

浙江工业大学

移动应用开发实验：计算器实验

2023/2024(2)



学生姓名	李飞飞
学生学号	202105710309
学生班级	软工 2206
任课教师	邱杰凡
实验时间	2024.5.15

计算器实验

一、 列出主要用到的组件和控件：

1. MainActivity (Activity 组件)

a) 一堆样例代码给的控件，这里就不列出了

b) `Double nowAns` 字段

实际上是作为消息队列使用的，但是由于一次只会有一个

数据被计算，所以只用一个 `Double` 就可以解决了

`LoginActivity` (Activity 组件)

c) `editTextUsername` `EditText` 控件
用于输入账号

d) `editTextPassword` `EditText` 控件
用于输入密码

e) `buttonLogin` `Button` 控件
用于确认登录

2. AddService(Service 组件)

```
public int onStartCommand(Intent intent, int flags, int  
startId)
```

此方法通过 `Intent` 接收数据并将处理后的结果(计算结果，
`PID,TID`)打包进 `Bundle` 返回给 `MainActivity`，下面几个
`Service` 机制相同

3. SubService(Service 组件)

类似 `AddService`

4. MulService(Service 组件)

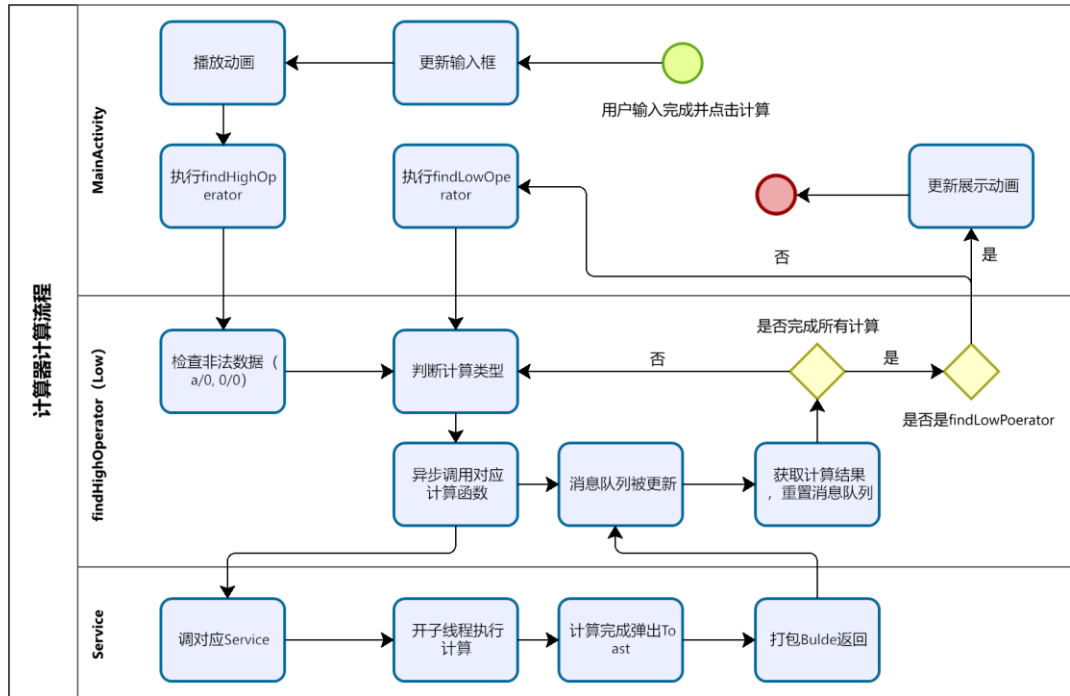
类似 `AddService`

5. DivService(Service 组件)

类似 `AddService`

二、 请写出为了实现主要逻辑，需要使用的控件之间的逻辑。

泳道图:



*这里只说新增代码，不解释实例给出的部分

用户输入完成字符串并点击“=”后会执行 `findHighOperator`，将乘除运算执行掉，同时处理非法情况（`a/0,0/0`）再执行 `findLowOperator` 将加减运算执行掉。

其中每次计算会调用对应 `add/sub/mul/div` 函数，这四个函数为异步函数，通过模拟消息队列 `nowAns` 字段与 `findHighOperator` 和 `findLowOperator` 进行信息传递。当 `nowAns` 有值说明计算已经完成，继续下次计算。全部计算完成后更新显示框。

add/sub/mul/div 函数会将计算数据通过 **Intent** 传给对应 **Service**, **Service** 完成计算后打包 **Bundle** 传回数据，同时 **Service** 通过 **Toast** 抛出此次计算的信息（计算结果，**PIT**, **TID**）。

三、说明任务 1 与任务 2 的主要差异在哪里，可分多点描述？

- 要求必须异步调用 Service,因此需要借助消息队列处理结果。下图

红色部分为生产数据已经消费数据过程，蓝色为恢复队列

```
public int findLowOperator(int index) {
    if (mInputList.size() > 1 && index >= 0 && index < mInputList.size()) {
        for (int i = index; i < mInputList.size(); i++) {
            InputItem item = mInputList.get(i);
            if (getResources().getString(R.string.sub).equals(item.getInput())
                || getResources().getString(R.string.add).equals(item.getInput())) {
                double c, d;
                c = Double.parseDouble(mInputList.get(i - 1).getInput());
                d = Double.parseDouble(mInputList.get(i + 1).getInput());

                if (mInputList.get(i + 1).getType() == InputItem.InputType.INT_TYPE && mInputList.get(i - 1).getType() == InputItem.
                    if (getResources().getString(R.string.add).equals(item.getInput())) {
                        add(c, d);
                        while (nowAns == null) {};
                        mInputList.set(i - 1, new InputItem(String.valueOf(nowAns.intValue()), InputType.INT_TYPE));
                    } else {
                        sub(c, d);
                        while (nowAns == null) {};
                        mInputList.set(i - 1, new InputItem(String.valueOf(nowAns.intValue()), InputType.INT_TYPE));
                    }
                } else {
                    if (getResources().getString(R.string.add).equals(item.getInput())) {
                        add(c, d);
                        while (nowAns == null) {};
                        mInputList.set(i - 1, new InputItem(String.valueOf(nowAns), InputType.DOUBLE_TYPE));
                    } else {
                        sub(c, d);
                        while (nowAns == null) {};
                        mInputList.set(i - 1, new InputItem(String.valueOf(nowAns), InputType.DOUBLE_TYPE));
                    }
                }
                nowAns = null;
                mInputList.remove(index: i + 1);
                mInputList.remove(i);
                return findLowOperator(i);
            }
        }
    }
}
```

- 要求在进入 Service 抛出 Toast,所以 Toast 要与 Lopper 绑定。下图

红色为绑定，蓝色为抛出 Toast 部分

```

public void onCreate() {
    super.onCreate();
    handler = new Handler(Looper.getMainLooper());
}

@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    double a = intent.getDoubleExtra(EXTRA_A, 0);
    double b = intent.getDoubleExtra(EXTRA_B, 0);
    ResultReceiver receiver = intent.getParcelableExtra(EXTRA_RESULT_RECEIVER);

    new Thread() -> {
        double result = addBigNumbers(a, b);
        int tid = android.os.Process.myTid();
        int pid = android.os.Process.myPid();

        if (receiver != null) {
            Bundle bundle = new Bundle();
            bundle.putDouble("result", result);
            bundle.putInt("TID", tid);
            bundle.putInt("PID", pid);
            showToast("结果: " + result + ", TID: " + tid + ", PID: " + pid);
            receiver.send(0, bundle);
        }

        stopSelf(startId);
    }.start();

    return START_NOT_STICKY;
}

```

一个用法

```

private void showToast(final String message) {
    handler.post(new Runnable() {
        @Override
        public void run() {
            Toast.makeText(getApplicationContext(), message, Toast.LENGTH_SHORT).show();
        }
    });
}

```