

项目实施方案 九职便笺

(一) 读写文件模块



=====文件读写工具类 FileIO=====

```
package com.jvtc.ad.ys0202;
import android.content.Context;
import android.widget.Toast;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
public class FileIO {
    /**
     * 方法功能: 写入文件
     * @param context: 上下文对象
     * @param fileName: 文件名
     * @param fileContent: 文件内容
     */
    public static void write(Context context, String fileName, String fileContent) {
        FileOutputStream fos=null; //创建输出流对象
        try{ //尝试执行 try 模块
            fos=context.openFileOutput(fileName, Context.MODE_PRIVATE);
```

//打开文件，构建输出流通道若无文件 *fileName*，则新建文件；若已有文件 *fileName*，则打开文件，之后写入文件的内容将会覆盖掉原有内容

```
        fos.write(fileContent.getBytes());    //转换成字节数组，写入文件
        Toast.makeText(context, "文件保存成功", Toast.LENGTH_SHORT).show();
    } catch (Exception e) {    //执行 try 模块时遇到异常跳入 catch 模块继续执行
        Toast.makeText(context, "文件保存失败", Toast.LENGTH_SHORT).show();
    } finally {    //不论执行完 try 模块还是 catch 模块之后，均要执行 finally 模块
        if(fos!=null) {
            try{
                fos.flush();    //清除缓存
                fos.close();    //关闭文件
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

/**

* 方法功能：读取文件

* @param context: 上下文对象

* @param fileName: 文件名

* @return : 文件内容

*/

```
public static String read(Context context,String fileName){
    FileInputStream fis=null;
    String fileContent=null;
    try{    //尝试执行 try 模块
        fis=context.openFileInput(fileName);    //打开文件，构建输出流通道
        //若无文件 fileName，则新建文件；若已有文件 fileName，则打开文件，之后
        写入文件的内容将会覆盖掉原有内容
        byte[] buffer=new byte[fis.available()];
        fis.read(buffer);    //转换成字节数组，写入文件
        fileContent=new String(buffer);
    } catch (Exception e) {    //执行 try 模块时遇到异常跳入 catch 模块继续执行
        Toast.makeText(context, "文件读取失败", Toast.LENGTH_SHORT).show();
    } finally {    //不论执行完 try 模块还是 catch 模块之后，均要执
        行 finally 模块
        if(fis!=null) {
            try{
                fis.close();    //关闭文件
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```

    }
    return fileContent;
}
}

```

=====读取文件界面处理 ReadActivity.class=====

```

package com.jvtc.ad.y0202;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
public class ReadActivity extends AppCompatActivity {
    private EditText readName;           //文件名编辑框
    private TextView readContent;        //文件内容文本框
    private Button read;                 //读取文件按钮
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_read);
        //-----获取布局组件-----

        readName=(EditText)findViewById(R.id.read_name);
        readContent=(TextView) findViewById(R.id.read_content);
        read=(Button)findViewById(R.id.read);
        //-----添加事件监听-----
        read.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String fileName=readName.getText().toString().trim()+".txt";
                //获取用户输入的文件名
                String fileContent=FileIO.read(ReadActivity.this,fileName);
                readContent.setText(fileContent);
            }
        });
    }
}

```

=====写入文件界面处理 WriteActivity.class=====

```

package com.jvtc.ad.y0202;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

```

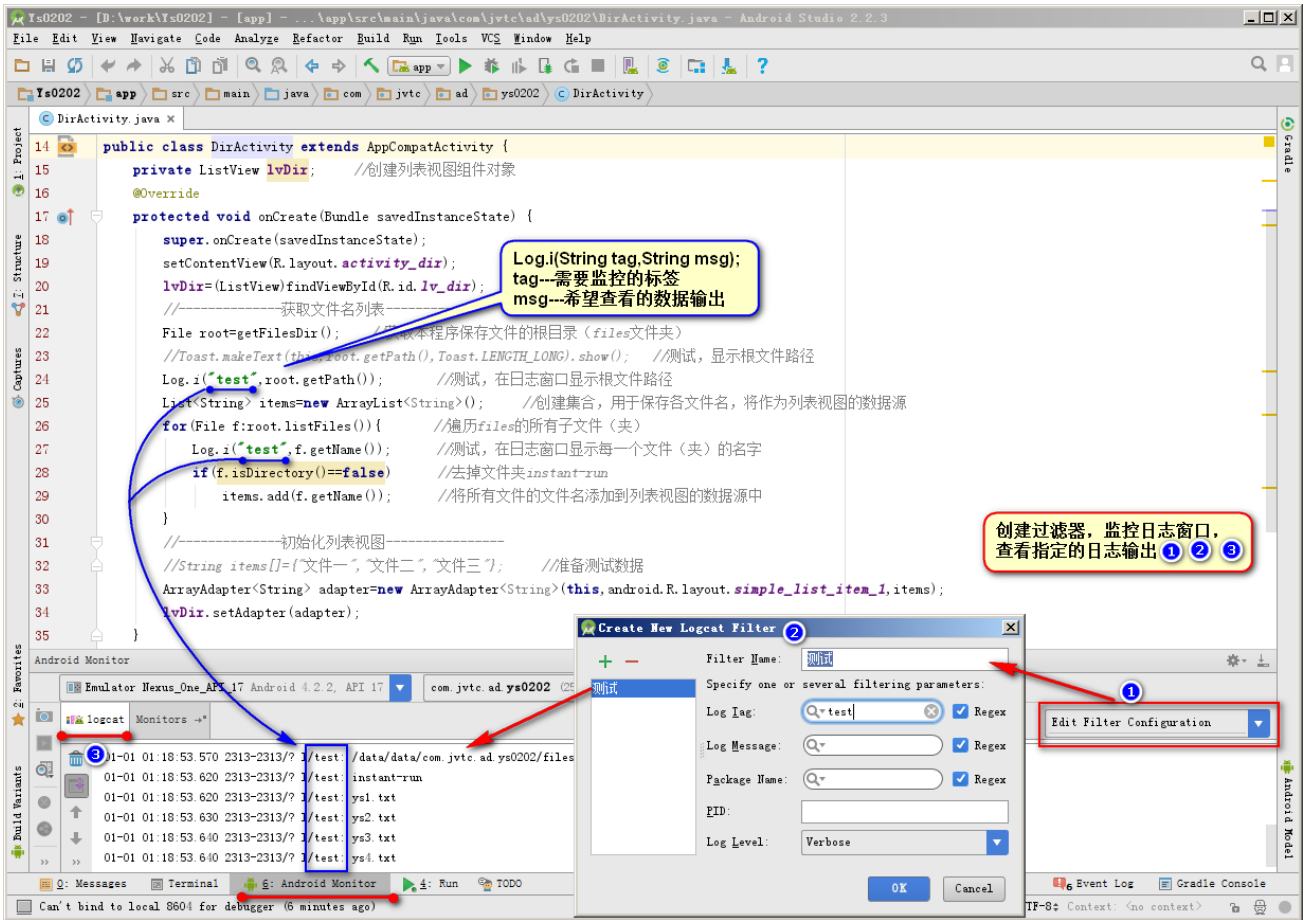
```

import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class WriteActivity extends AppCompatActivity {
    private EditText writeName,writeContent; // “文件名”、“文件内容” 编辑框组件
    private Button write; // “保存文件” 按钮
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_write);
        //-----获取界面组件-----
        writeName=(EditText)findViewById(R.id.write_name);
        writeContent=(EditText)findViewById(R.id.write_content);
        write=(Button)findViewById(R.id.write);
        //-----添加事件监听-----
        write.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String fileName=writeName.getText().toString().trim()+".txt";
                //从界面获取用户输入的文件名
                String fileContent=writeContent.getText().toString().trim();
                //从界面获取用户输入的文件内容
                FileIO.write(WriteActivity.this,fileName,fileContent);
                Intent it=new Intent(WriteActivity.this,MainActivity.class); //
                //保存文件完毕跳转回主界面
                startActivity(it);
            }
        });
    }
}

```

(二) 浏览文件模块

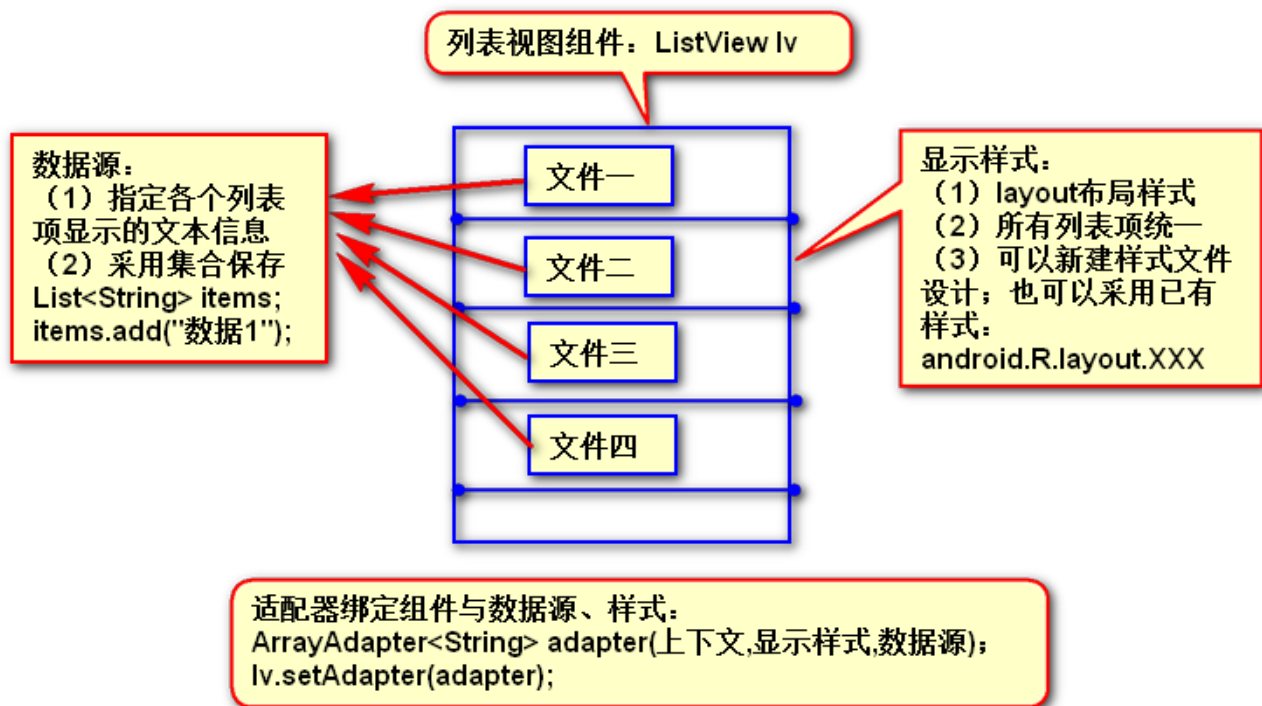


- ① 找到程序创建的那些文件所在的目录——files文件夹
 - (1) 固定的 data/data/项目包/files
 - (2) File root=getFilesDir()
 - ② 遍历files文件夹下的所有子文件：

```
for(File f:root.listFiles())
```

 - ③ 判定每个子文件f，获取所有文件的文件名，加入到集合，作为列表视图的数据源

```
if(f.isDirectory==false)  
    items.add(f);
```



=====获取文件目录界面处理 DirActivity.class=====

```
package com. jvtc. ad. ys0202;
```

```
import android. content. Intent;
import android. support. v7. app. AppCompatActivity;
import android. os. Bundle;
import android. util. Log;
import android. view. View;
import android. widget. AdapterView;
import android. widget. ArrayAdapter;
import android. widget. ListView;
import android. widget. Toast;
import java. io. File;
import java. util. ArrayList;
import java. util. List;
```

```
public class DirActivity extends AppCompatActivity {
    private ListView lv;    //列表视图组件
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super. onCreate(savedInstanceState);
        setContentView(R. layout. activity_dir);
        lv=(ListView)findViewById(R. id. lv);

        //-----获取文件名列表-----
```

```

File root=getFilesDir();    //获取本程序保存文件的根目录 (files 文件夹)
//Toast.makeText(this, root.getPath(), Toast.LENGTH_LONG).show();    //测试,
查看文件夹 root 的路径
Log.i("test",root.getPath());    //日志窗口查看文件夹 root 的路径, 判断是否
成功获取到 files 文件夹
List<String> items=new ArrayList<String>(); //创建集合, 获取所有用户创建文
件的文件名, 将作为列表视图的数据源
for(File f:root.listFiles())    //遍历 files 文件夹下的所有子文件 (夹)
{
    Log.i("test",f.getName());    //日志窗口查看获取到的文件名
    if(f.isDirectory()==false)    //判断当前文件不是文件夹
        items.add(f.getName());    //将文件的文件名添加进数据源集合中
}
//-----初始化列表视图组件-----
/*
for(int i=1;i<=5;i++)    //准备测试数据, 暂时显示在列表视图上
    items.add("文件"+i);
*/
ArrayAdapter<String> adapter=new
ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, items);
//适配器: 参数 1--上下文; 参数 2--列表项统一采用的样式; 参数 3--数据源 (各
列表项显示的内容, 一般是集合对象)
//本例采用已有样式: android.R.layout.XXX
lv.setAdapter(adapter);    //为列表视图组件绑定适配器
//-----添加事件监听-----
lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
        //列表项单击事件处理
        //进入文件内容界面, 同时传递文件名, 读取该文件的详细内容
        //参数 1-parent: 列表视图的适配器 (列表项集), 参数 3-position: 当
前单击列表项的索引号 (0~n-1)
        String fileName=parent.getItemAtPosition(position).toString();
        //用户单击列表项的文本内容 (文件名)
        Log.i("test", "选择文件: "+fileName);
        Intent it=new Intent(DirActivity.this, DirContActivity.class);
        it.putExtra("fName", fileName);    //跳转界面时封装数据
        startActivity(it);
    }
});
}
}
}

```

=====文件内容界面处理 DirContActivity.class=====

```
package com.jvtc.ad.ys0202;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

public class DirContActivity extends AppCompatActivity {
    private TextView dirName, dirContent;    //文件名、文件内容文本框
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dir_cont);
        dirName=(TextView)findViewById(R.id.dir_name);
        dirContent=(TextView)findViewById(R.id.dir_content);
        //-----接收界面跳转时传递的数据-----
        Intent it=getIntent();
        String fileName=it.getStringExtra("fName");
        dirName.setText(fileName);
        //-----根据文件名读取文件-----
        String fileContent=FileIO.read(this, fileName);
        //调用工具类FileIO的读取文件方法，获取文件内容
        dirContent.setText(fileContent);
    }
}
```

=====主界面处理 DirContActivity.class=====

```
package com.jvtc.ad.ys0202;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```



```

/**
 * 方法功能：从主界面跳转到各个子模块界面
 * @param v: 按钮组件
 */
public void toActi(View v) {
    Class c=null;
    switch (v.getId()) {           //获取当前单击按钮的 id
        case R.id. to_write:      //单击“创建文件”按钮时
            c=WriteActivity.class;
            break;
        case R.id. to_read:       //单击“读取文件”按钮时
            c=ReadActivity.class;
            break;
        case R.id. to_dir:        //单击“读取文件目录”按钮时
            c=DirActivity.class;
            break;
        default:
            c=MainActivity.class;
            break;
    }
    Intent it=new Intent(MainActivity.this,c); //执行界面跳转
    startActivity(it);
}
}

```