



# 实施方案 读写指定文件

## 任务描述

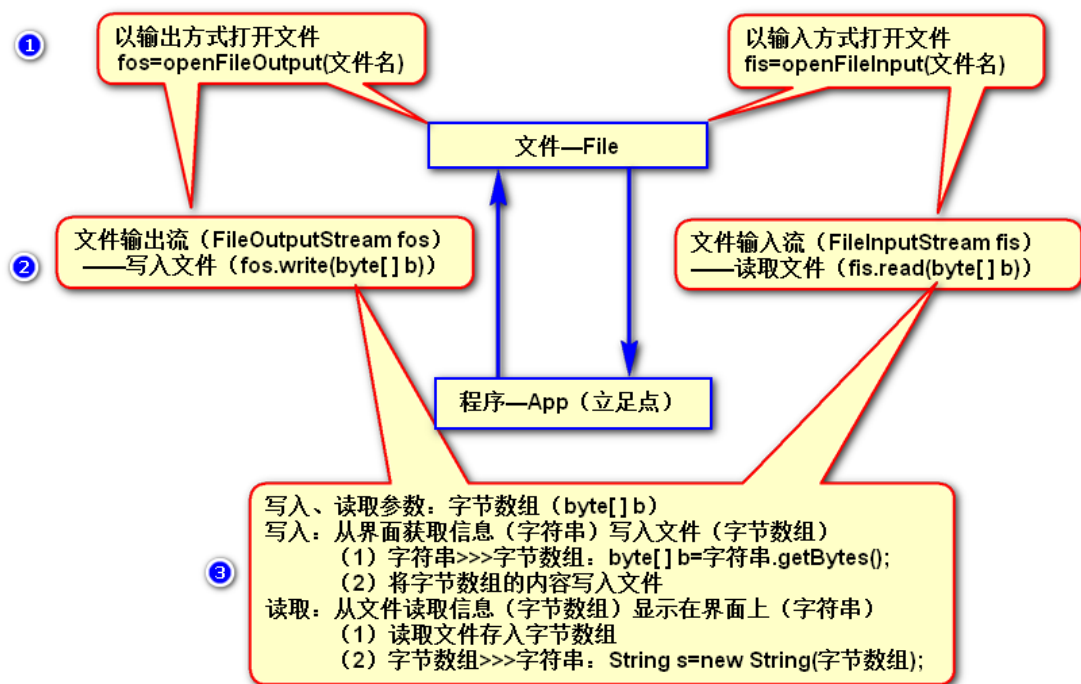
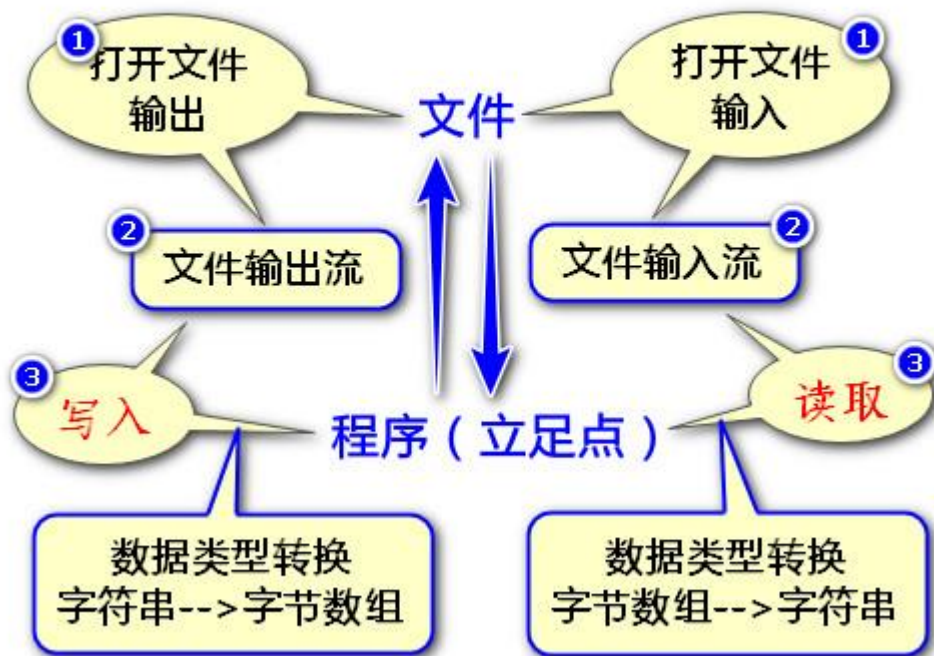
- 1、单击“**写入内容**”按钮将编辑框中的内容写入到指定文件“**test.txt**”中。
- 2、单击“**读取内容**”按钮将指定文件“test.txt”中的内容读取出来显示在屏幕上。





## 技能解析

## 文件读写





## 代码实现

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">

    <EditText
        android:id="@+id/edt_in"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        />

    <Button
        android:id="@+id/btn_write"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="写入文件"/>

    <Button
        android:id="@+id/btn_read"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="读取文件"/>

    <TextView
        android:id="@+id/tv_out"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="显示读取到的内容"
        android:gravity="center"
        android:textSize="20sp"
        android:layout_marginTop="20dp"/>

</LinearLayout>
```

MainActivity.xml:



```
package jvtc.ad.demo06;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.io.FileInputStream;
import java.io.FileOutputStream;

public class MainActivity extends AppCompatActivity implements
View.OnClickListener {
    private EditText edtIn;           //输入编辑框
    private Button btnWrite, btnRead; //写入文件、读取文件按钮
    private TextView tvOut;          //显示文本框
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        initView();
    }

    private void initView() {
        edtIn=findViewById(R.id.edt_in);
        btnWrite=findViewById(R.id.btn_write);
        btnRead=findViewById(R.id.btn_read);
        tvOut=findViewById(R.id.tv_out);
        btnWrite.setOnClickListener(this);
        btnRead.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.btn_write:
                // TODO: 写入文件功能实现
                FileOutputStream fos; //声明文件输出流对象
                try {
                    fos = openFileOutput("test.txt",
```



```

Context.MODE_PRIVATE);
    //参数 1: 文件名 (含扩展名)
    //参数 2: 打开文件的模式。MODE_APPEND--追加;
MODE_PRIVATE--私有 (覆盖)
    String stIn = edtIn.getText().toString().trim();
    //获取界面上的用户输入
    byte byIn[] = stIn.getBytes();
    //类型转换: 字符串-->字节数组
    fos.write(byIn);
    //写入
    fos.close();    //关闭文件输出流
    Toast.makeText(this, "文件写入成功",
Toast.LENGTH_SHORT).show();
    } catch (Exception e) {
        Toast.makeText(this, "文件写入失败",
Toast.LENGTH_SHORT).show();
    }
    break;
case R.id.btn_read:
    // TODO: 读取文件功能实现
    try {
        FileInputStream fis;    //声明输入流对象
        fis=openFileInput("test.txt");
        byte[] buffer=new byte[fis.available()];
        //fis.available(): 返回文件的大小 (字节数)
        fis.read(buffer);
        fis.close();
        String stOut=new String(buffer);
        tvOut.setText(stOut);
        Toast.makeText(this, "文件读取成功",
Toast.LENGTH_SHORT).show();
    } catch (Exception e) {
        Toast.makeText(this, "文件读取失败",
Toast.LENGTH_SHORT).show();
    }
    }
}
}
}

```



## 注意事项

Device File Explorer

Emulator emulator-5554 Android 8.1.0, API 27

菜单: View --> Tool Windows --> Device File Explorer

选择模拟器

写入的文件在模拟器中的存放位置 (8.1版本为例)

项目包路径

写入文件成功后才会生成该文件夹, 写入的文件就在这个文件夹里

Name	Permissions	Date	Size
acct	dr-xr-xr-x	2020-03-30 11:38	0 B
cache	drwxrwx---	2020-03-30 11:38	4 KB
config	drwxr-xr-x	2020-03-30 11:38	0 B
d	lrwxrwxrwx	1970-01-01 00:00	17 B
data	drwxrwx--x	2020-03-30 11:38	4 KB
adb	drwx-----	2020-03-30 11:38	4 KB
anr	drwxrwxr-x	2020-03-30 11:38	4 KB
app	drwxrwx--x	2020-04-27 03:41	4 KB
app-asec	drwx-----	2020-03-30 11:38	4 KB
app-ephemeral	drwxrwx--x	2020-03-30 11:38	4 KB
system	drwxrwxr-x	2020-04-27 03:41	4 KB
system_ce	drwxrwx---	2020-03-30 11:39	4 KB
system_de	drwxrwx---	2020-03-30 11:38	4 KB
tombstones	drwxrwx---	2020-03-30 11:38	4 KB
user	drwxrwx--x	2020-03-30 11:38	4 KB
0	drwxrwx---	2020-03-30 11:38	10 B
android	drwx-----	2020-03-30 11:39	4 KB
android.ext.services	drwx-----	2020-03-30 11:39	4 KB
android.ext.shared	drwx-----	2020-03-30 11:39	4 KB
com.android.backupconfirm	drwx-----	2020-03-30 11:39	4 KB
com.android.bips	drwx-----	2020-03-30 11:39	4 KB
com.android.webview	drwx-----	2020-03-30 11:39	4 KB
com.svox.pico	drwx-----	2020-03-30 11:39	4 KB
jp.co.omronsoft.openwnn	drwx-----	2020-03-30 11:39	4 KB
jvtc.ad.demo03	drwx-----	2020-03-30 11:40	4 KB
jvtc.ad.demo04	drwx-----	2020-04-08 03:32	4 KB
jvtc.ad.demo05	drwx-----	2020-04-17 12:38	4 KB
cache	drwxrws--x	2020-04-17 12:38	4 KB
code_cache	drwxrws--x	2020-04-17 12:38	4 KB
shared_prefs	drwxrws--x	2020-04-17 13:05	4 KB
jvtc.ad.demo06	drwxrws--x	2020-04-27 03:41	4 KB
cache	drwxrws--x	2020-04-27 03:35	4 KB
code_cache	drwxrws--x	2020-04-27 03:41	4 KB
files	drwxrwx--x	2020-04-27 03:36	4 KB
org.chromium.webview_shell	drwx-----	2020-03-30 11:39	4 KB
user_de	drwx--x--x	2020-03-30 11:38	4 KB
var	drwxr-xr-x	2020-03-30 11:38	4 KB
vendor	drwxrwx--x	2020-03-30 11:38	4 KB
dev	drwxr-xr-x	2020-03-30 11:38	1.3 KB
etc	lrwxrwxrwx	1970-01-01 00:00	11 B
mnt	drwxr-xr-x	2020-03-30 11:38	220 B
oem	drwxr-xr-x	1970-01-01 00:00	40 B
proc	dr-xr-xr-x	2020-03-30 11:38	0 B
root	drwx-----	2018-01-25 02:43	40 B
sbin	drwxr-x---	1970-01-01 00:00	100 B
sdcard	lrwxrwxrwx	1970-01-01 00:00	21 B
storage	drwxr-xr-x	2020-03-30 11:39	80 B