"
```

```
        android:ems="10"
        android:gravity="top|left"
        android:inputType="textMultiLine" >

        <requestFocus />
    </EditText>

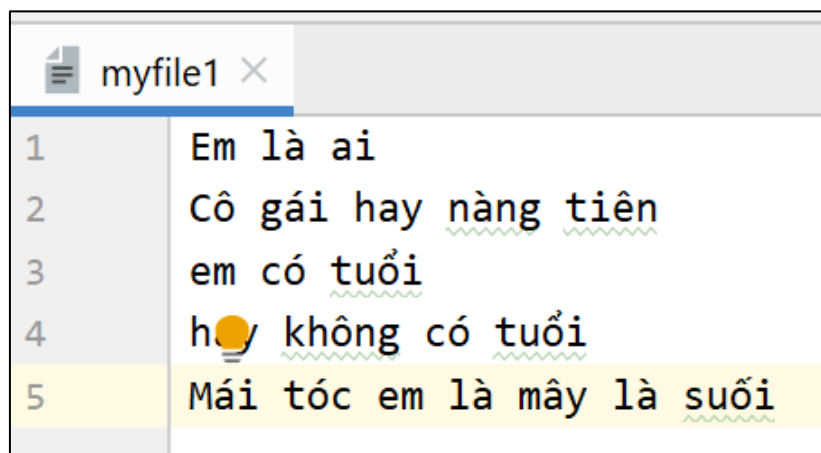
    <Button
        android:id="@+id/btnwritedata"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Write Data" />

</LinearLayout>
```

Có 2 cách đọc file:

- File đặt trong Resource File (Read only)
- Đọc và ghi tập tin trong thiết bị (Read & Write)

File Myfile1.txt. : đặt trong Resource File. Soạn thảo nội dung bất kỳ.



File có thể Read & Write: Thực hiện copy file đặt vào trong thư mục của ứng dụng.

Minh họa như hình sau:

Data/data/<package ID>/Files/<tên file>

Device File Explorer		
Emulator Nexus_5_API_27 Android 8.1.0, API 27		
Name	Date	Size
▶ folder	2020-11-24 1	4 KB
▶ folder	2020-09-15 1	4 KB
▶ folder	2022-05-17 1	4 KB
▶ folder	2020-11-17 1	4 KB
▶ folder	2020-11-25 1	4 KB
▶ folder	2022-05-25 1	4 KB
▼ folder	2022-05-25 1	4 KB
▶ folder	2021-07-28 1	4 KB
▶ folder	2021-08-03 1	4 KB
▶ folder	2021-07-28 2	4 KB
▼ folder	2022-05-25 1	4 KB
myfile.txt	2022-05-26 0	80 B

MainActivity.java

Phương thức ReadData sẽ đọc file trong Resource file, còn ReadData2 sẽ đọc file trong thư mục ứng dụng.

```
public class MainActivity extends AppCompatActivity implements
OnClickListener{
    Button btnread,btnwrite;
    EditText editdata;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getWindow();
        btnread.setOnClickListener(this);
        btnwrite.setOnClickListener(this);
    }
    //lấy các điều khiển trên giao diện
    private void getWindow() {
        btnread=(Button) findViewById(R.id.btnreaddata);
        btnwrite=(Button) findViewById(R.id.btnwritedata);
        editdata=(EditText) findViewById(R.id.editdata);
    }
}
```

```
//xử lý sự kiện chọn nút lệnh
public void onClick(View v) {
    if(v.getId()==R.id.btnreaddata)
    {
        readData();
        readData2();
    }
    else if(v.getId()==R.id.btnwritedata)
    {
        writeData();
    }
}
/**
 * Hàm đọc tập tin trong Android
 * Dùng openFileInput trong Android để đọc
 * ra FileInputStream
 */
public void readData()
{
    try {
        FileInputStream in= openFileInput("myfile.txt");
        BufferedReader reader=new BufferedReader(new
InputStreamReader(in));
        String data="";
        StringBuilder builder=new StringBuilder();
        while((data=reader.readLine())!=null)
        {
            builder.append(data);
            builder.append("\n");
        }
        in.close();
        editdata.setText(builder.toString());
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

/**
 * Hàm ghi tập tin trong Android
 * dùng openFileOutput để ghi
```

```
    * openFileOutput tạo ra FileOutputStream
    */
    public void writeData()
    {
        try {
            FileOutputStream out= openFileOutput("myfile.txt",0);
            OutputStreamWriter writer= new
OutputStreamWriter(out);
            writer.write(editdata.getText().toString());
            writer.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    /*
    * Đọc file Resource File
    */
    public void readData2() {
        String data;
        InputStream in =
getResources().openRawResource(R.raw.myfile);
        InputStreamReader inreader = new InputStreamReader(in);
        BufferedReader bufreader = new BufferedReader(inreader);
        StringBuilder builder = new StringBuilder();
        if (in != null) {
            try {
                while ((data = bufreader.readLine()) != null) {
                    builder.append(data);
                    builder.append("\n");
                }
                in.close();
                editdata.setText(editdata.getText()
+builder.toString());
            } catch (IOException ex) {
                Log.e("ERROR", ex.getMessage());
            }
        }
    }
}
```

- **Nội dung phương thức ReadData2():**

`InputStream in= getResources().openRawResource(R.raw.myfile);` Lệnh này đọc Resource File và trả về `InputStream`. Khi có `InputStream` rồi thì mọi xử lý theo cách thông thường theo quy tắc của Java.

- **Nội dung phương thức ReadData():**

`FileInputStream in= openFileInput("myfile.txt");` Lệnh ở dòng này mở tập tin để đọc, Lệnh trả về `FileInputStream` và các xử lý tiếp theo như Java đã học.

- **Nội dung phương thức WriteData():**

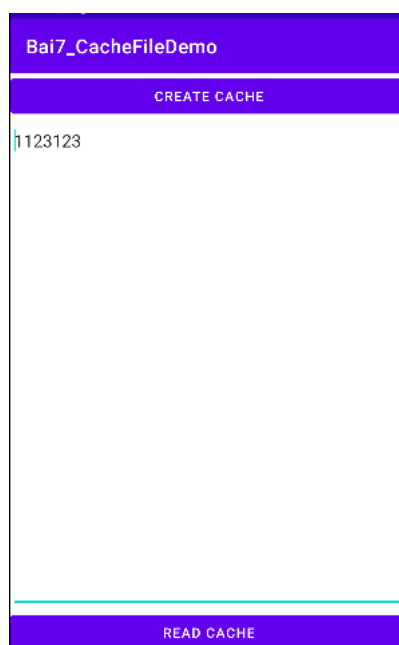
`FileOutputStream out=openFileOutput("myfile.txt",0);` Lệnh này mở tập tin lên để ghi. Đối số thứ 2 nhận giá trị = 0 (**MODE_PRIVATE**) là giá trị mặc định.

Các mode khác sinh viên thử nghiệm tiếp như: **MODE_APPEND**, **MODE_WORLD_READABLE**, **MODE_WORLD_WRITEABLE**.

3. Thực hành lưu trữ dữ liệu sử dụng bộ nhớ cache

Vị trí lưu trữ file cache. **data/data/<gói thư mục ứng dụng>/cache/<tên file>.**

Minh họa hình sau: **Cho giao diện: ứng dụng gồm 2 nút lệnh đọc, tạo cache file. 1 EditText nhập thông tin.**

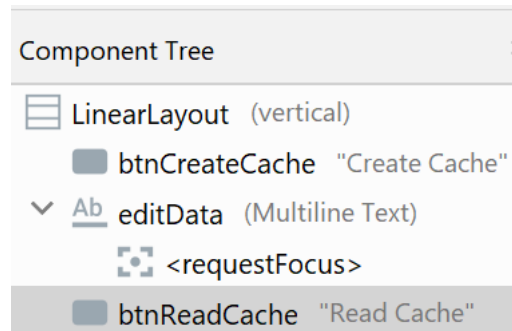


Nút lệnh Create cache: Lưu thông tin trong EditText vào bộ nhớ cache.

Nút lệnh Read cache: đọc thông tin từ bộ nhớ cache gán hiển thị lại EditText

Hướng dẫn đọc ghi file với bộ nhớ cache:

1. Giả sử có giao diện có các điều khiển đặt tên theo mẫu sau: Trong đó nút lệnh ghi file – btnCreateCache; nút lệnh đọc btnReadCache và EditText; editData.



Các xử lý trong MainActivity:

```
public class MainActivity extends AppCompatActivity
implements View.OnClickListener {

    Button btnCreateCache, btnReadCache;
    EditText editData;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnCreateCache=(Button)
findViewById(R.id.btnCreateCache);
        btnReadCache=(Button)
findViewById(R.id.btnReadCache);
        editData =(EditText) findViewById(R.id.editData );
        btnCreateCache.setOnClickListener(this);
        btnReadCache.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        if(v.getId()==R.id.btnCreateCache)
        {
            createCache();
        }
        else if(v.getId()==R.id.btnReadCache)
        {
            readCache();
        }
    }
    /**
     * Lấy toàn bộ file cache
     */
    public void loadAllCache()
    {
```



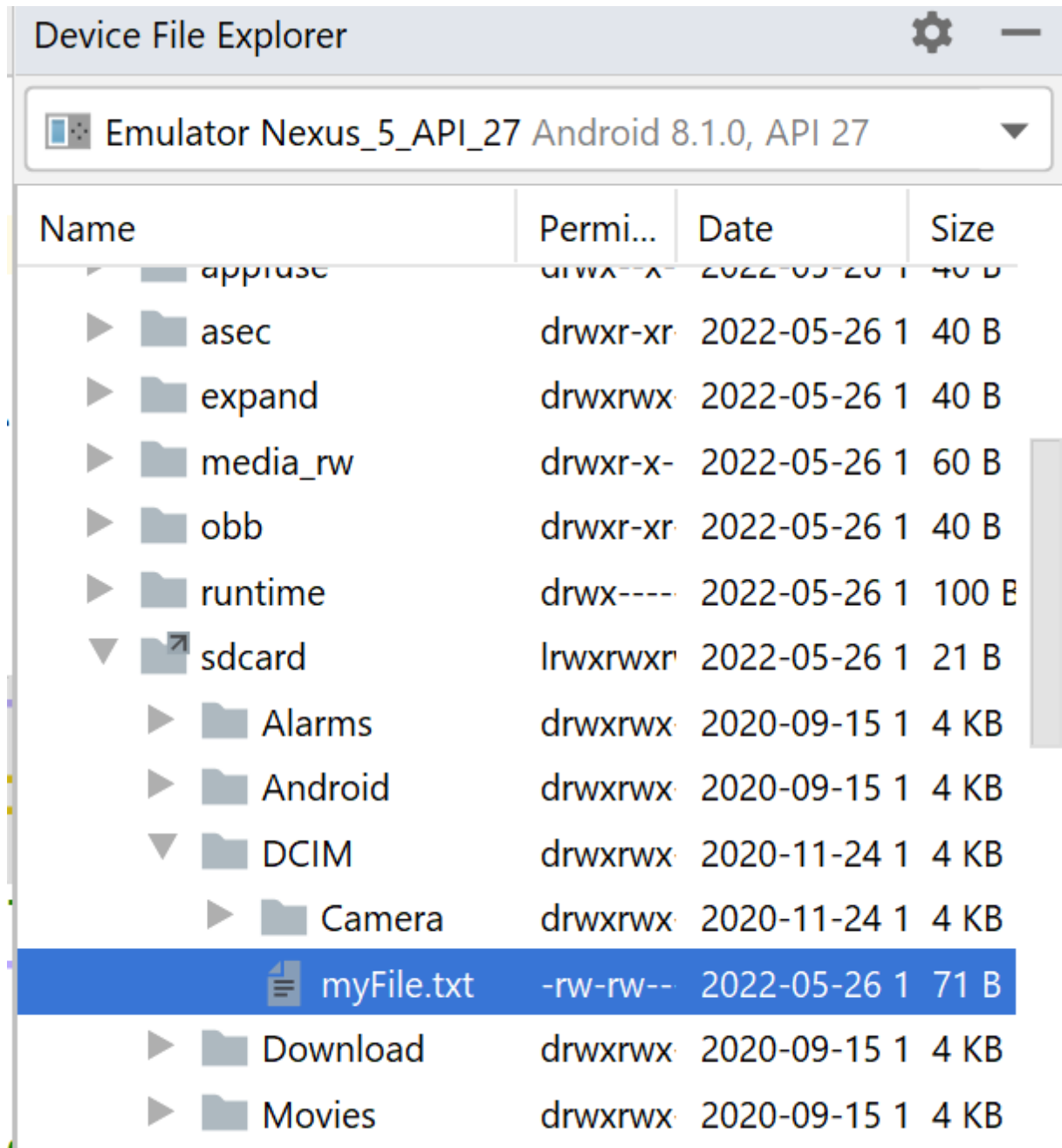
```
File pathCacheDir = getCacheDir();
File []listCache= pathCacheDir.listFiles();
for(File f :listCache)
{
    //process f here
    f.delete();
}
}
/**
 * đọc cache file
 * getCacheDir() trả về đúng thư mục cache
 */
public void readCache() {
    try {
        File pathCacheDir = getCacheDir();
        String strCacheFileName = "myCacheFile.cache";
        File newCacheFile = new
            File(pathCacheDir, strCacheFileName);
        Scanner sc=new Scanner(newCacheFile);
        String data="";
        while(sc.hasNext())
        {
            data+=sc.nextLine()+"\n";
        }
        editData.setText(data);
        sc.close();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
}
/**
 * Lưu cache file
 */
public void createCache()
{
    try {
        File pathCacheDir = getCacheDir();
        String strCacheFileName = "myCacheFile.cache";
        String strFileContents =
editData.getText()+"";
        File newCacheFile = new
            File(pathCacheDir, strCacheFileName);
        newCacheFile.createNewFile();
        FileOutputStream foCache = new
FileOutputStream(
newCacheFile.getAbsolutePath());
        foCache.write(strFileContents.getBytes());
        foCache.close();
    }
```

```
} catch (IOException e) {  
    e.printStackTrace();  
}  
  
}  
  
}
```

4. Thực hành lưu trữ dữ liệu với thẻ nhớ - External storage

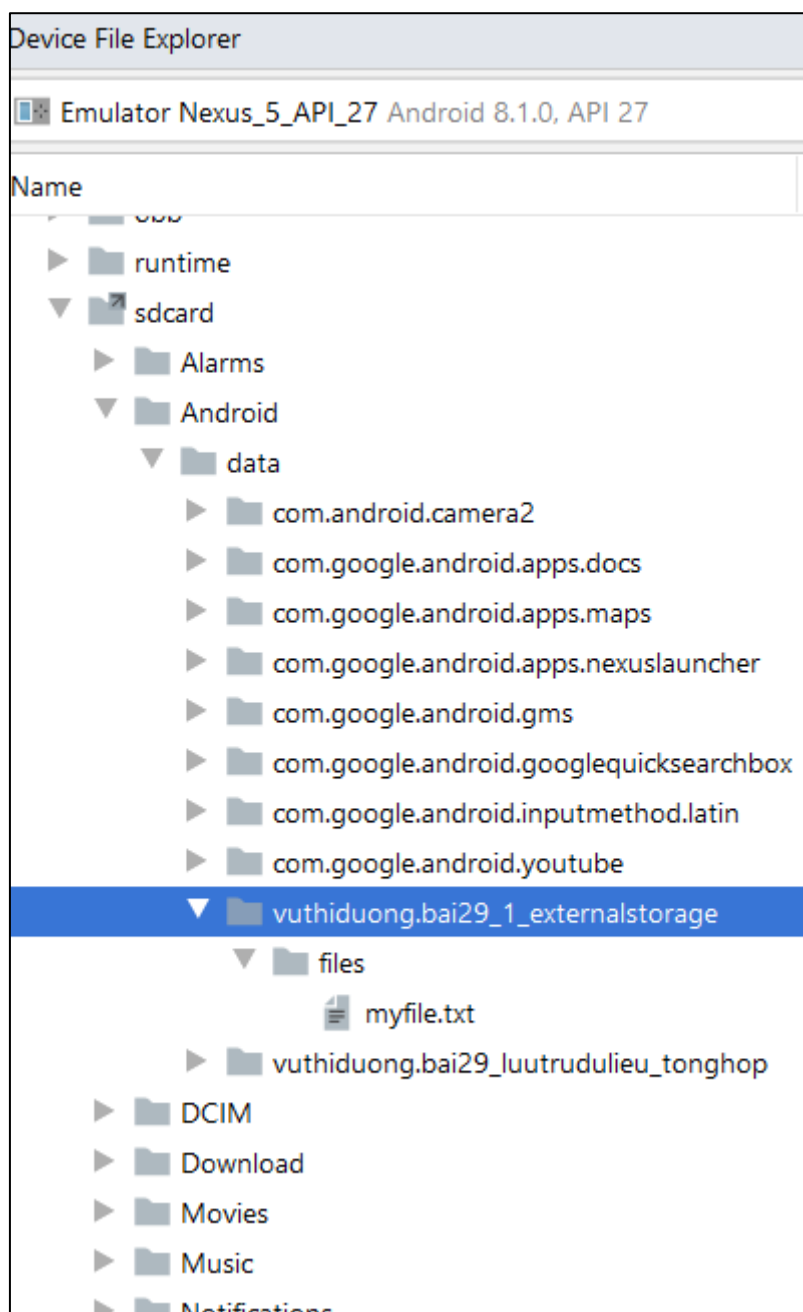
API level <19. File ứng dụng lưu vào thư mục sau của điện thoại

Sdcard/DCIM/<tên file>

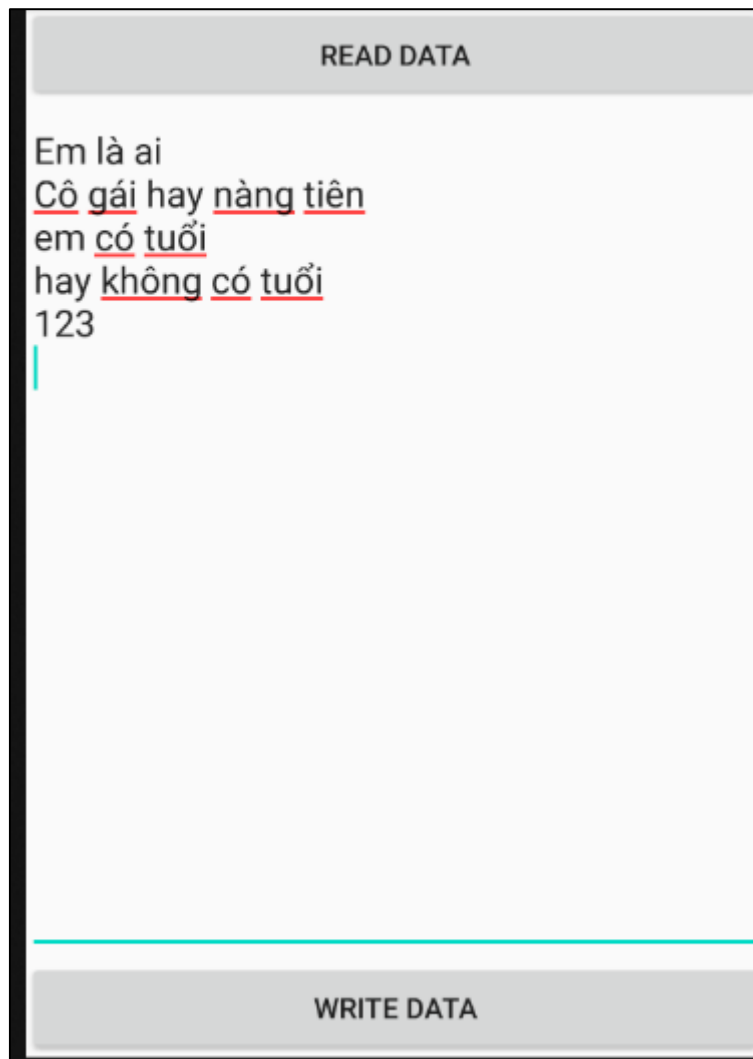


Với API level 29+, file lưu trữ riêng trong thư mục ứng dụng

Sdcard/Android/data/<package>/files/<tên file>



Thực hiện chương trình có giao diện như phần 2 (đọc nghi trên thiết bị)



AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    package="vuthiduong.bai29_1_externalstorage">
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"
/>
```

```
        <category
android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

</manifest>
```

MainActivity.java

```
public class MainActivity extends AppCompatActivity
    implements OnClickListener{

    Button btnread,btnwrite;
    EditText editdata;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnread=(Button) findViewById(R.id.btnreaddata);
        btnwrite=(Button) findViewById(R.id.btnwritedata);
        editdata=(EditText) findViewById(R.id.editdata);
        btnread.setOnClickListener(this);
        btnwrite.setOnClickListener(this);
    }
    public void onClick(View v) {
        if(v.getId()==R.id.btnreaddata)
        {
            readData();
        }
        else if(v.getId()==R.id.btnwritedata)
        {
            writeData();
        }
    }
    /*
    * đọc từ SD Card
    * Environment.getExternalStorageDirectory().getAbsolutePath()
    * để Lấy đường dẫn trên SD Card
    */
    public void readData() {
        //API 29+
        String
        sdcard=this.getExternalFilesDir(null).getAbsolutePath()+"/myFile.
        txt";
        //Cách 1, sử dụng Scanner đọc file
```

```
try {
    Scanner scan=new Scanner(new File(sdcard));
    String data="";
    while(scan.hasNext())
    {
        data+=scan.nextLine()+"\n";
    }
    scan.close();
    editdata.setText(data+"");
    Toast.makeText(this, "đọc file ok",
    Toast.LENGTH_SHORT).show();

    } catch (IOException e) {
        e.printStackTrace();
        Toast.makeText(this, "lỗi đọc file:",
        Toast.LENGTH_SHORT).show();
        Log.e("DocFile",e.toString());
    }
}
/**
 * ghi tập tin trên SD Card
 */
public void writeData(){
    //API 29+
    String sdcard=this.getExternalFilesDir(null).
        getAbsolutePath()+"/myfile.txt";
    try {
        OutputStreamWriter writer= new
            OutputStreamWriter(
                new FileOutputStream(sdcard));
        writer.write(editdata.getText()+"");//dữ liệu cần ghi
        writer.close();
        Toast.makeText(this, "ghi file ok",
        Toast.LENGTH_SHORT).show();
    } catch (IOException e) {
        Toast.makeText(this, "lỗi ghi file",
        Toast.LENGTH_SHORT).show();
        Log.e("GhiFile",e.toString());
    }
}
}
```