程序设计实训(Java)

个人报告

题　　目：　基于 ANDROID 开发的计算器

学生姓名：

学　　院：

班　　级：

2022 年 6 月 24 日

目录

一、 课设角色

本人在项目中个人独立完成计算器 Android 客户端和服务器（用于计算器客户端更新用途）开发

二、需求分析

计算器是日常生活中十分便捷有效的工具，能实现加减乘除等简单运算。计算器可以大大的降低运算难度，提高准确率和精确率。而用户在使用是，也希望拥有一个良好的界面，简约美观的效果，能够实现快捷简单的操作，简单的加减乘除。而计算器也需要简洁大方，易于操作，直观明了，突出显示重要及出错信息。另外为了保证用户能够实时使用最新的版本，我在此软件此软件加入了更新机制。

三、采用何种技术实现

本算法通过采用了 Collection 集合类和 Stack 栈类,其中通过栈来分别存储后缀式和运算符。另外本算法的部分计算（比如乘方和开方），使用了 math 类实现的计算。计算式的部分字符处理通过使用 String 类实现了运算符的转义、特殊数值（π）的数值转换。

客户端的检查更新通过 Socket 实现，利用 TCP 通信完成客户端与服务器之间的数据交换。

Android 客户端通过 Android Studio 设计。

四、设计思想

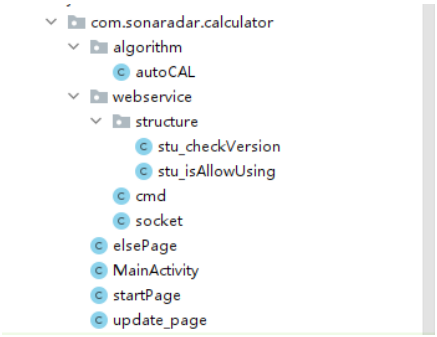
由于表达式由数字和运算符两部分，可以将运算符和数字作为两个整体。数字一般比较基本，不涉及顺序，但是部分数字有负号，需要通过转义字符~实现。而运算符涉及到优先级问题，在本程序中，加、减，乘、除、余，阶乘、乘方、开方，括号分别具有不同的优先级，需要

优先计算运算优先级高的算式。此外，由于日常用户会经常遇到进制转换、三角函数的计算等问题，本算法中提供了相应的计算方案。

五、具体实现

1.Android 客户端

项目结构：



代码：

```
package com.sonaradar.calculator.algorithm;

import java.util.Collections;
import java.util.Stack;

public class autoCAL {
    public static String getResult(String formula){
        return
String.valueOf(conversion(triangle_cal(format_changer(f
ormula))));
    }
    //通过使用栈来实现
    private Stack<String> postfixStack = new
Stack<String>();//存储后缀
    private Stack<Character> opStack = new
Stack<Character>();//存储运算符
    //外部调用这个方法
    public static double conversion(String expression) {
        double result = 0;
        autoCAL cal = new autoCAL();
        try {
            expression = transform(expression);//负号转义
```

```
            result = cal.calculate(expression);//计算结果调
用
        }
        catch (Exception e) {
            //报 NAN 错误，算不出来
            return 0.0 / 0.0;
        }
        return result;
    }

    //负数符号翻转为~，防止和减号弄混了
    private static String transform(String expression) {
        char[] arr = expression.toCharArray();
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] == '-') {
                if (i == 0) {
                    arr[i] = '~';
                } else {
                    char c = arr[i - 1];
                    if (c == '+' || c == '-' || c == '*' || c
== '/' || c == '(' || c == 'E' || c == 'e'||c == '%'|| c
== '^'|| c == '!' || c == '$') {
                        arr[i] = '~';
                    }
                }
            }
        }
        if(arr[0]=='~'||arr[1]=='('){
            arr[0]='-';
            return "0"+new String(arr);
        } else{
            return new String(arr);
        }
    }

    //计算表达式
    private double calculate(String expression) {
        Stack<String> resultStack = new Stack<String>();
        prepare(expression);
        Collections.reverse(postfixStack);
        //将后缀式栈反转
        String firstValue, secondValue, currentValue;
        //参与计算的第一个值，第二个值和算术运算符
        while (!postfixStack.isEmpty()) {
            currentValue = postfixStack.pop();
```

```
            if (!isOperator(currentValue.charAt(0))) {
                //如果不是运算符则存入操作数栈中
                currentValue = currentValue.replace("~",
"-");
                resultStack.push(currentValue);
            } else {
                //如果是运算符则从操作数栈中取两个值和该数值一起参
与运算
                secondValue = resultStack.pop();
                firstValue = resultStack.pop();
                //将负数标记符改为负号
                firstValue = firstValue.replace("~", "-");
                secondValue = secondValue.replace("~", "-");
                String tempResult = calculate(firstValue,
secondValue, currentValue.charAt(0));
                resultStack.push(tempResult);
            }
        }
        return Double.valueOf(resultStack.pop());
    }

    //数据准备阶段将表达式转换成为后缀式栈
    private void prepare(String expression) {
        opStack.push(',');
        // 运算符放入栈底元素逗号，此符号优先级最低
        char[] arr = expression.toCharArray();
        int currentIndex = 0;
        // 当前字符的位置
        int count = 0;
        // 上次算术运算符到本次算术运算符的字符的长度便于或者之间
的数值
        char currentOp, peekOp;
        // 当前操作符和栈顶操作符
        for (int i = 0; i < arr.length; i++) {
            currentOp = arr[i];
            if (isOperator(currentOp)) {
                // 如果当前字符是运算符
                if (count > 0) {
                    postfixStack.push(new String(arr,
currentIndex, count));
                    // 取两个运算符之间的数字
                }
                peekOp = opStack.peek();
                if (currentOp == ')') {
```

```
                    // 遇到反括号则将运算符栈中的元素移除到后缀式
栈中直到遇到左括号
                    while (opStack.peek() != '(') {

postfixStack.push(String.valueOf(opStack.pop()));
                    }
                    opStack.pop();
                } else {
                    while (currentOp != '(' && peekOp != ','
&& compare(currentOp, peekOp)) {

postfixStack.push(String.valueOf(opStack.pop()));
                        peekOp = opStack.peek();
                    }
                    opStack.push(currentOp);
                }
                count = 0;
                currentIndex = i + 1;
            } else {
                count++;
            }
        }
        if (count > 1 || (count == 1
&& !isOperator(arr[currentIndex]))) {
            // 最后一个字符不是括号或者其他运算符的则加入后缀式栈
中
            postfixStack.push(new String(arr, currentIndex,
count));
        }
        while (opStack.peek() != ',') {

postfixStack.push(String.valueOf(opStack.pop()));
            // 将操作符栈中的剩余的元素添加到后缀式栈中
        }
    }

    //运算符号判断
    private Boolean isOperator(char c) {
        return c == '+' || c == '-' || c == '*' || c == '/'
|| c == '(' || c == ')'|| c == '%'|| c == '^'|| c == '!'
|| c == '$';
    }

    //优先级的注册与判断
    private Boolean compare(char cur, char peek) {
```

```
        // 如果是peek优先级高于cur，返回true，默认都是peek优先级要低
        switch (cur){
            case '+':
            case '-':cur=1;break;
            case '*':
            case '/':
            case '%':cur=2;break;
            case '^':
            case '$':
            case '!':cur=3;break;
            case '(':
            case ')':cur=0;break;
        }
        switch (peek){
            case '+':
            case '-':peek=1;break;
            case '*':
            case '/':
            case '%':peek=2;break;
            case '^':
            case '$':
            case '!':peek=3;break;
            case '(':
            case ')':peek=0;break;
        }
        Boolean result = false;
        if (peek >= cur) {
            result = true;
        }
        return result;
    }

    //简单运算处理
    private String calculate(String firstValue, String secondValue, char currentOp) {
        String result = "";
        switch (currentOp) {
            case '+':
                result = String.valueOf(Double.valueOf(firstValue) + Double.valueOf(secondValue));
                break;
            case '-':
                result =
```

```
String.valueOf(Double.valueOf(firstValue) -
Double.valueOf(secondValue));
                break;
            case '*':
                result =
String.valueOf(Double.valueOf(firstValue) *
Double.valueOf(secondValue));
                break;
            case '/':
                result =
String.valueOf(Double.valueOf(firstValue) /
Double.valueOf(secondValue));
                break;
            case '$'://开方
                result =
String.valueOf(Math.pow(Double.valueOf(secondValue),
(double)1/Double.valueOf(firstValue)));
                break;
            case '^':
                result =
String.valueOf(Math.pow(Double.valueOf(firstValue),Doub
le.valueOf(secondValue)));
                break;
            case '!':
                result =
String.valueOf(factorial(Integer.valueOf(firstValue)));
                break;
            case '%':
                result =
String.valueOf(Double.valueOf(firstValue) %
Double.valueOf(secondValue));
                break;
        }
        return result;
    }
    //阶乘计算
    private static long factorial(long number) {
        if (number <= 1)
            return 1;
        else
            return number * factorial(number - 1);
    }
    //对于一些符号的转换
    private static String format_changer(String formula){
        return
```

```
formula.replace("√","$").replace("!","!0").replace("×",
"*").replace("π",String.valueOf(Math.PI));
    }
    //三角函数计算
    private static String triangle_cal(String formula){
        if(formula.contains("sin")){
          return formula.replace("sin(" +
getSubString(formula,"sin(",")") +
")",String.valueOf(Math.sin(Double.valueOf(conversion(g
etSubString(formula,"sin(",")"))))));
        }
        if(formula.contains("cos")){
            return formula.replace("cos(" +
getSubString(formula,"cos(",")") +
")",String.valueOf(Math.cos(Double.valueOf(conversion(g
etSubString(formula,"cos(",")"))))));
        }
        if(formula.contains("tan")){
            return formula.replace("tan(" +
getSubString(formula,"tan(",")") +
")",String.valueOf(Math.tan(Double.valueOf(conversion(g
etSubString(formula,"tan(",")"))))));
        }
        return formula;
    }
    //取中间文本
    public static String getSubString(String text, String
left, String right) {
        String result = "";
        int zLen;
        if (left == null || left.isEmpty()) {
            zLen = 0;
        } else {
            zLen = text.indexOf(left);
            if (zLen > -1) {
                zLen += left.length();
            } else {
                zLen = 0;
            }
        }
        int yLen = text.indexOf(right, zLen);
        if (yLen < 0 || right == null || right.isEmpty()) {
            yLen = text.length();
        }
        result = text.substring(zLen, yLen);
```

```
        return result;
    }
    //转二进制
    public static String toBin(String num){
        try{
            return Integer.toBinaryString((int)
Math.round(Double.valueOf(num)));
        }catch (Exception e){
            return "NaN";
        }
    }
    //转八进制
    public static String toOct(String num){
        try{
            return Integer.toOctalString((int)
Math.round(Double.valueOf(num)));
        }catch (Exception e){
            return "NaN";
        }
    }
    //转 16 进制
    public static String toHex(String num){
        try{
            return Integer.toHexString((int)
Math.round(Double.valueOf(num)));
        }catch (Exception e){
            return "NaN";
        }
    }
}
package com.sonaradar.calculator.webservice.structure;

public class stu_checkVersion {
    //本类是用于存储检查最新版本时保存数据用途
    public int latestVersionCode = 0;//最新的版本代码
    public int requiredOldestVersionCode = 0;//最老允许使用
的版本代码
    public String downloadAddress = "";//新版本下载地址
    public stu_checkVersion(int LVC,int ROVC,String DA){//
这个功能是用来设置数据的，能方便点
        latestVersionCode = LVC;
        requiredOldestVersionCode = ROVC;
        downloadAddress = DA;
    }
}
```

```
package com.sonaradar.calculator.webservice.structure;
//这个类用来存储检查是否允许使用的
public class stu_isAllowUsing {
    public boolean isAllowUsing = false;//是否允许登录
    public String reason = "";//禁止登录的原因
    public stu_isAllowUsing(boolean IAU,String R){//和上一个一样，不讲了
        isAllowUsing = IAU;
        reason = R;
    }
}

package com.sonaradar.calculator.webservice;

import com.sonaradar.calculator.algorithm.autoCAL;
import
com.sonaradar.calculator.webservice.structure.stu_check
Version;
import
com.sonaradar.calculator.webservice.structure.stu_isAll
owUsing;

public class cmd {
    public static int myVersion = 110;//软件现行版本
    public static String da = "";//禁止登录的原因
    //发送检测更新命令，获取到最新版本号、可允许使用的最老版本号和新版本下载地址
    ///[checkversion]

///[checkversion]<latestversion:100><requiredoldestvers
ion:100><downaddress:****>
    public static stu_checkVersion checkVersion(){
        try{

            String str_return =
socket.Sender("[checkversion]");
            String lv =
autoCAL.getSubString(str_return,"<latestversion:",">");
            String rov =
autoCAL.getSubString(str_return,"<requiredoldestversion
:",">");
            String da =
autoCAL.getSubString(str_return,"<downaddress:",">");
            return new
stu_checkVersion(Integer.valueOf(lv),Integer.valueOf(ro
```

```
v),da);
        }catch (Exception e){e.printStackTrace();}
        return new stu_checkVersion(0,0,"");
    }
    //向服务器发送命令，请求是否可以使用软件，返回是否允许使用的布尔值和禁止登录的原因(如果禁止登录的话)
    ///[isallowusing]<version:100>
    ///[isallowusing]<status:accept/deny><reason:******>
    public static stu_isAllowUsing isAllowUsing(int versionCode){
        try{
            String str_return = socket.Sender("[isallowusing]<version:" + versionCode + ">");
            String status = autoCAL.getSubString(str_return,"<status:",">");
            String reason = autoCAL.getSubString(str_return,"<reason:",">");
            if(str_return.contains("accept")){
                return new stu_isAllowUsing(true,reason);
            }else{
                return new stu_isAllowUsing(false,reason);
            }
        } catch (Exception e) {}
        return new stu_isAllowUsing(false,"");
    }
}
package com.sonaradar.calculator.webservice;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.InetAddress;
import java.net.Socket;
import java.net.UnknownHostException;

public class socket {
    private static String serverdomain = "www.thundersoftware.top";//域名
    private static int serverport = 54300;///服务器端口
```

```
    private static int overTime = 5000;///超时设置时间
1s=1000
    //向服务器发送信息，服务器获取到信息本端再接受，返回值是接受到
的信息
    public static String Sender(String message){
        try {
            Socket s = new
Socket(getip(serverdomain),serverport);
            s.setSoTimeout(overTime);
            //构建
            InputStream is = s.getInputStream();
            OutputStream os = s.getOutputStream();
            BufferedWriter bw = new BufferedWriter(new
OutputStreamWriter(os));
            //向服务器端发送一条消息
            bw.write(message);
            System.out.println("[CAL-WS]Sending command to
server:"+message);
            bw.flush();
            //读取服务器返回的消息
            BufferedReader br = new BufferedReader(new
InputStreamReader(is));
            String mess = "";
            while (true) {
                String info = br.readLine();
                if (!info.isEmpty()) {
                    System.out.println("[CAL-WS]Received
command from server:"+info);
                    mess=info;
                    break;
                }
            }
            br.close();
            s.close();
            return mess;
        } catch (UnknownHostException e) {
            e.printStackTrace();
            System.out.println("[CAL-WS]Socket Error
UnknownHostException");
        } catch (IOException e) {
            e.printStackTrace();
            System.out.println("[CAL-WS]Socket Error
IOException");
        }
        return "";
```

```
    }
    ///域名转 ip
    public static String getip(String domain) {
        try {
            return
InetAddress.getByName(domain).getHostAddress();
        } catch (Exception e) {
            return "";
        }
    }
}
package com.sonaradar.calculator;

import androidx.appcompat.app.AppCompatActivity;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.os.StrictMode;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.sonaradar.calculator.webservice.cmd;
import
com.sonaradar.calculator.webservice.structure.stu_check
Version;

public class elsePage extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        //页面初始化代码
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_else_page);
        //安卓 SDK 大于 9 时请求网络严苛模式可能会导致 socket 发不出
去，修改 policy 适用于数据请求量小的应用
        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);
```

```
        }
        //显示当前版本
        TextView txt_version =
(TextView)findViewById(R.id.txt_version);

txt_version.setText(String.valueOf(cmd.myVersion));
        //检测版本按钮
        Button btn_update = (Button)
findViewById(R.id.btn_update);
        btn_update.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                stu_checkVersion sc = cmd.checkVersion();
                cmd.da = sc.downloadAddress;
                //检测到新版本
                if(sc.latestVersionCode>cmd.myVersion){
                    AlertDialog.Builder builder = new
AlertDialog.Builder(elsePage.this);
                    builder.setTitle("检测到新版本!");
                    builder.setMessage("当前版本:" +
cmd.myVersion + "\n" +
                            "最新版本:" + sc.latestVersionCode
+ "\n" +
                            "最低可登录版本:" +
sc.requiredOldestVersionCode + "\n" +
                            "为了保证您的正常使用,请更新软件!");
                    builder.setPositiveButton("更新", new
DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface
dialog, int which) {
                            //跳转到更新页面
                            Intent it = new
Intent(getApplicationContext(), update_page.class);//启动
MainActivity
                            startActivity(it);
                        }
                    });
                    builder.setNeutralButton("取消", new
DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface
dialog, int which) {}
                    });
```

```
                builder.show();
            }else{
                //是最新版本
                AlertDialog.Builder builder = new
AlertDialog.Builder(elsePage.this);
                builder.setTitle("您的软件已是最新版本!");
                builder.setMessage("当前版本:" +
cmd.myVersion + "\n" +
                        "最低可登录版本:" +
sc.requiredOldestVersionCode + "\n" +
                        "最新版本:" + sc.latestVersionCode
+ "\n");
                builder.setPositiveButton("确认", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface
dialog, int which) {}});
                builder.show();
            }
        }
    });

    }
}
package com.sonaradar.calculator;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Button;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import com.sonaradar.calculator.algorithm.autoCAL;

public class MainActivity extends AppCompatActivity {

    public static boolean isGetresult = false;
    public static String formula = "";
    public static String M = "";
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ///控件注册
        Button num_0 =  (Button) findViewById(R.id.btn_0);
        Button num_1 =  (Button) findViewById(R.id.btn_1);
        Button num_2 =  (Button) findViewById(R.id.btn_2);
        Button num_3 =  (Button) findViewById(R.id.btn_3);
        Button num_4 =  (Button) findViewById(R.id.btn_4);
        Button num_5 =  (Button) findViewById(R.id.btn_5);
        Button num_6 =  (Button) findViewById(R.id.btn_6);
        Button num_7 =  (Button) findViewById(R.id.btn_7);
        Button num_8 =  (Button) findViewById(R.id.btn_8);
        Button num_9 =  (Button) findViewById(R.id.btn_9);
        Button char_percentage = (Button)
findViewById(R.id.btn_percentage);
        Button char_dot = (Button)
findViewById(R.id.btn_dot);
        Button char_plus = (Button)
findViewById(R.id.btn_plus);
        Button char_reduce = (Button)
findViewById(R.id.btn_reduce);
        Button char_multply = (Button)
findViewById(R.id.btn_muliply);
        Button char_devide = (Button)
findViewById(R.id.btn_devide);
        Button char_clear = (Button)
findViewById(R.id.btn_clear);
        Button char_remove = (Button)
findViewById(R.id.btn_remove);
        Button char_getresult = (Button)
findViewById(R.id.btn_getresult);
        Button char_sqrt = (Button)
findViewById(R.id.btn_sqrt);
        Button char_square = (Button)
findViewById(R.id.btn_square);
        Button char_sin = (Button)
findViewById(R.id.btn_sin);
        Button char_cos = (Button)
findViewById(R.id.btn_cos);
        Button char_tan = (Button)
findViewById(R.id.btn_tan);
        Button char_bin = (Button)
findViewById(R.id.btn_bin);
        Button char_oct = (Button)
```

```
findViewById(R.id.btn_oct);
        Button char_hex = (Button)
findViewById(R.id.btn_hex);
        Button char_set = (Button)
findViewById(R.id.btn_setm);
        Button char_m = (Button) findViewById(R.id.btn_m);
        Button char_factorial = (Button)
findViewById(R.id.btn_factorial);
        Button char_pi = (Button) findViewById(R.id.btn_pi);
        Button char_setM = (Button)
findViewById(R.id.btn_set);
        Button char_leftbracket = (Button)
findViewById(R.id.btn_leftbracket);
        Button char_rightbracket = (Button)
findViewById(R.id.btn_rightbracket);
        TextView txt_mainbar =
(TextView)findViewById(R.id.txt_mainbars);
        TextView txt_maindescribe =
(TextView)findViewById(R.id.txt_maindescribe);
        TextView txt_subbar =
(TextView)findViewById(R.id.txt_subbar);
        TextView txt_subdescribe =
(TextView)findViewById(R.id.txt_subdescribe);

        ///置低
        ScrollView sv =
(ScrollView)findViewById(R.id.scrollView2);
        Handler handler = new Handler();
        handler.post(new Runnable() {
            @Override
            public void run() {
                sv.fullScroll(ScrollView.FOCUS_DOWN);
            }
        });

        ///数字输入
        num_0.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("0");
            }
        });

        num_1.setOnClickListener(new
```

```
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("1");
            }
        });

        num_2.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("2");
            }
        });

        num_3.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("3");
            }
        });

        num_4.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("4");
            }
        });

        num_5.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("5");
            }
        });

        num_6.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("6");
            }
```

```
        });

        num_7.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("7");
            }
        });

        num_8.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("8");
            }
        });

        num_9.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("9");
            }
        });
        char_percentage.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("%");
            }
        });
        char_dot.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula(".");
            }
        });
        char_plus.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("+");
            }
```

```
        });
        char_reduce.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("-");
            }
        });
        char_multply.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("×");
            }
        });
        char_devide.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("/");
            }
        });
        char_clear.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                formula = "0";
                showFormula();
            }
        });
        char_sin.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("sin(");
            }
        });
        char_cos.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("cos(");
            }
        });
        char_tan.setOnClickListener(new
```

```
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("tan(");
            }
        });
        char_sqrt.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("√");
            }
        });
        char_square.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("^");
            }
        });
        char_factorial.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("!");
            }
        });
        char_pi.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("π");
            }
        });
        char_leftbracket.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                appendFormula("(");
            }
        });
        char_rightbracket.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

```
                appendFormula(")");
            }
        });
        //退格功能，遇到sin cos tan nan时自动全退
        char_remove.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(formula.length()>0){
                    try{

if(formula.substring(formula.length()-4,formula.length(
)).contains("sin(")||

formula.substring(formula.length()-4,formula.length()).
contains("cos(")||

formula.substring(formula.length()-4,formula.length()).
contains("tan(")
                        ){
                            formula =
formula.substring(0,formula.length()-4);
                        }else{
                            if(formula.contains("NaN")){
                                formula = "0";
                            }else{
                                formula =
formula.substring(0,formula.length()-1);
                            }
                        }
                    }catch (Exception e){
                        if(formula.contains("NaN")){
                            formula = "0";
                        }else{
                            formula =
formula.substring(0,formula.length()-1);
                        }
                    }
                }
                showFormula();
            }
        });
        //计算表达式，就是等号的功能，通过编的autocal算法实现
        char_getresult.setOnClickListener(new
View.OnClickListener() {
```

```
            @Override
            public void onClick(View v) {
                formula = autoCAL.getResult(formula);
                showFormula();
            }
        });
        //二进制转换
        char_bin.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                formula =
autoCAL.toBin(autoCAL.getResult(formula+"+0"));
                showFormula();
            }
        });
        //八进制转换
        char_oct.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                formula =
autoCAL.toOct(autoCAL.getResult(formula+"+0"));
                showFormula();
            }
        });
        //十六进制转换
        char_hex.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                formula =
autoCAL.toHex(autoCAL.getResult(formula+"+0"));
                showFormula();
            }
        });
        //计算机记忆功能，把数据记忆起来
        char_setM.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                M = formula;
                Toast.makeText(getApplicationContext(), "保
存数据成功:" + formula, Toast.LENGTH_LONG).show();
            }
```

```
        });
        //记忆输出功能
        char_m.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                formula += M;
                showFormula();
            }
        });
        //更新检测按钮
        char_set.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent it = new
Intent(getApplicationContext(), elsePage.class);//启动
MainActivity
                startActivity(it);
            }
        });


    }
    //显示计算后得到的表达式
    public void showFormula(){
        try{
            TextView txt_mainbar =
(TextView)findViewById(R.id.txt_mainbars);
            TextView txt_subbar =
(TextView)findViewById(R.id.txt_subbar);
            if(formula.length()<=0){
                txt_mainbar.setText("0");
            }else{
                String tmp_result1 = formula;
                try{

if(tmp_result1.substring(tmp_result1.length()-2).equals
(".0")){

txt_mainbar.setText(tmp_result1.substring(0,tmp_result1
.length()-2));
                    }else{
                        txt_mainbar.setText(tmp_result1);
                    }
```

```
                }catch(Exception
e){txt_mainbar.setText(tmp_result1);}
            }
            String tmp_result = autoCAL.getResult(formula);
            try{

if(tmp_result.substring(tmp_result.length()-2).equals("
.0")){

txt_subbar.setText(tmp_result.substring(0,tmp_result.le
ngth()-2));
            }else{
                txt_subbar.setText(tmp_result);
            }
            }catch(Exception
e){txt_subbar.setText(tmp_result);}
        }catch (Exception e){}
    }
    //将数字或者运算符接到表达式后面
    public void appendFormula(String append_str) {
        if(formula.replace("0","") == ""){
            formula = append_str;
        }else{
            formula += append_str;
        }
        showFormula();
    }
}
package com.sonaradar.calculator;

import androidx.appcompat.app.AppCompatActivity;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.widget.Toast;

import com.sonaradar.calculator.webservice.cmd;
import
com.sonaradar.calculator.webservice.structure.stu_check
Version;
import
com.sonaradar.calculator.webservice.structure.stu_isAll
owUsing;
```

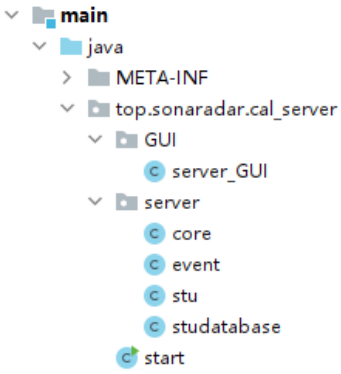
```
//这个页面是启动页，别问，就是用来水的
public class startPage extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_start_page);
        //检测是否允许使用
        stu_isAllowUsing si =
cmd.isAllowUsing(cmd.myVersion);
        if(si.isAllowUsing==true){
            //不能使用执行下面步骤
            AlertDialog.Builder builder = new
AlertDialog.Builder(startPage.this);
            builder.setTitle("服务器拒绝登录");
            builder.setMessage("原因:" + si.reason);
            builder.setPositiveButton("确认", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog,
int which) {
                    System.exit(0);
                }
            });
            builder.setOnDismissListener(new
DialogInterface.OnDismissListener() {
                @Override
                public void onDismiss(final DialogInterface
arg0) {
                    System.exit(0);
                }
            });
            builder.show();
            return;
        }
        //检测可用的最低版本，判断能否使用
        stu_checkVersion sc = cmd.checkVersion();
        cmd.da = sc.downloadAddress;
        if(sc.requiredOldestVersionCode>cmd.myVersion){
            Thread myThread = new Thread() {//创建子线程
                @Override
                public void run() {
                    try {
                        sleep(3000);
                        Intent it = new
```

```
Intent(getApplicationContext(), update_page.class);//启动
MainActivity
                    startActivity(it);
                    finish();
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        };
        myThread.start();
        return;
    }
    //载入到计算器页面
    Thread myThread = new Thread() {//创建子线程
        @Override
        public void run() {
            try {
                sleep(3000);
                Intent it = new
Intent(getApplicationContext(), MainActivity.class);//启
动MainActivity
                startActivity(it);
                finish();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    };
    myThread.start();
  }
}
package com.sonaradar.calculator;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.sonaradar.calculator.webservice.cmd;
//这个页面是用来升级的，很简单的
public class update_page extends AppCompatActivity {
```

```
    public static String downaddress = cmd.da;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_update_page);
        Button qust_btn_update = (Button)
findViewById(R.id.button5);
        Button qust_btn_exitupdate = (Button)
findViewById(R.id.button6);
        qust_btn_update.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                try{
                    Intent intent= new Intent();

intent.setAction("android.intent.action.VIEW");
                    Uri content_url = Uri.parse(downaddress);
                    intent.setData(content_url);
                    startActivity(intent);
                }catch(Exception e){

Toast.makeText(getApplicationContext(),"无法打开网站:" +
downaddress,Toast.LENGTH_LONG).show();
                }
            }
        });
        qust_btn_exitupdate.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                System.exit(0);
            }
        });
    }
}
```

2.服务器端

项目结构：



代码：

```
package top.sonaradar.cal_server.GUI;

import top.sonaradar.cal_server.server.core;
import top.sonaradar.cal_server.server.stu;
import top.sonaradar.cal_server.server.studatabase;
import java.util.Scanner;

//服务器的命令行页面

public class server_GUI {
    //方便输出，不用 system.out.print 了
    private void print(String str){
        System.out.print("[CAL-SERVER]" + str);
    }
    //初始化显示
    public void init(){
        print("-----------------------------------\n");
        print("       CALCULATOR SERVER\n");
        print(" POWERED BY SONARADAR E-TRON INC.\n");
        print("-----------------------------------\n");
        print("You can enter command help for any help.\n");
        studatabase.getData();//读取数据
        core.setAddress(54300);//设置服务器端口
        Thread myThread = new Thread() {//以线程形式启动服务
器的 serversocket
            @Override
            public void run() {
                try {
                    core.Socket_Receiver();
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        };
```

```
            myThread.start();//启动线程
            try{
                Thread.sleep(2000);
            }catch (Exception e){}
            command();//进入到命令识别部分
        }
        //命令识别
        public void command(){
            print("command>");
            Scanner sc = new Scanner(System.in);
            switch (sc.next()){
                case "set latestversion"://设置最新版本的代号
                case "latestversion":
                case "set lv":
                case "lv":setlv();break;
                case "set allowloginversion"://设置最老可使用版本
的代号
                case "allowloginversion":
                case "set alv":
                case "alv":setalv();break;
                case "set denyuserlogin"://是否允许用户使用
                case "denyuserlogin":
                case "set dul":
                case "dul":setdul();break;
                case "set downaddress"://设置新版本下载的网址
                case "downaddress":
                case "set da":
                case "da":setda();break;
                case "help":help();break;//显示帮助
                case "info":info();break;//显示服务器信息
                default: command();return;//未识别到命令
            }
        }
        public void setlv(){//设置最新版本的代号
            print("Enter Latest Version Code>");
            Scanner sc = new Scanner(System.in);
            stu.latestVersionCode=sc.nextInt();
            studatabase.setData();
            command();
            return;
        }
        public void setalv(){//设置最老可使用版本的代号
            print("Enter Allowing Login Version>");
            Scanner sc = new Scanner(System.in);
            stu.allowLoginVersion=sc.nextInt();
```

```
        studatabase.setData();
        command();
        return;
    }
    public void setdul(){//是否允许用户使用
        print("Enter whether deny user login(Y for yes,N for
no)>");
        Scanner sc = new Scanner(System.in);
        switch (sc.next()){
            case "y":
            case "Y":stu.denyAllUser=true;break;
            case "n":
            case "N":stu.denyAllUser=false;break;
        }
        studatabase.setData();
        command();
        return;
    }
    public void setda(){//设置新版本下载的网址
        print("Enter Download Address>");
        Scanner sc = new Scanner(System.in);
        stu.downAddress=sc.next();
        studatabase.setData();
        command();
        return;
    }
    public void help(){//显示帮助
        print("Command List:\n");
        print("lv - set latest version code\n");
        print("alv - set minium allowing login version
code\n");
        print("dul - set whether allowing user login\n");
        print("da - set download address\n");
        print("info - show config info\n");
        command();
        return;
    }
    public void info(){//显示服务器信息
        print("Config Information:\n");
        print("Latest Version
Code:"+stu.latestVersionCode+"\n");
        print("Allowing Login Version
Code:"+stu.allowLoginVersion+"\n");
        print("Deny All User:"+ (stu.denyAllUser ? "True" :
"False")+"\n");
```

```
            print("Download Address:"+stu.downAddress+"\n");
            command();
            return;
        }
    }
package top.sonaradar.cal_server.server;

import java.io.*;
import java.net.*;

//服务器核心部分

public class core{
    private static int Global_Port = 0;//服务器端口

    public static void setAddress(int myport) {//设置服务器
的端口
            Global_Port = myport;
            System.out.println("[CAL-SERVER]Set address port at:
" + Global_Port);
        }

    public static void Socket_Receiver() {//通过
serversocket 收发信息，收到信息后处理后发送到客户端
            try {
                ServerSocket serverSocket = new
ServerSocket(Global_Port);
                System.out.println("[CAL-SERVER]Receive socket
started!");
                Socket accept = serverSocket.accept();
            while (true) {
                        try{
                            BufferedInputStream
bufferedInputStream = new
BufferedInputStream(accept.getInputStream());
                            if
(bufferedInputStream.available() > 0) {
                                byte[] receive = new byte[1024];

bufferedInputStream.read(receive);

//System.out.println("[CAL-SERVER]Received command:" +
new String(receive));
                                BufferedOutputStream
bufferedOutputStream = new
```

```
BufferedOutputStream(accept.getOutputStream());
                        String response =
event.analysisCommand(new String(receive)) + "\n";

//System.out.println("[CAL-SERVER]Send command:" +
response);

bufferedOutputStream.write(response.getBytes());
                        bufferedOutputStream.flush();
                        accept = serverSocket.accept();
                    }
                }catch (Exception e){
                    e.printStackTrace();
                }
        }
        } catch (Exception k) {
            //System.out.println("[CAL-SERVER]Server
socket error");
        }

    }
    //截取子字符串，autocal 的
    public static String getSubString(String text, String
left, String right) {
        String result = "";
        int zLen;
        if (left == null || left.isEmpty()) {
            zLen = 0;
        } else {
            zLen = text.indexOf(left);
            if (zLen > -1) {
                zLen += left.length();
            } else {
                zLen = 0;
            }
        }
        int yLen = text.indexOf(right, zLen);
        if (yLen < 0 || right == null || right.isEmpty()) {
            yLen = text.length();
        }
        result = text.substring(zLen, yLen);
        return result;
    }
}
```

```
package top.sonaradar.cal_server.server;

//收到相应指令需要做出的反应事件集

public class event {
    public static String analysisCommand(String
str_cmd){//分析需要执行哪个命令
        String cmdName = core.getSubString(str_cmd,"[","]");
        switch (cmdName){
            case "checkversion"://检测新版本命令
                return checkVersion(str_cmd);
            case "isallowusing"://判断是否允许登录
                return isAllowLogin(str_cmd);
            default:break;
        }
        return "[command_notfound]";
    }
    ///样例格式
    private static String example(String str_cmd){
        return "";
    }
    //检测新版本命令
    private static String checkVersion(String str_cmd){
        return
"[checkversion]<latestversion:"+stu.latestVersionCode+"
><requiredoldestversion:"+stu.allowLoginVersion+"><down
address:"+stu.downAddress+">";
    }
    //判断是否允许登录
    private static String isAllowLogin(String str_cmd){
        int versionCode =
Integer.valueOf(core.getSubString(str_cmd,"<version:","
>"));
        if(versionCode<stu.allowLoginVersion){
            return
"[isallowusing]<status:deny><reason:version is too old>";
        }
        if(stu.denyAllUser==true){
            return
"[isallowusing]<status:deny><reason:server deny>";
        }
        return "[isallowusing]<status:accept><reason:>";
    }
}
```

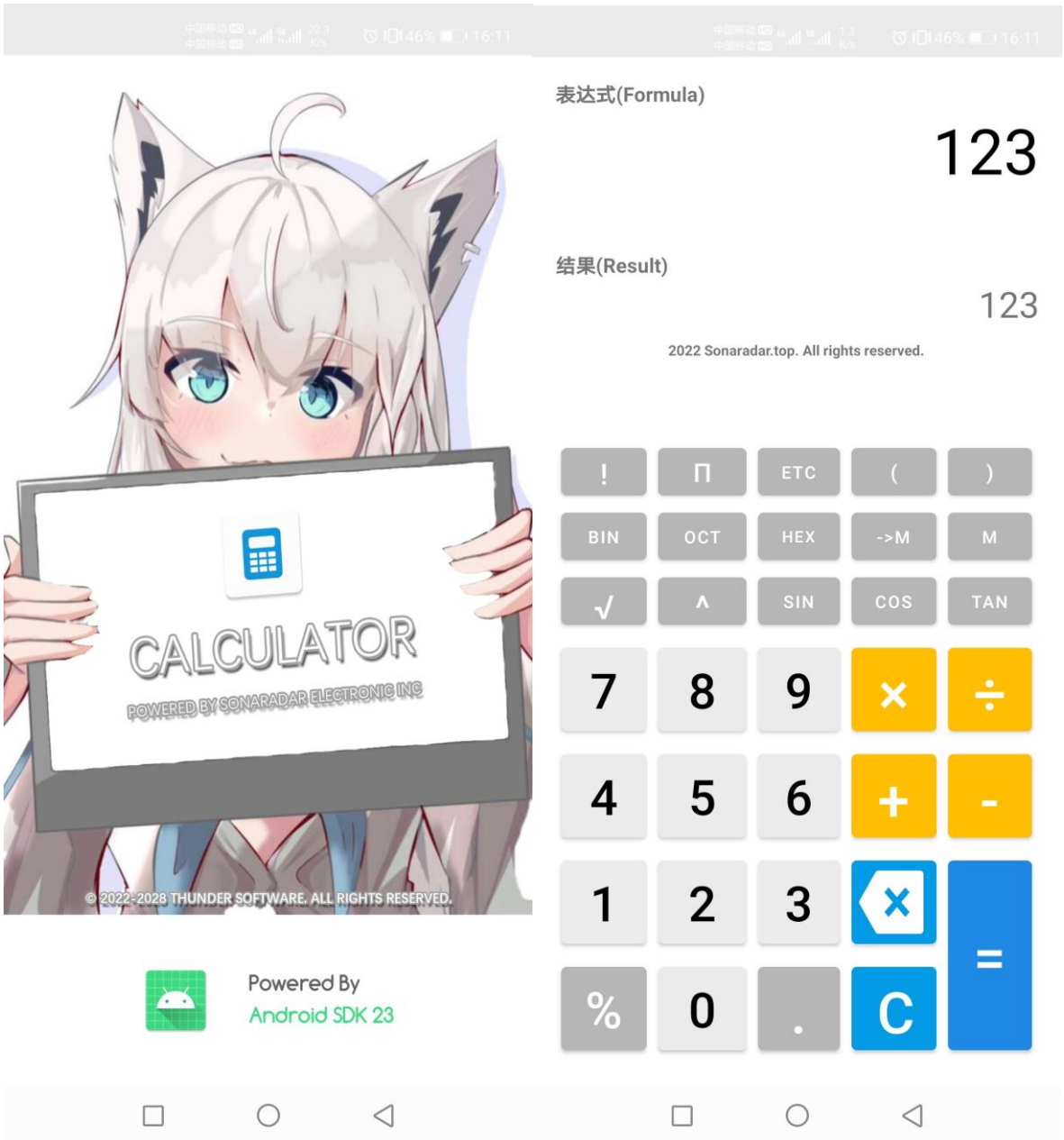
```
package top.sonaradar.cal_server.server;

//这个用来存储一些数据

public class stu {
    public static int latestVersionCode = 0;//最新版本代码
    public static int allowLoginVersion = 0;//最低可登录版本号
    public static boolean denyAllUser = false;//是否允许用户登录
    public static String downAddress = "";//新版本下载网址
    public static void setStu(int lvc,int alv,boolean adu,String da){//设置这些数据
        latestVersionCode = lvc;
        allowLoginVersion = alv;
        denyAllUser=adu;
        downAddress = da;
    }
}
package top.sonaradar.cal_server.server;

import java.io.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Properties;

public class studatabase {
        public static void setData(){//存储 stu 中的所有数据
            try {
                int lvc=stu.latestVersionCode;
                int alv=stu.allowLoginVersion;
                int adu= stu.denyAllUser ? 1 : 0;
                String da = stu.downAddress;
                writeTxt(System.getProperty("user.dir") + "/" + "latestVersion.txt",String.valueOf(lvc));
                writeTxt(System.getProperty("user.dir") + "/" + "allowLoginVersion.txt",String.valueOf(alv));
                writeTxt(System.getProperty("user.dir") + "/" + "denyAllUser.txt",String.valueOf(adu));
                writeTxt(System.getProperty("user.dir") + "/" + "downAddress.txt",da);
            } catch (Exception e) {}
        }
```

```
        public static void getData(){//保存 stu 中的所有数据
            try {
                String latestVersion =
readTxt(System.getProperty("user.dir") + "/" +
"latestVersion.txt");
                String allowLoginVersion =
readTxt(System.getProperty("user.dir") + "/" +
"allowLoginVersion.txt");
                String denyAllUser =
readTxt(System.getProperty("user.dir") + "/" +
"denyAllUser.txt");
                String downAddress =
readTxt(System.getProperty("user.dir") + "/" +
"downAddress.txt");

stu.setStu(Integer.valueOf(latestVersion),Integer.value
Of(allowLoginVersion),Integer.valueOf(denyAllUser)==1,d
ownAddress);
            } catch (Exception e) {}
        }
    private static String readTxt(String txtPath) {//读取
txt 中的文本
        File file = new File(txtPath);
        if(file.isFile() && file.exists()){
            try {
                FileInputStream fileInputStream = new
FileInputStream(file);
                InputStreamReader inputStreamReader = new
InputStreamReader(fileInputStream);
                BufferedReader bufferedReader = new
BufferedReader(inputStreamReader);
                StringBuffer sb = new StringBuffer();
                String text = null;
                while((text = bufferedReader.readLine()) !=
null){
                    sb.append(text);
                }
                return sb.toString();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        return null;
    }
    private static void writeTxt(String txtPath,String
```

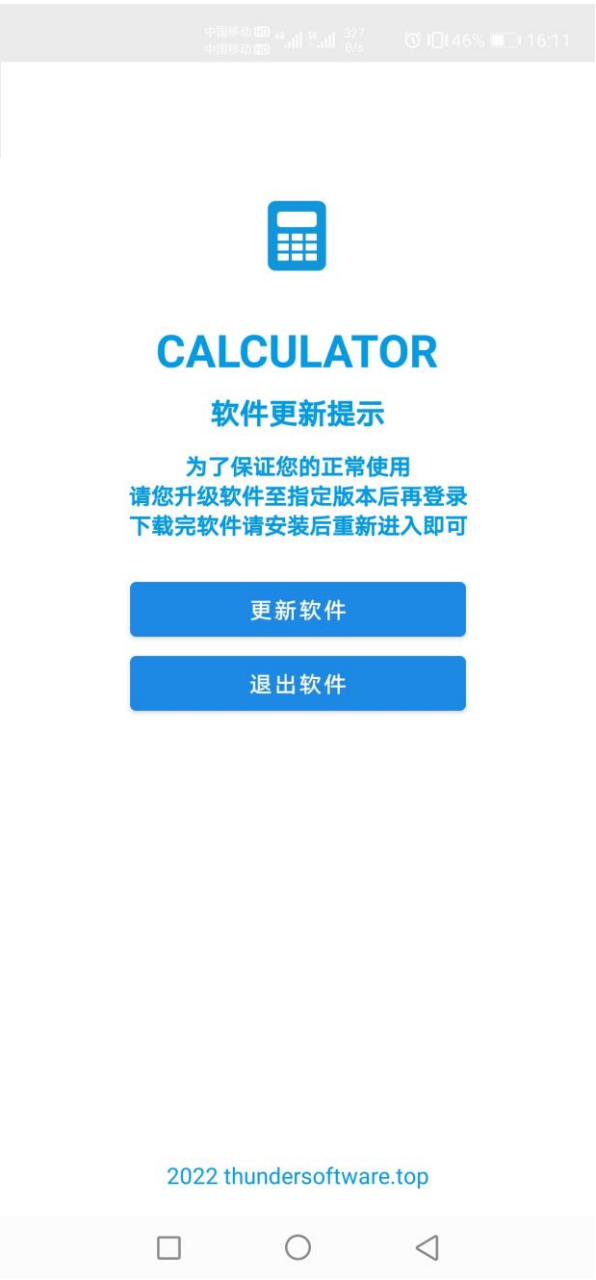
```
content){//写入文本到 txt
        FileOutputStream fileOutputStream = null;
        File file = new File(txtPath);
        try {
            if(file.exists()){
                file.createNewFile();
            }
            fileOutputStream = new FileOutputStream(file);
            fileOutputStream.write(content.getBytes());
            fileOutputStream.flush();
            fileOutputStream.close();
        } catch (Exception e) {
            e.printStackTrace();
        }

    }

}
package top.sonaradar.cal_server;

import top.sonaradar.cal_server.GUI.server_GUI;
import top.sonaradar.cal_server.server.stu;
import top.sonaradar.cal_server.server.studatabase;

public class start {
    public static void main(String args[]){
        server_GUI sg =new server_GUI();
        sg.init();
    }
}
```

六、运行截图

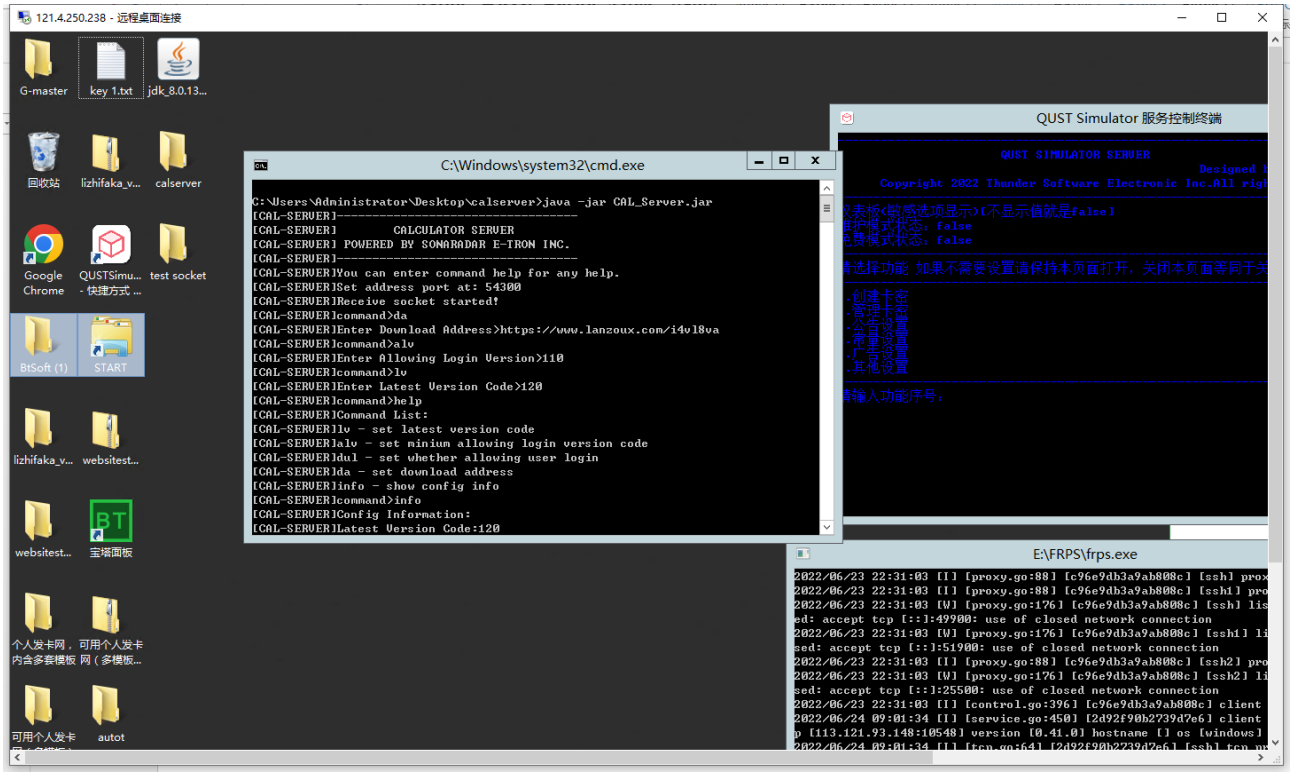
1.Android 客户端




CALCULATOR - Etcetera
Powered By Sonaradar.top. All rights reserved.
Application Information
VERSION
110
Developed By
Sonaradar
Electronic Inc
GUI Designer
Sonaradar
检测更新


CALCULATOR
软件更新提示
为了保证您的正常使用
请您升级软件至指定版本后再登录
下载完软件请安装后重新进入即可
更新软件
退出软件
2022 thundersoftware.top

2.服务器



上图为服务器端在腾讯云轻量服务器部署的运行截图

七、所分配任务完成情况

本项目中，我实际承担的任务是计算器整体的设计。计算器表达式算法能够计算带括号和更多运算符（取余、三角函数等）计算。功能层次上已经完成，但是部分细节问题需要继续完善。页面设计部分由于相对布局部分地方使用不够完善，实际在小于 5.5 英寸的手机测试下界面会冲突。服务器由于采用的是 socket 的简单通信，只能满足少量用户使用，若同时使用可能会导致服务器无法对客户端做出响应。客户端更新这个功能经过测试没啥大问题，基本挺正常的。

八、存在的问题

在本算法中，计算后的表达式如果为整数，运算结果实际会显示带一个小数位的数。对于三

角函数，仅能进行简单的一次三角函数计算，如果嵌套三角函数计算则无法计算。而进制转换仅能在计算完成后将结果转换为对应进制。对于如果没有写完的表达式，该算法会判断该表达式不成立，只有等到表达式全部写完才能运算，缺少更完善的实时计算预览显示。

九、个人体会

通过本次课题设计，本人认为计算器的设计并非是一个简单的项目，存在些许难度。

计算器不仅仅是简单的将给出的两个数通过计算得出结果。由于计算表达式涉及运算符众多，需要考虑运算优先级，如何设置检查优先级和数字与运算符分离都是本次项目的难点。而网上相关资料较少，部分功能基本无法查找相应资料，大部分时间需要自行尝试进行设计。而由于数学运算中部分算式是不成立（如 X/0 型）或者表达式压根不合法（如+ + --323），需要算法进行识别，而这又是项目的一个难点。

十、参考文献

1. Java 实现表达式计算求值

https://blog.csdn.net/a1439775520/article/details/96763582

2. 类集--Collections 工具类， 栈操作：Stack(理解)， 属性操作类：Properties（重点）

https://blog.csdn.net/a584898/article/details/81037514

3. java 表达式计算_基于 Java 语言的表达式计算

https://blog.csdn.net/weixin_35257663/article/details/114032323