



星云软件自动化测试工具

专业的IT培训专家
顾翔



目录

客户端下载与配置

项目编译

执行测试

视图

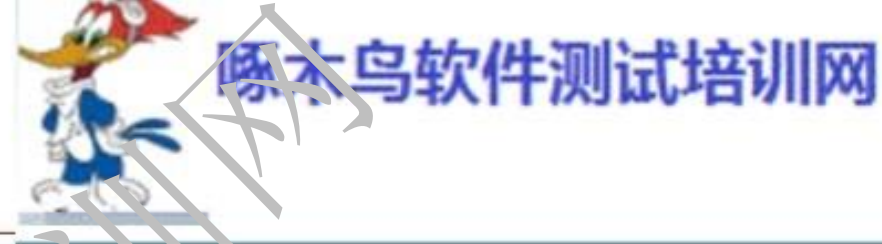


目

录



客户端下载与配置



1, 进入<http://www.teststars.cc/>, 选择“星云在线测试”

敬爱的用户，星云测试采用账户绑定模式，一个账户只能绑定一种模式的免费测试，如需要多种请重新申请星云用户



星云精准测试限量VIP码（企业版）

价值：12,000元/年

邀请码：

邀请码提交



客户端下载与配置



2, 星云在线测试客户端下载



[Android客户端环境配置详细文档](#)

星云Android版测试支持JAVA开发的android手机APK，免费测试为单用户测试

- 1 下载星云测试Android版客户端
- 2 星云客户端配置
需要安装JDK、SDK（测试中手机USB链接需要）、Eclipse/Android studio 等编译工具



客户端下载与配置



2, 星云在线测试客户端下载

测试账号信息

购买的时间	购买的云测类型	订单描述	价格	license数量	时间
2016-06-06 15:03:03	Horn-Android版本	云测天使体验	0/RMB	1	--

处理状态：已受理



已受理

时间：2016-06-06



已开通

时间：2016-06-06



服务结束

时间：--

客户端管理账号：xianggu625ts1554f

初始密码：xianggu625ts1554f

星云测试web端访问地址：120.131.81.105

星云测试web端访问端口：3000

高级企业功能：否



客户端下载与配置



3, 星云在线测试客户端配置

配置Java环境以及Android环境



目录

客户端下载与配置

项目编译

执行测试

视图



目

录





1, 项目编译使用前准备

第一步：android项目添加获取数据权限

注：权限为必须添加，不然会引发APK闪退等问题



修改需要编译工程下的AndroidManifest.xml文件加入以下内容用与接受数据时获得权限。

```
<uses-permission android:name="android.permission.READ_PHONE_STATE"></uses-permission>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"></uses-permission>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"></uses-permission>
<uses-permission android:name="android.permission.WAKE_LOCK"></uses-permission>
<uses-permission android:name="android.permission.INTERNET"></uses-permission>
```



第二步：在本机上用eclipse、或者其它编译工具先编译生成.class和APK，确保环境无错和代码结构正确。

2，建立项目以及版本
账号登录

注：星云在线客户端连接平台需要访问17262、17263端口。
请保证网络连接限制是否存在。

项目编译



星云在线测试客户端

文件 编辑 视图 企业项

多工程

控制流程图 函数列表 覆盖率 设置层级

登录 工程管理 编译 查看版本信息 导入导出测试用例 退出

服务端ip: 120.131.81.105 用户名: qwert 密码: 记住密码 强制登录 确认 取消

导航树

函数调用图 类继承图 函数调用聚集图

SC0 Both True SC1 Branch False C/DC SC1+ MC/DC

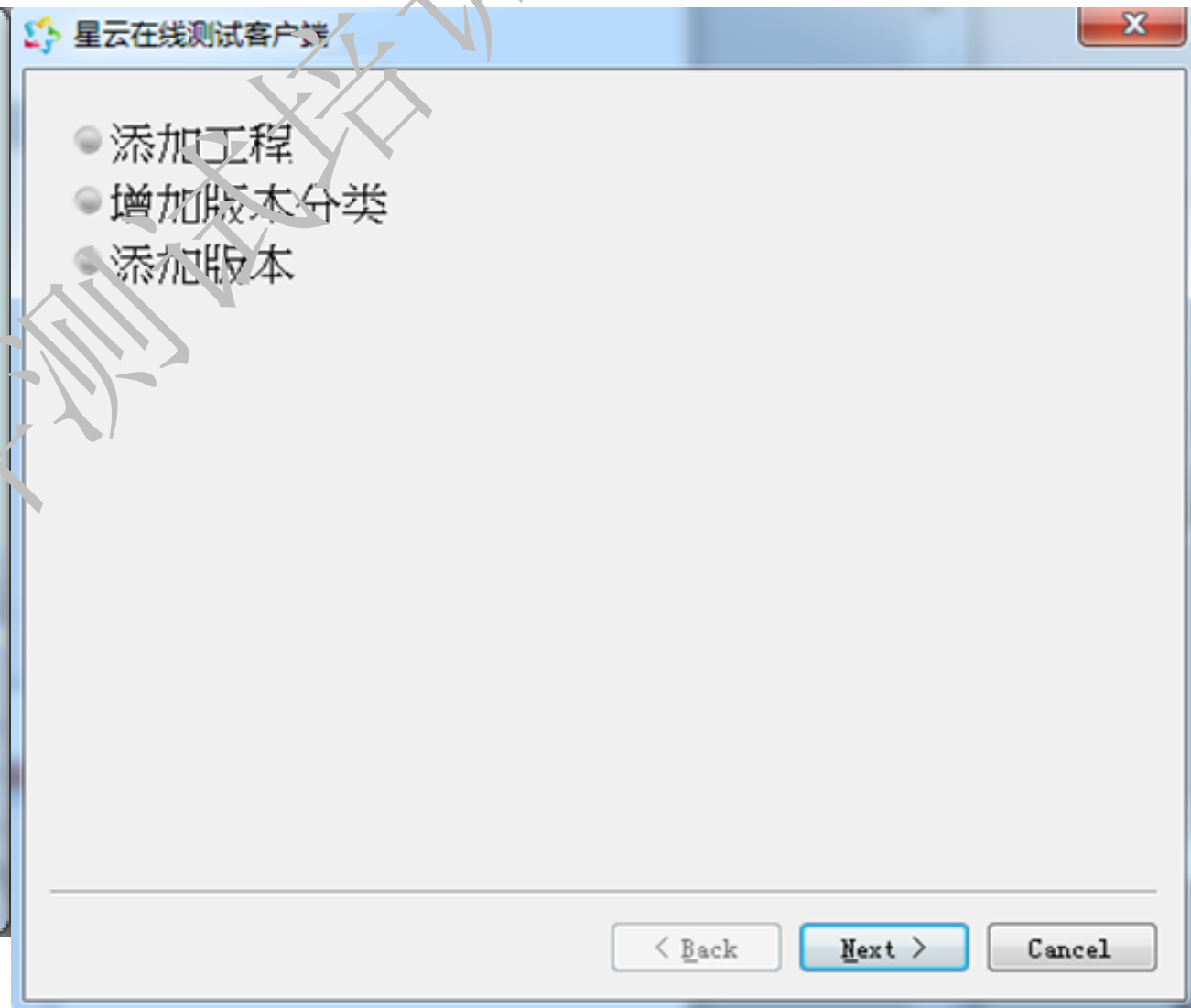
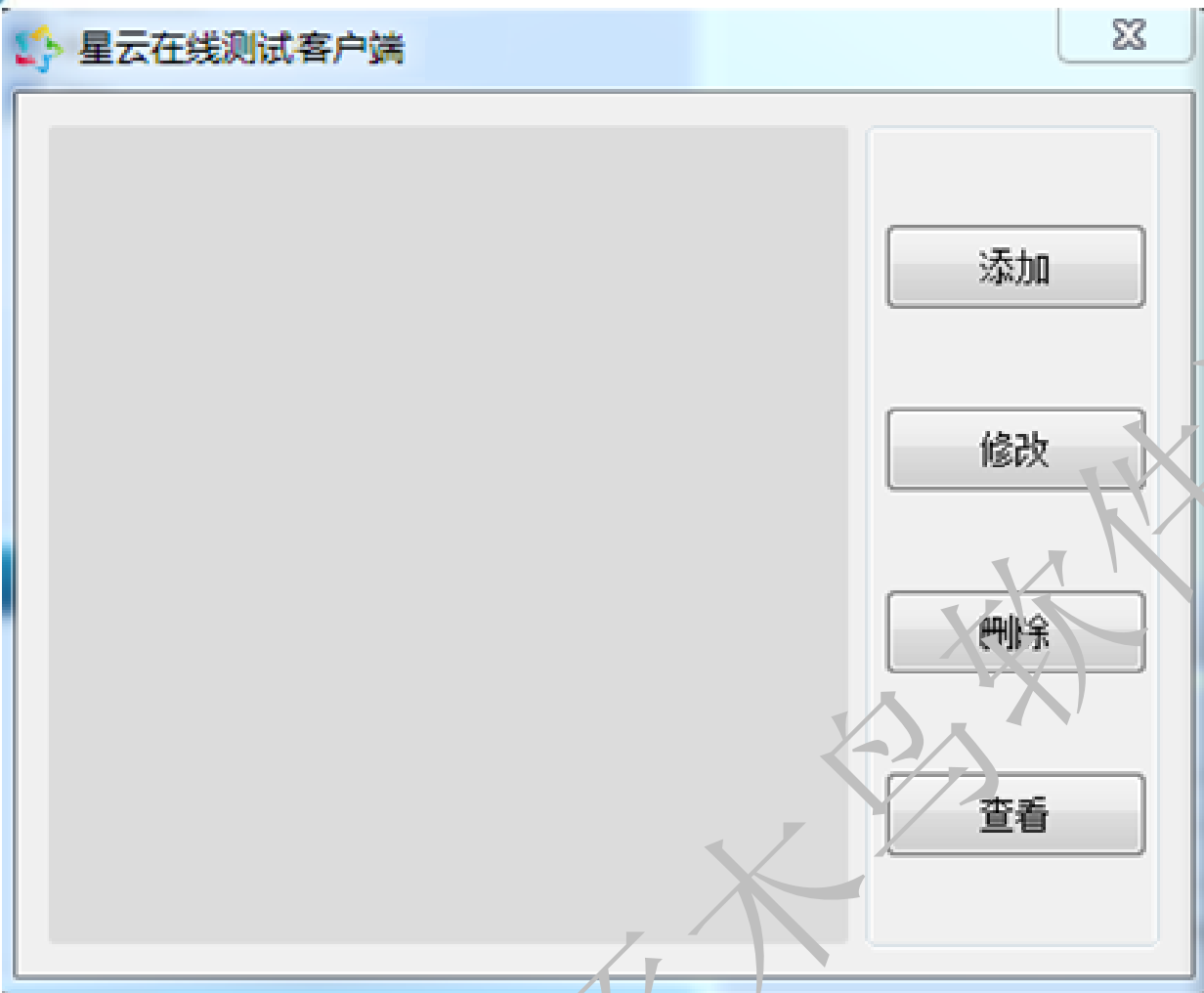
历史操作 书签

函数名称	JC1	JC1+	JC2	CC0	CC1
	0	0	0	0	0

函数原型: 属性: member function 声明文件: 声明类: 重载: 0 (modules) 内部类: 0 (modules) 全局静态变量: 0 (modules)

行数: 空行: 注释行: 基本信息 扇入扇出

项目编译



项目编译



星云在线测试客户端

增加工程：

名称：星云测试

备注：

创建时间：2016-04-12 17:01:26

负责人：

星云在线测试客户端

增加版本：

工程名称：

星云测试

分类名称：

无

版本名称：星云测试项目

数据库驱动：QMYSQL

创建时间：

2016/4/12 17:02

备注：

测试人员预计人数：0

测试预计结束时间：

2016/4/12 17:02

svn参数配置

SvnUrl

用户名

密码

项目名称

版本序号

< Back

Finish

Cancel

< Back

Finish

Cancel



项目编译



星云在线测试客户端 文件 编辑 视图 企业项

多工程

星云测试 工程
星云测试演示 版本
W

导航树

历史操作 书签

函数调用图 控制流程图 函数列表
● 复杂度 ● 覆盖率 ● 设置层级

函数调用图 类图 函数调用聚集团

SC0

Both

True

SC1

%

0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

Branch

False

C/DC

SC1+

MC/DC

函数名称	JC1	JC1+	JC2	CC0	CC1
	0	0	0	0	0

函数原型:
属性: member function
声明文件:
定义:
重载: 0(modules)
内部类: 0(modules)
全局静态变量: 0(modules)
行数:
空行:
注释行:

基本信息 层入层出





3, 版本编译

在这里以Eclipse为例:

用户只需要保证要编译的项目在eclipse或者别的编译IDE中编译通过。此过程会产生要编译工程中*.java对应的*.class文件, 该目录对应下面第二个参数, 默认会在工程下./bin/class下面。如果不是请手动选择对应目录。

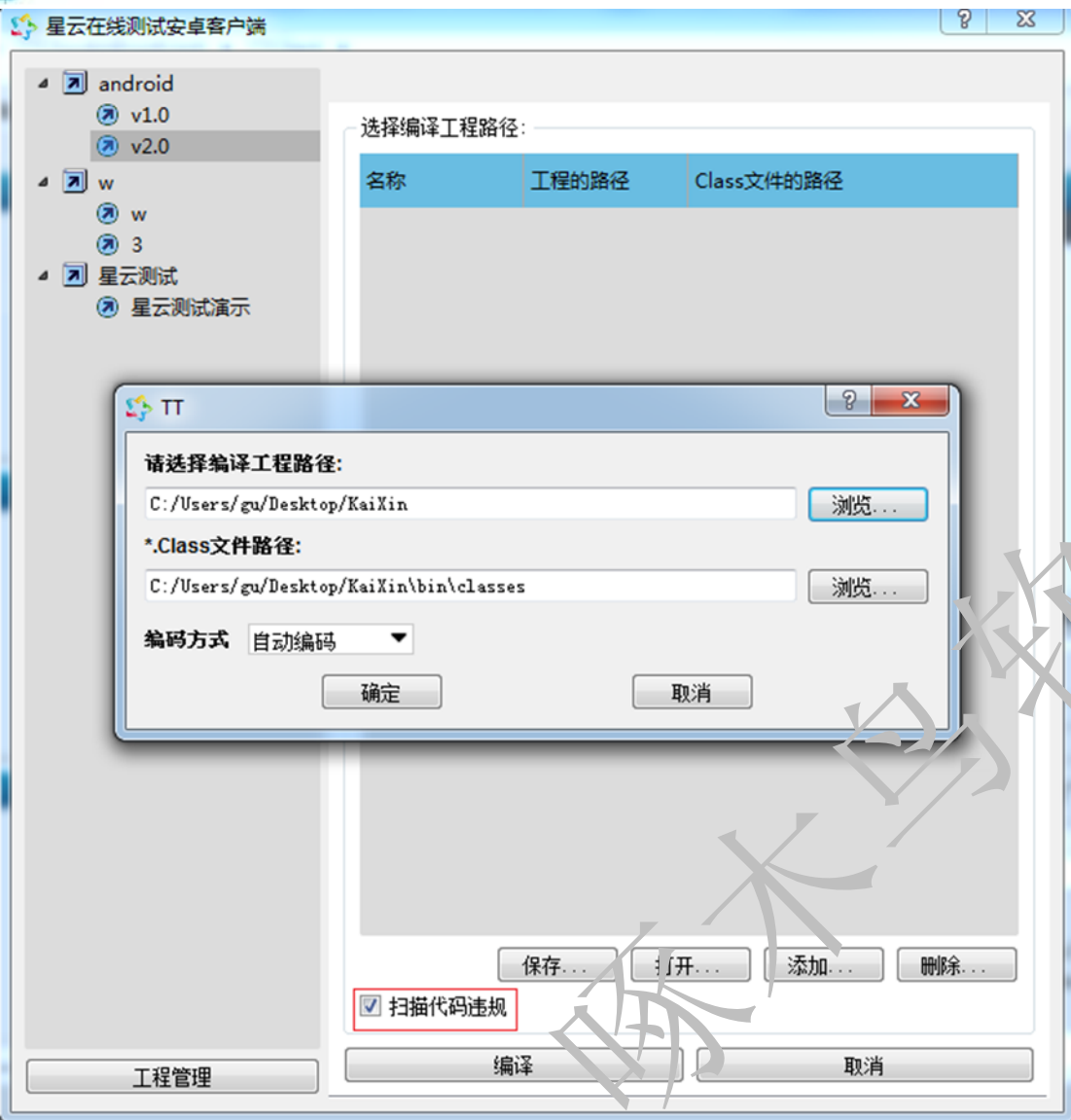
用户只需要选择需要编译的项目代码路径, *.class路径“星云测试”的默认为: 工程路径\bin\class, 在客户端选择“文件->编译”



项目编译

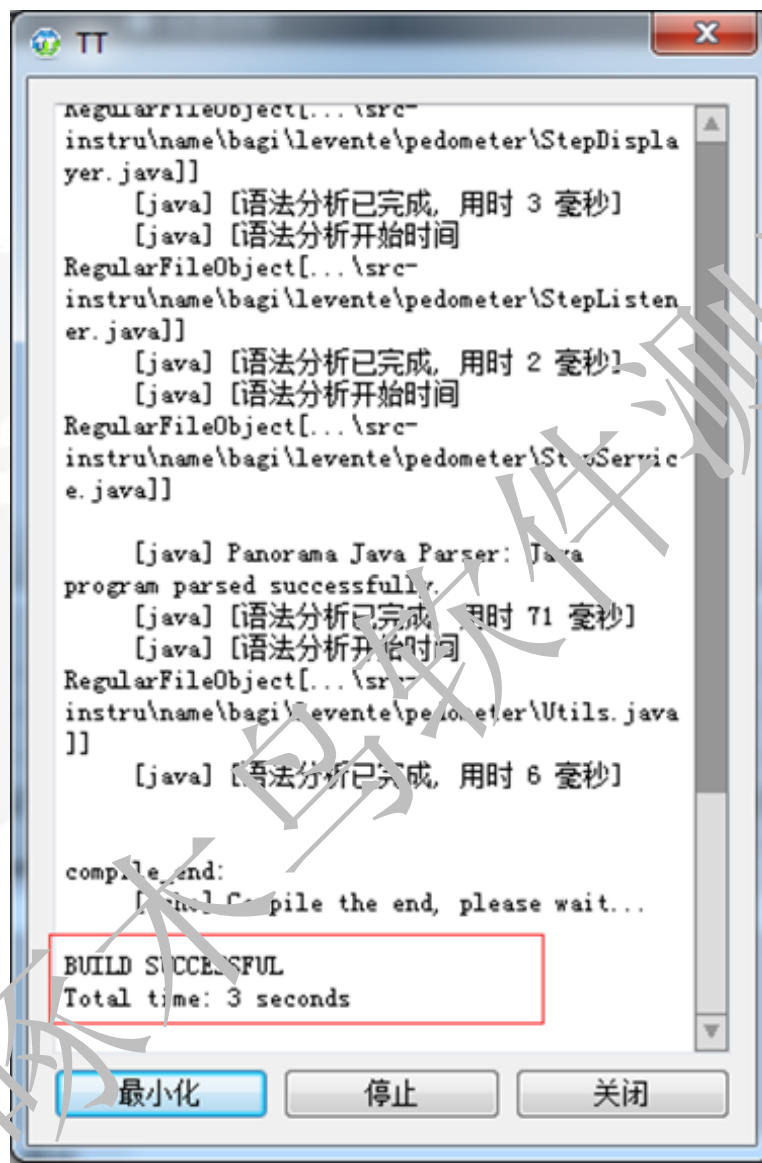


扫描代码违规采用静态PMD扫描方式，可以检测出代码中的违规和重复项，但是插装时间会相应的加上，如不需要可以不够选。

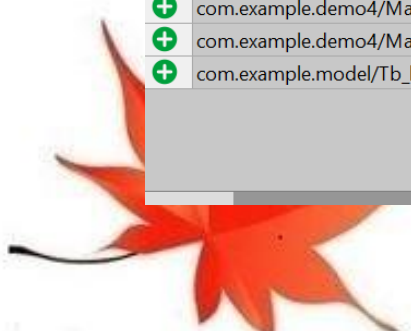


项目编译

点击【编译】



10



项目编译



啄木鸟软件测试培训网

星云在线测试安卓客户端

多工程 x 函数调用图控制流程图 函数列表

名称 块ID 行数 覆盖率

子块

com.example.demo4
v1.0

导航树

- com.example.demo4
 - MainActivity
 - void MainActivity::c
 - void MainActivity::L
 - void MainActivity::c
 - bool MainActivity::c
 - bool MainActivity::c
 - MainActivity\$Placehol
 - WellcomActivity

历史操作 书签

- + com.example.demo4/Mair
- + com.example.demo4/Mair
- + com.example.demo4/Mair
- + com.example.demo4/Mair
- + com.example.demo4/Mair
- + com.example.model/Tb_lo

代码

```
[0043] 36 2 public void LginOnClick(view view){
[0045] 37 2     username=et_user_name.getText().toString().trim();
38     password=et_password.getText().toString().trim();
[0046] 39 2     if ((username.isEmpty())||(password.isEmpty())){
[0047] 40 1         Toast.makeText(MainActivity.this, "请输入完全信息!",Toast.LENGTH_LONG).show(
41             et_username.setText(password);
42             et_password.setText(username);
[0048] 43 -     }else{
44         Tb_login tb_login=new Tb_login(0,username,password);
45         LoginDAO LD = new LoginDAO(MainActivity.this);
[0050] 46 1         if (LD.getCount(tb_login)==0){
[0051] 47 0             Toast.makeText(MainActivity.this, "不存在这个用户或者密码错误",Toast.LE
48             et_username.setText("");
49             et_password.setText("");
50         }
```

控制流程图

MC/DC

100



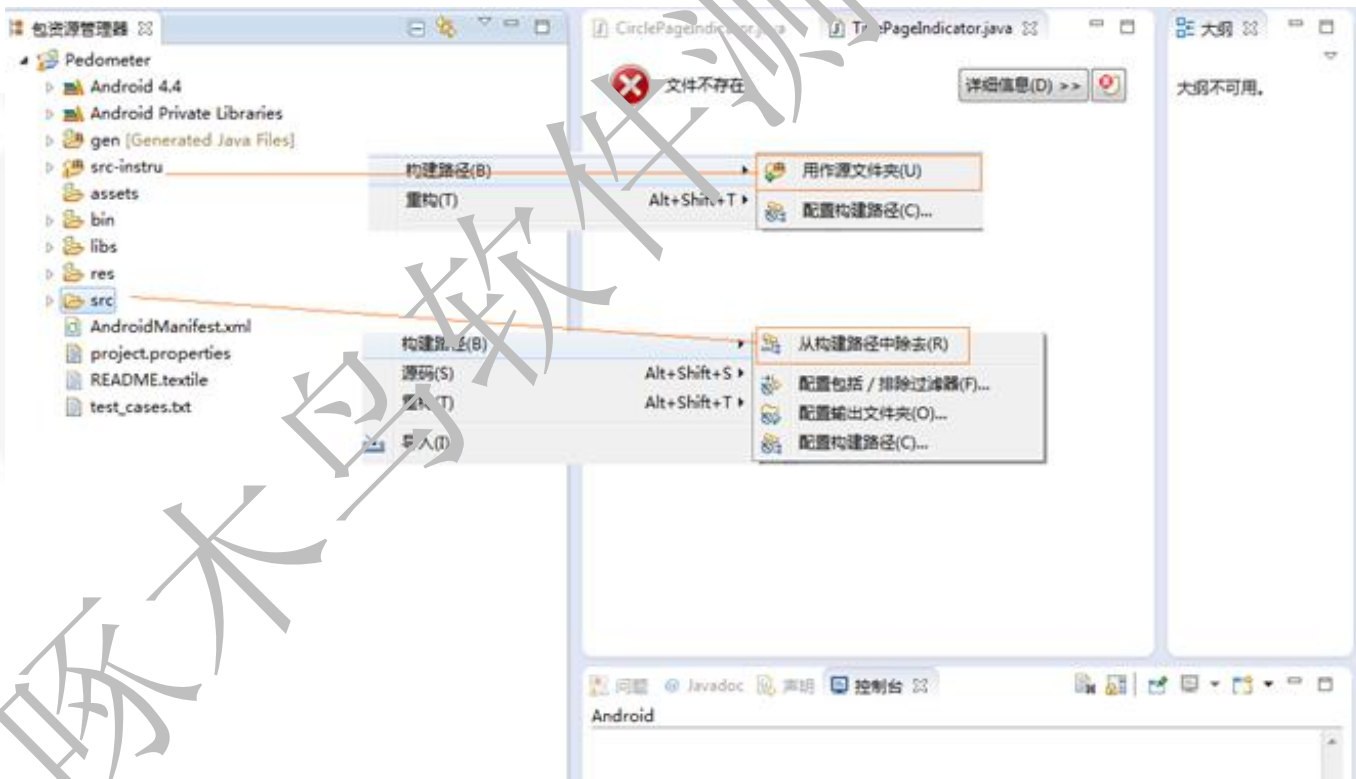
+	com.example.demo4/M
+	com.example.model/Tb

函数列表

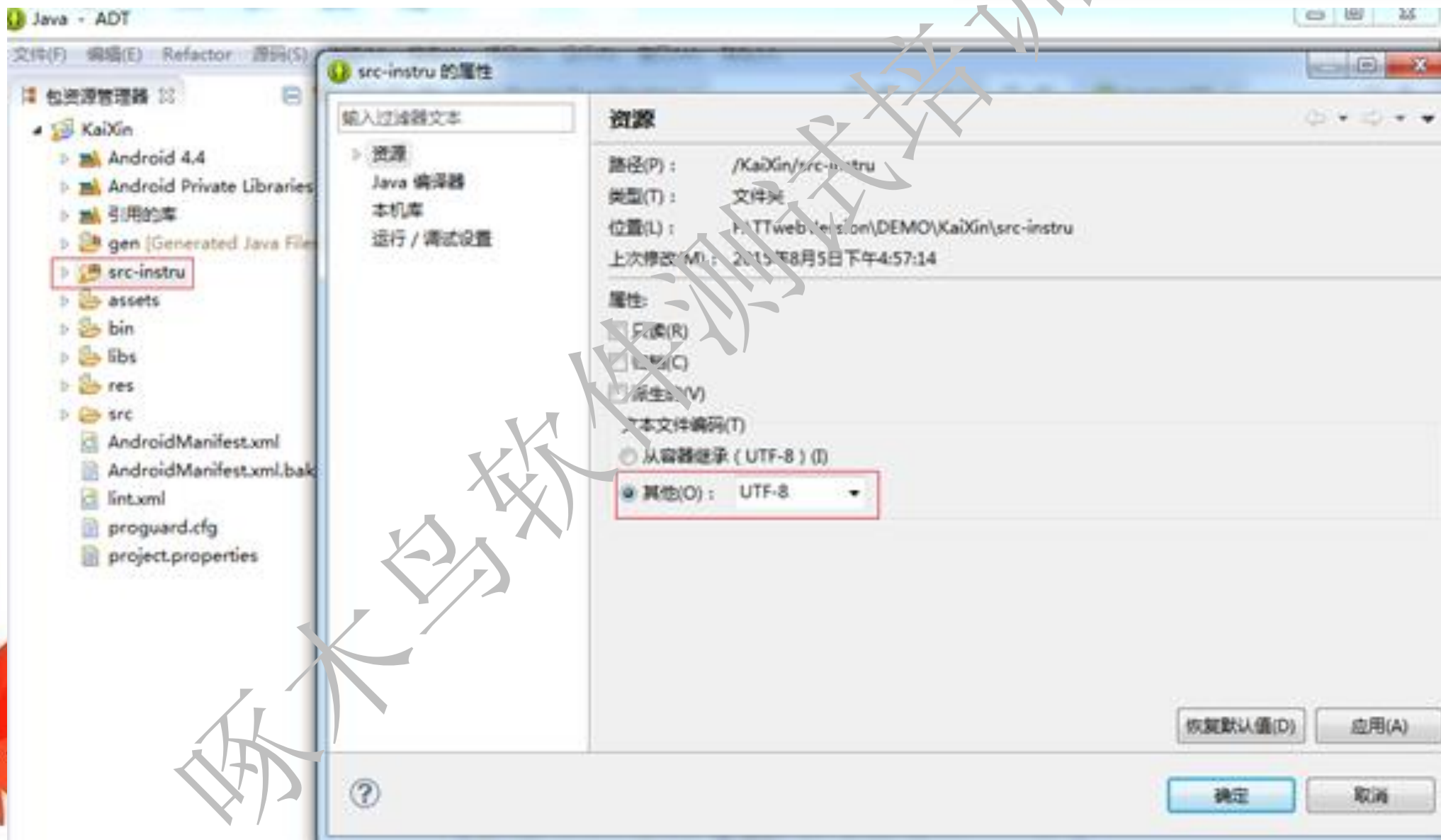


项目编译

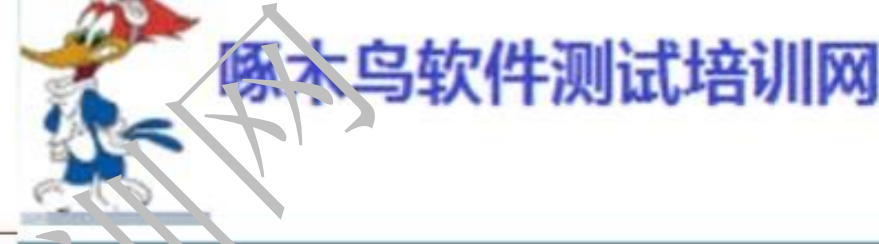
对应“星云测试”处理后的代码路径为编译项目路径/src-instru文件夹下，用户需要结合自己的工程配置来使用src-instru下的代码，使用src-instru下的代码来进行打包，注意此时src-instru下面是utf-8编码格式请在相应IDE中调整相应的源码文件的格式设置，生成的APK和‘星云测试’打包出来是一样的，之后即可测试。



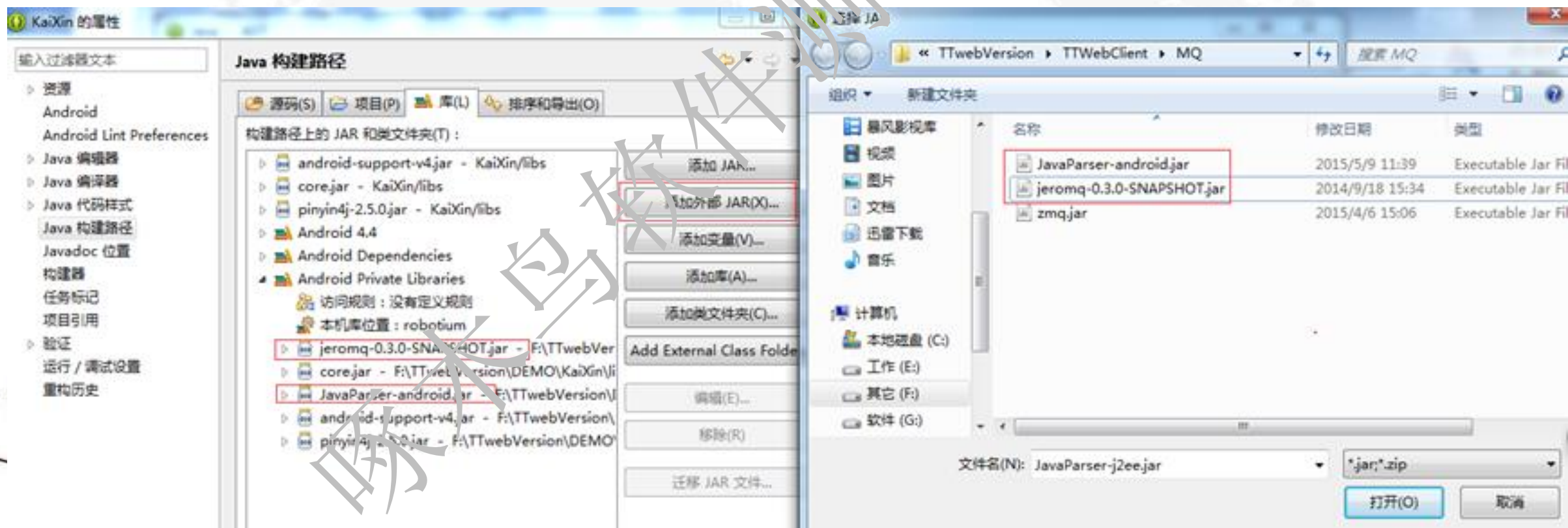
项目编译



项目编译



TTWebClient\MQ下的JavaParser-android.jar和jeromq-0.3.0-SNAPSHOT.jar，jar包会在‘星云测试’编译过程中自动加到项目的libs中，（注：有些开发软件不会主动加入，需手动添加依赖包）。**然后通过虚拟机或者打成APK文件，通过>adb push my.apk /data/data放入到真机中进行测试。**





目录



客户端下载与配置

项目编译

执行测试

视图



目	
	录



执行测试



- 1、安装星云在线编译后生成的专属APK进入测试手机
- 2、整个测试如更换了电脑设备则无需重新配置环境，只需直接打开星云在线测试客户端进行测试即可。

进入实时监控界面，点击菜单栏中的“视图”，选择数据传输监控图，进入实时监控界面。



执行测试

1, 在右上角的+该按钮操作进行功能分类的添加



执行测试



2, 添加需要的测试用例

星云在线测试安卓客户端

测试用例名称： 正确登录

预期输入：	预期输出：	注释：
输入正确的用户名和密码	正常登录	

操作步骤：	实际结果：
1, 输入用户名 2, 输入密码 3, 点击提交	正常登录

添加

执行测试

3，选中新建立的测试用例，将焦点放在该测试用例上，运行插桩后的源代码编译生成的程序

星云在线测试安卓客户端

测试用例

+

×

↺

↻

登录

☒ 正确登录

数据传输控制

块: Write/Receive

0/0

0 b/s

条件: Write/Receive

0/0

0 b/s

函数: Write/Receive

0/0

0 b/s

开始(&F5)

暂停

停止(&F7)

设置

提交问题

函数调用

自动刷新

类名称	函数名称	参数	返回类型	参数类型

基本信息

BUG信息

测试用例ID: 4

名称: 正确登录

开始运行时间: 未运行

当前运行时间: 0 s

设备:

预期输入: 输入正确的用户名和密码

操作步骤: 1, 输入用户名

预期输出: 正常登录

实际结果: 正常登录

注释:



执行测试

数据接收方式有以下几种方式：USB，WIFI，蓝牙等。

1, WIFI连接方式

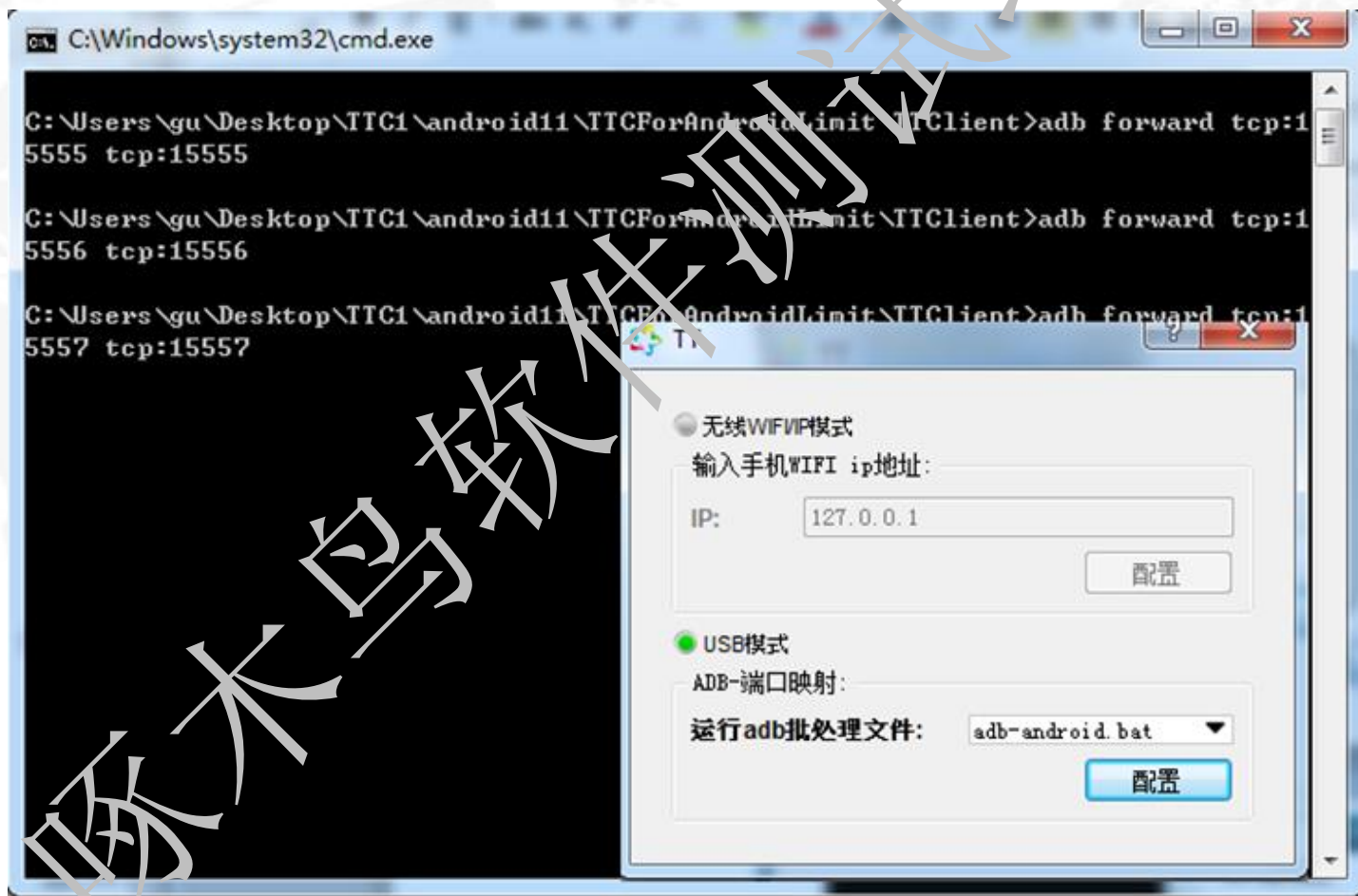
点击数据传输控-控制页面的“设置”，在IP输入框中输入手机当前连接的WIFI的IP地址点击确认后关闭小窗口。



执行测试

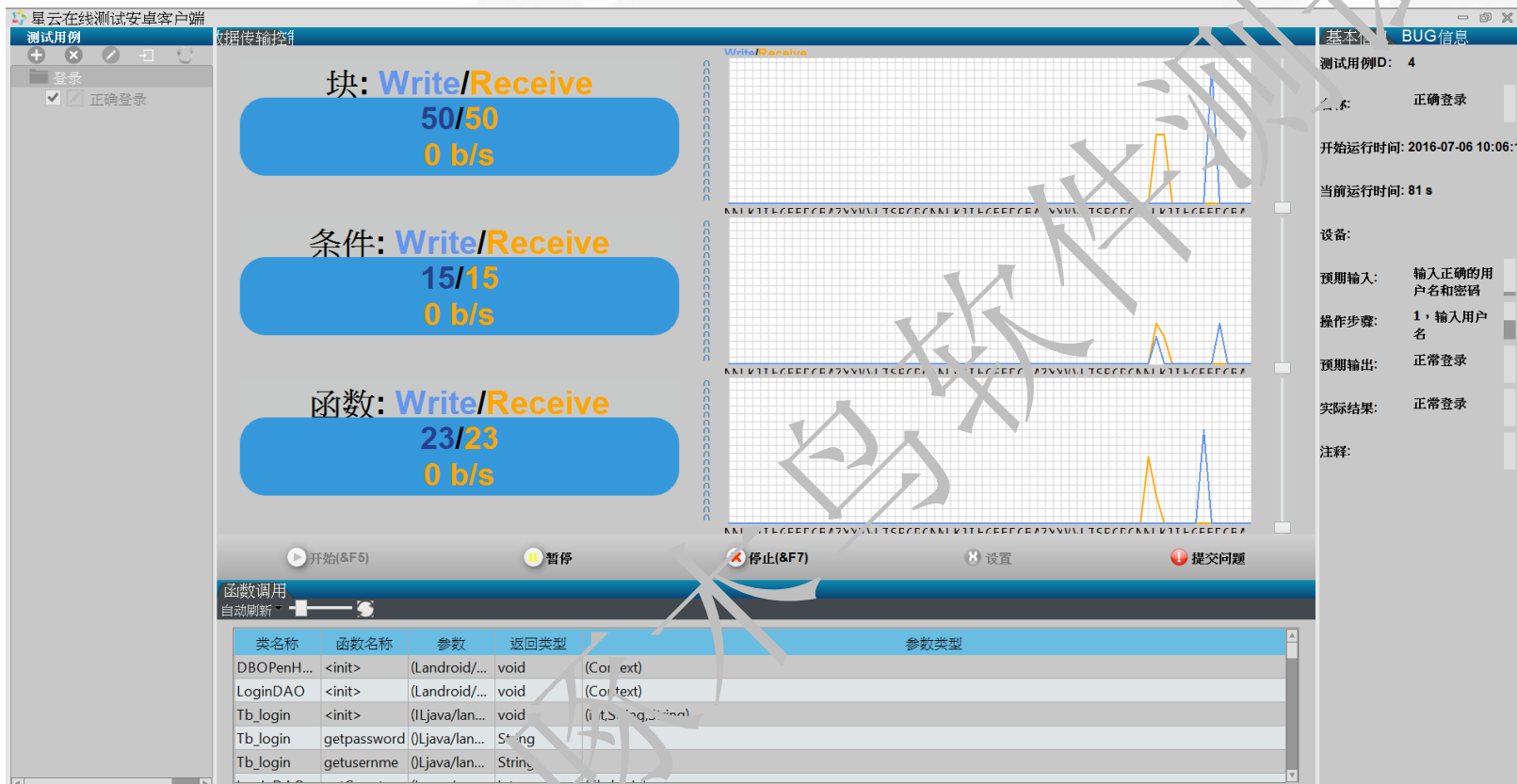
2, USB连接方式

USB模式默认IP为127.0.0.1, 选择ADT设置运行adb-android.bat脚本



执行测试

设置连接完毕后点击  开始 按钮，手机同时开始执行当前选择的用例，数据接收会显示波形图分为“块”，“条件”，“函数”这三部分，数据传输速度还有执行过的块，函数，条件都显示

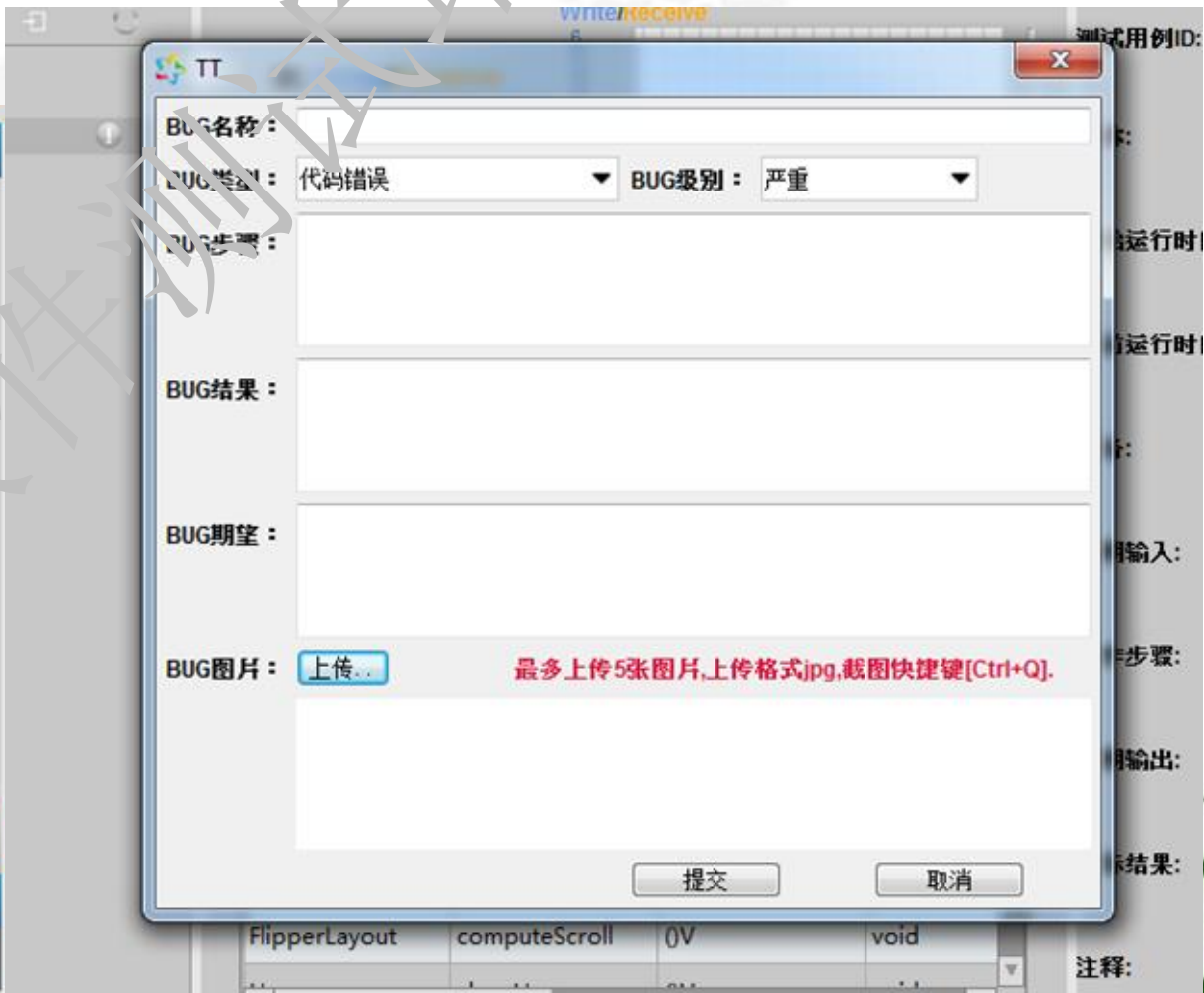




执行测试

3, BUG提交

在用例执行完毕后,如发现BUG,可点击提交问题,则可以对BUG进行描述并记录。



执行测试

提交完测试用例后，点击BUG信息就可以查看该测试用例所对应的BUG，星云在线测试的BUG是和测试用例相对应的，在该测试用例中只会反应出

该测试用例相关的BUG和历史BUG，当提交完BUG后该测试用例会出现一个！的状态，如出现崩溃则会自动捕获出现X的状态，点击详细就能对BUG进行各类操作，当BUG解决后状态消失。





目录



客户端下载与配置

项目编译

执行测试

视图



目	
	录



视图-函数列表

我们依次选中视图---主视图-----函数列表，选择上面的刷新按钮，可以看到我们之前测试的数据已经产生了测试覆盖率信息。

星云在线测试安卓客户端

多工程

函数调用图控制流程图 函数列表

跳转 排序 单页函数数目

请输入搜索函数名、... 搜索

编号	函数Id	函数名称	类	SC0	SC1	SC1+	RUI	ALS	LOT	C/DC	tranc	IC/D	JC0	JC1	IC1+	JC2	CC0	CC1
1	23	getpassword	com/example/...	100	100	100	--	--	--	--	--	--	1	1	1	1	1	1
2	21	getuserme	com/example/...	100	100	100	--	--	--	--	--	--	1	1	1	1	1	1
3	18	<init>	com/example/...	100	100	100	--	--	--	--	--	--	1	1	1	1	1	1
4	15	onCreateOptio...	com/example/...	100	100	100	--	--	--	--	--	--	1	1	1	1	1	1
5	14	LginOutClick	com/example/...	100	100	100	--	--	--	--	--	--	1	1	1	1	1	1
6	13	onCreate	com/example/...	100	100	100	--	--	--	--	--	--	1	1	1	1	1	1
7	8	onActivityResult	com/example/...	100	66.7	66.7	100	0	0	0	0	0	2	3	3	6	3	3
8	5	getCount	com/example/...	100	75	75	100	0	0	0	0	0	3	4	4	5	2	2
9	4	<init>	com/example/...	100	100	100	--	--	--	--	--	--	1	1	1	1	1	1
10	1	<init>	com/example/...	100	100	100	--	--	--	--	--	--	1	1	1	1	1	1
11	7	LginOnClick	com/example/...	60	60	60	0	100	0	0	0	0	5	7	5	9	4	4
12	24	setpassword	com/example/...	0	0	0	--	--	--	--	--	--	1	1	1	1	1	1
13	22	setusername	com/example/...	0	0	0	--	--	--	--	--	--	1	1	1	1	1	1
14	20	setid	com/example/...	0	0	0	--	--	--	--	--	--	1	1	1	1	1	1
15	19	getid	com/example/...	0	0	0	--	--	--	--	--	--	1	1	1	1	1	1
16	17	<init>	com/example/...	0	0	0	--	--	--	--	--	--	1	1	1	1	1	1
17	16	onOptionsIte...	com/example/...	0	0	0	0	0	0	0	0	0	3	4	4	5	2	2
18	12	onCreateView	com/example/...	0	0	0	--	--	--	--	--	--	1	1	1	1	1	1
19	11	<init>	com/example/...	0	0	0	--	--	--	--	--	--	1	1	1	1	1	1
20	10	onOptionsIte...	com/example/...	0	0	0	0	0	0	0	0	0	3	4	4	5	2	2

导航树

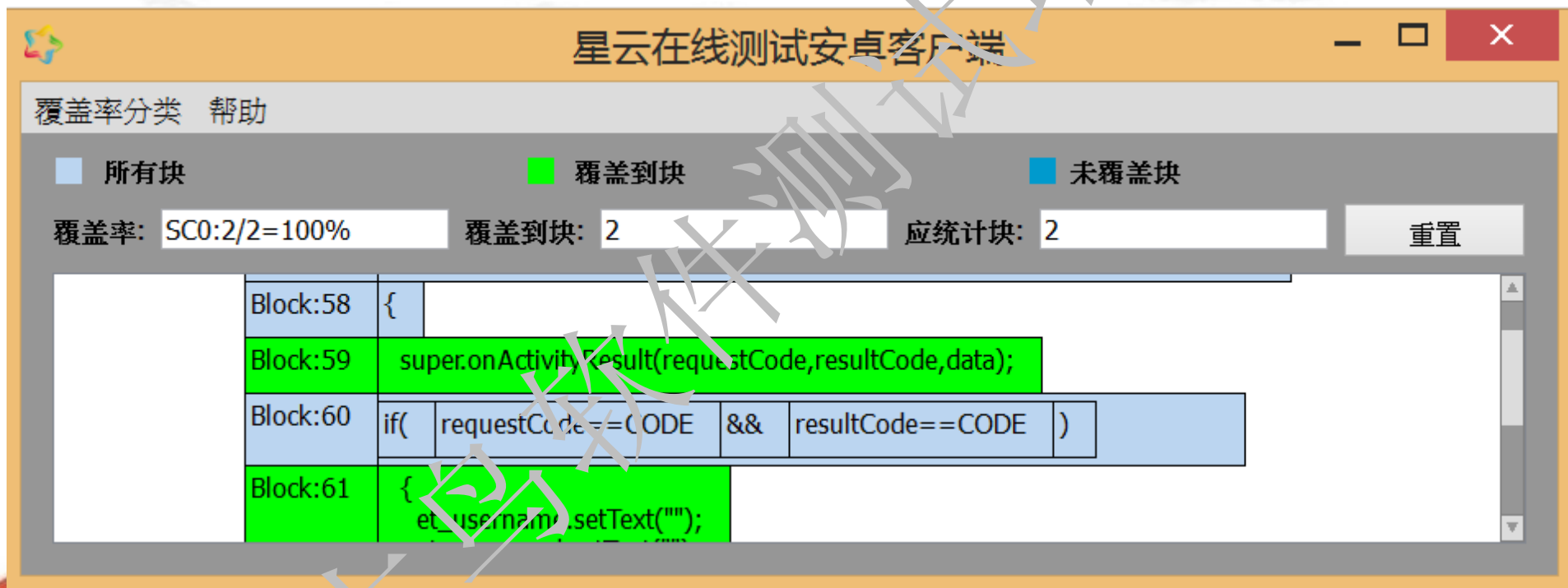
- com.example.dao
 - DBOPenHelp
 - void DBOPenHelp::
 - void DBOPenHelp::
 - void DBOPenHelp::

历史操作 书签

- com.example.model/Tb_login/
- com.example.demo4/Wellcom
- com.example.model/Tb_login/
- com.example.dao/DBOPenHel
- com.example.demo4/MainActi

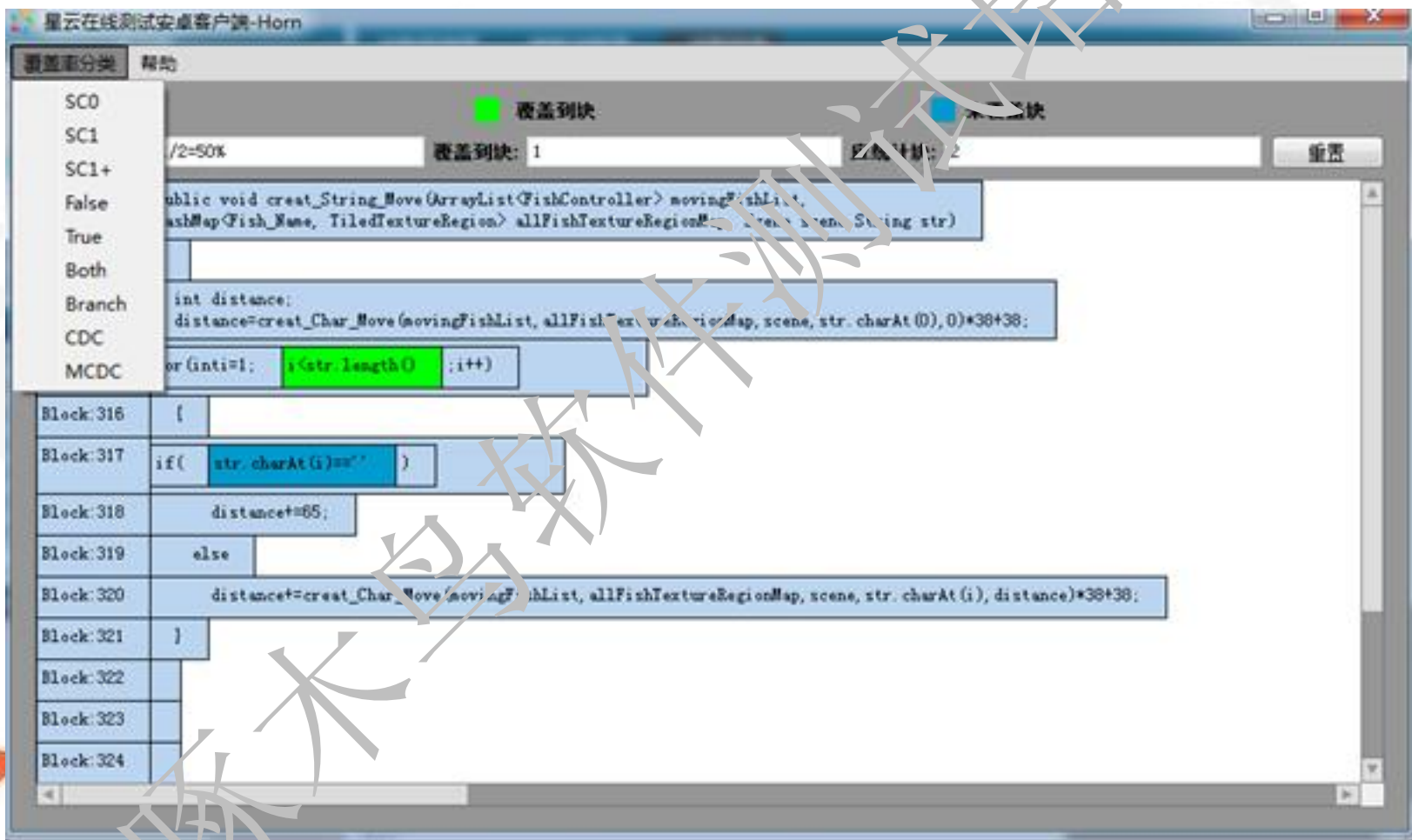
视图-函数列表

点中其个函数会展现出覆盖率窗口，通过绿色与蓝色的对比我们能发现之前运行过程中没有运行到的代码部分以及它的%比计算方式。



视图-函数列表

继续点击覆盖率分类可以选择不同的覆盖率计算方法用于显示



视图-控制流程图

通过函数列表中的跳转按钮，选择控制流程图，再次选择函数列表中的某个函数，这时窗口会调转到控制流程图窗口，可以看到每行代码的运行数，以及条件的运行状况与整个条件的进入与输出的流程状况。

多工程 函数调用图控制流程图 函数列表

子块

名称	块ID	行数	覆盖率
----	-----	----	-----

MC/DC 0%

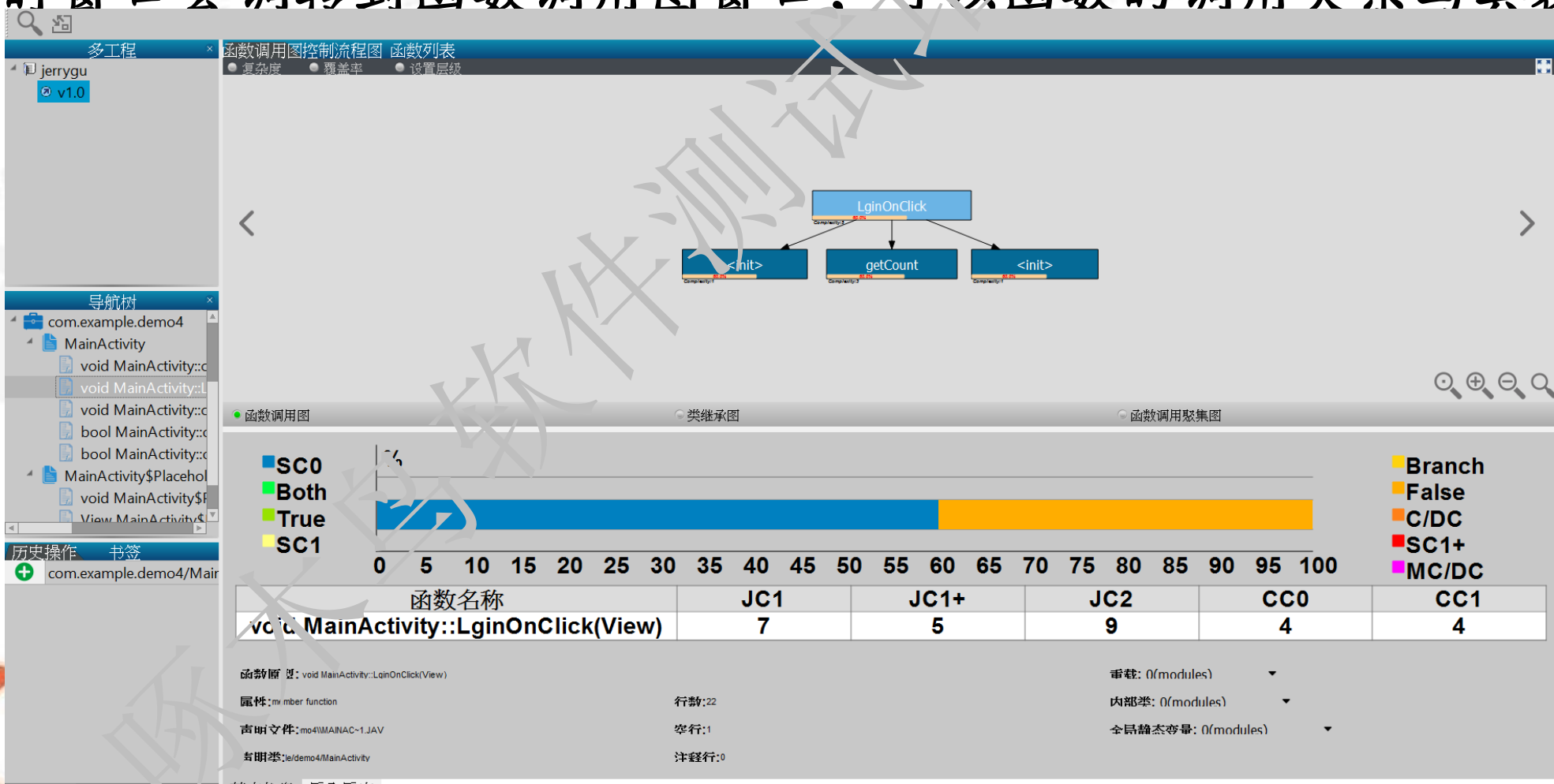
Condition	T	F
(userna...		✓
(passw...		✓
(userna...		✓

代码

```
public void LoginOnClick(View view){
    username=et_username.getText().toString().trim();
    password=et_password.getText().toString().trim();
    if ((username.isEmpty())||(password.isEmpty())){
        Toast.makeText(MainActivity.this, "请输入完全信息!", Toast.LENGTH_LONG).show();
        et_username.setText(password);
        et_password.setText(username);
    }else{
        Tb_login tb_login=new Tb_Login(0,username,password);
        LoginDAO LD = new LoginDAO(MainActivity.this);
        if (LD.getCount(tb_login)==0){
            Toast.makeText(MainActivity.this, "不存在这个用户或者密码错误", Toast.LENGTH_LONG).show();
            et_username.setText( "");
            et_password.setText( "");
        }
    }
}
```

视图-函数调用图

再次通过函数列表中的跳转按钮，选择函数调用图，选择函数列表中的某个函数，这时窗口会调转到函数调用图窗口，可以函数的调用关系与其覆盖率情况。



视图-测试用例与代码的追溯

依次选中视图---测试台，就能进行测试用例与代码的追溯界面，选择我们之前建立的测试用例，我们会有趣的发现我们选中的测试用例他会通过直接接收的测试数据进行一系列的追溯，包裹起接收的函数以及运行的代码，这样有利于我们对这个测试用例的执行进行了追踪

星云在线测试安卓客户端

正向追溯

反向追溯

DBOPenHelp

LoginDAO

- void LoginDAO::<init>(Context)
- int LoginDAO::getCount(Tb_login)

com.example.model

- Tb_login
 - void Tb_login::<init>()
 - void Tb_login::<init>(int,String,String)
 - int Tb_login::getid()
 - void Tb_login::setid(int)
 - String Tb_login::getusername()
 - void Tb_login::setusername(String)
 - String Tb_login::getpassword()
 - void Tb_login::setpassword(String)

com.example.demo4

控制流程图

代码

函数

函数调用图

版本 v1.0

基础版本: v1.0

	函数ID	函数名称	函数	类	修改
1	1	<init>	(Landroid...	com/exa...	--
2	4	<init>	(Landroid...	com/exa...	--
3	5	getCount	(Lcom/ex...	com/exa...	--
4	7	onCli...	(Landroid...	com/exa...	--
5	8	onActi...	(ILLandroi...	com/exa...	--
6	13	onCreate	(Landroid...	com/exa...	--
7	14	onOutC...	(Landroid...	com/exa...	--
8	15	onCreate...	(Landroid...	com/exa...	--
9	16	<init>	(Ljava/la...	com/exa...	--
10	21	getusern...	(Ljava/la...	com/exa...	--
11	23	getpassw...	(Ljava/la...	com/exa...	--
12	2	onCreate	(Landroid...	com/exa...	--
13	3	onUpgrade	(Landroid...	com/exa...	--
14	6	onCreate	(Landroid...	com/exa...	--
15	9	onCreate...	(Landroid...	com/exa...	--
16	10	onOption...	(Landroid...	com/exa...	--

测试用例追溯 函数追溯 代码追溯 依赖用例追溯最后运行的块

测试用例ID 测试用例名 始运行时 预期输入 预期输出 操作步骤 预期结果 备注 已运行时间

视图-测试用例与代码的追溯

正向追溯

登录

☒ 正确登录

反向追溯

DBOPenHelp

LoginDAO

void LoginDAO::<init>(Context)

int LoginDAO::getCount(Tb_login)

com.example.model

Tb_login

void Tb_login::<init>()

void Tb_login::<init>(int,String,String)

int Tb_login::getId()

void Tb_login::setId(int)

String Tb_login::getusername()

void Tb_login::setusername(String)

String Tb_login::getpassword()

void Tb_login::setpassword(String)

com.example.demo4

MainActivity

控制流程图 代码 函数 函数调用图

已经运行 未运行 不可运行

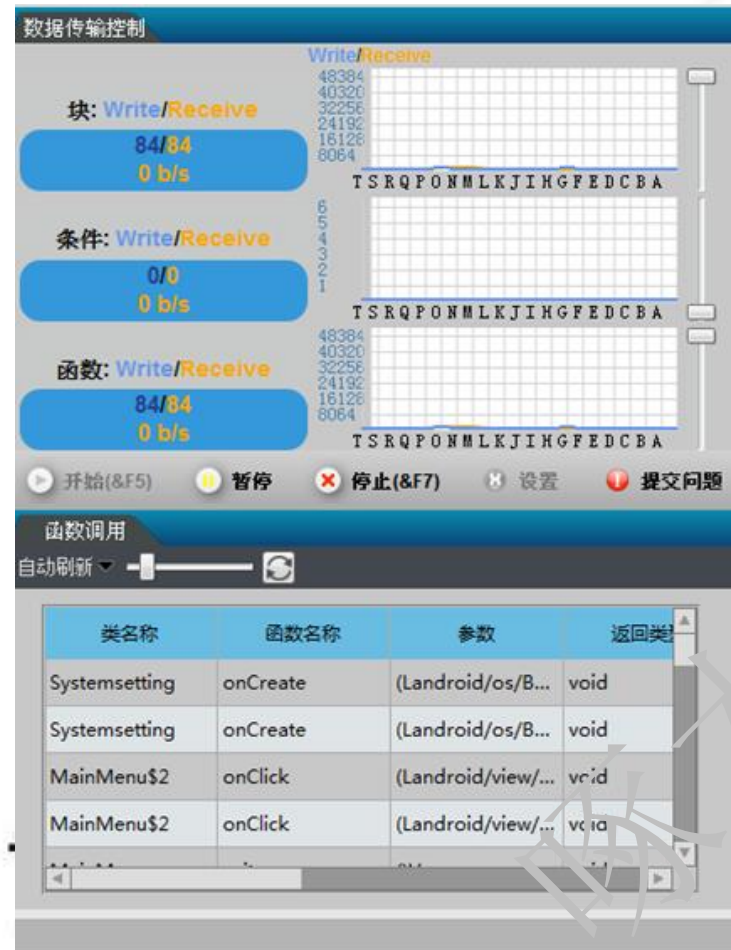
```
块:43--行:36 public void LginOnClick(View view)
块:44--行:36 {
块:45--行:37 username=et_username.getText().toString().trim();
password=et_password.getText().toString().trim();
块:46--行:39 if ((username.isEmpty())||(password.isEmpty()))
块:47--行:40 {
Toast.makeText(MainActivity.this, "用户名或密码不能为空", Toast.LENGTH_LONG).show();
et_username.setText(password);
et_password.setText(username);
块:48--行:43 
块:49--行:43 }else{
Tb_login tb_login=new Tb_login(0,username,password);
LoginDAO LD=new LoginDAO(MainActivity.this);
块:50--行:46 if (LD.getCount(tb_login)
块:51--行:47 )==0){
Toast.makeText(MainActivity.this, "用户名或密码不存在", Toast.LENGTH_LONG).show();
et_username.setText("");
et_password.setText("");
et_username.requestFocus();
块:52--行:51 estF
```

测试用例追溯 函数追溯 代码追溯 依赖用例追溯 最后运行的块

测试用例ID	测试用例名	开始运行时	预期输入	预期输出	操作步骤	预期结果	备注	已运行时间
--------	-------	-------	------	------	------	------	----	-------

视图-BUG追溯与BUG管理

再次建立一个测试用例，取名音效崩溃，并且运行，该测试用例，话多之前音效下面一份轴，此时程序发生闪推现象



点击最右边的刷新按钮，可以看到该测试用例已经发生了状态变化





视图-BUG追溯与BUG管理

点回之前的代码与测试用例的追溯界面，选中音效崩溃测试，选择最后50个块，我们会惊奇的发现，星云工具已经帮我们定位到了出错代码，查看代码可以看到该代码是因为有个空指针导致了其崩溃。

星云在线测试安卓客户端-Horn

文件 编辑 视图 企业项

正向追溯

游戏设置

- ☒ 音效调控
- ☒ 音效崩溃

游戏运行

反向追溯

- fishjoy.model.capturedfishinformation
- fishjoy.control.game.operation.fishoperation
- fishjoy.control.game.factory
- fishjoy.model.numberinformation
- fishjoy.control.game
- fishjoy.model.gameinformation
- fishjoy.model.scoreinformation
- fishjoy.control.about
- fishjoy.control.menu
 - MainMenu
 - void MainMenu::onCreate(Bundle)
 - void MainMenu::startGame()
 - void MainMenu::set()
 - void MainMenu::help()
 - void MainMenu::billboard()

控制流程图 代码 函数 函数调用图

已经运行 未运行 不可运行

块:2192--行:69

```
Sound Silo = null; // SoundFactory.createSoundFromAsset(activity.getSoundManager(), act v  
Silo.setVolume(SystemSetting.this.soundVolume);  
Silo.play();
```

块:2193--行:73

测试用例追溯 函数追溯 代码追溯 函数调用追溯 最后运行的块

最后运行

	块号	文件	块首行	执行序号
1	2192	28	69	25
2	2184	28	61	24
3	2184	28	61	23
4	2184	28	61	22
5	2184	28	61	21
6	2184	28	61	20
7	2184	28	61	19

块 条件 判定



视图-BUG追溯与BUG管理

我们再次回到测试传输监控试图选择 **提交问题**，填写相信的报错信息提交，可以查看到测试用例的状态又一次发生了变化，并且对应上了提交BUG的信息。

TT

BUG名称：

崩溃

BUG类型：

代码错误

BUG级别：

严重

BUG步骤：

1. 选择游戏设置
2. 调节音效第二根轴键

BUG结果：

崩溃

BUG期望：

BUG图片：

上传...

最多上传5张图片,上传格式jpg, 截图快捷键[Ctrl+Q].

提交

取消

星云在线测试安卓客户端-Horn

测试用例

数据传输控制

块: Write/Receive

0/0
0 b/s

条件: Write/Receive

0/0
0 b/s

函数: Write/Receive

0/0
0 b/s

开始(&F5)

暂停

停止(&F7)

设置

提交问题

函数调用

自动刷新

类名称	函数名称	参数	返回类

Write/Receive

48384
40320
32256
24192
16128
8064

TSRQPONMLKJINGFEDCBA

基本信息

BUG信息

详细

BUG状态：

打开

BUG名称：

崩溃

BUG提交日期：

2016-06-15 16:49:35

BUG提交人：

user

BUG严重程度：

高

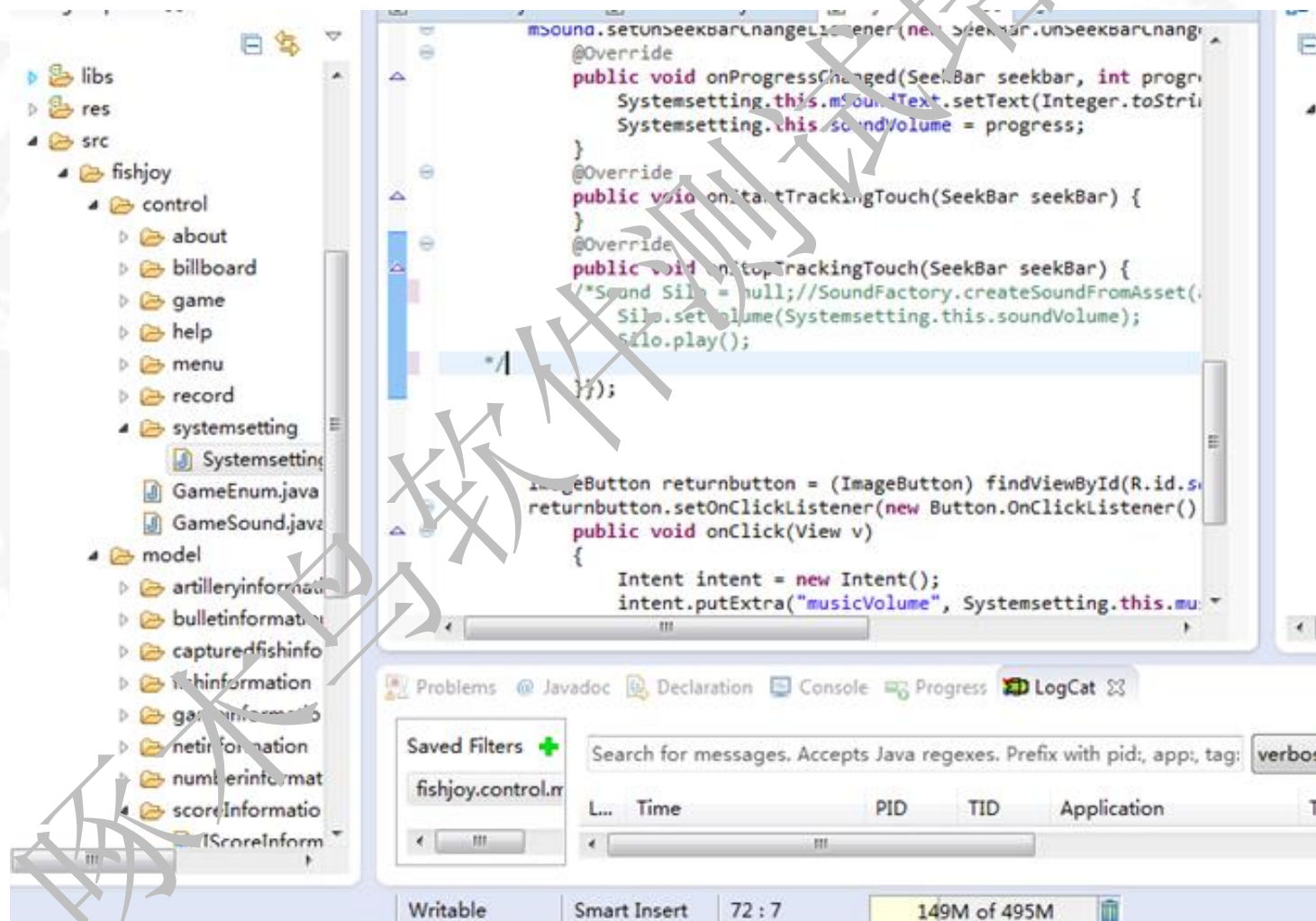
BUG所在版本：

1.0



视图-回归测试用例选取

我们打开eclipse找到该段代码，对其进行修改，并重新运行版本的创建于编译





视图-回归测试用例选取

点击**企业项**的--回归测试用例选取并进行运行，我们可以通过下方2张图，看到星云测试工具已经很好的帮我们选出了测试用例需要回归的优先级，以及变化的函数情况。

回归用例选取

回归生成测试用例数： 3

排序： 测试用例回归优先级

回归版本测试用例id	回归版本测试用例名称	测试用例最后运行版本	测试用例最后运行时间	测试用例最后运行状态	测试用例回归优先级	函数剖面变化比例	
8	音效崩溃	1.0	2016-06-15 16:40:52	崩溃	2	15.4%	详细
5	音效调控	1.0	2016-06-15 14:34:43	通过	1	8.3%	详细
7	普通模式游戏	1.0	2016-06-15 14:35:58	通过	0	0	详细

回归用例选取

函数总数： 13

改变的函数数： 2

删除的函数数： 0

未发生变化的函数数： 11

回归版本测试用例id	回归版本测试用例名称	测试用例最后运行版本	测试用例最后运行时间	测试用例最后运行状态	测试用例回归优先级	函数剖面变化比例
8	音效崩溃	1.0	2016-06-15 16:40:52	崩溃	2	15.4%

详细函数列表计算

回归版本函数ID	函数名称	类名	函数变化	标识位
215	onStopTrackingTouch	fishjoy/control/systemsetting/Systemsetting\$2	修改	+1
209	onCreate	fishjoy/control/systemsetting/Systemsetting	修改	+1
214	onStartTrackingTouch	fishjoy/control/systemsetting/Systemsetting\$2	未变	0
213	onProgressChanged	fishjoy/control/systemsetting/Systemsetting\$2	未变	0
186	onClick	fishjoy/control/menu/MainMenu\$2	未变	0
184	about	fishjoy/control/menu/MainMenu	未变	0
180	set	fishjoy/control/menu/MainMenu	未变	0
181	help	fishjoy/control/menu/MainMenu	未变	0



视图-代码违规与重复度

我们选中视图中的违规----代码度重复分析，可以看到星云已经提供给我们了一个静态扫描的结果，里面的代码规范以及代码的重复度都有了相应的展示。

The screenshot displays the Starry Sky software testing tool interface, specifically the 'View' section for 'Code Duplication Analysis'. The interface is divided into several panes:

- Left Pane (Table):** Lists duplication items with columns: ID, Name, Line Count, Character Count, and File Path. The table shows 7 items, with the first item (ID 0) selected.
- Top Right Pane (Table):** Shows a summary of duplication items with columns: File Path, Line Count, Start Line, End Line, Start Column, End Column, and Rule. The first item (ID 0) is selected.
- Bottom Left Pane (Code Editor):** Displays the source code for the selected item, showing a loop that iterates over a list of fish and checks for collisions.
- Bottom Right Pane (Code Editor):** Displays the source code for the selected item, showing a class definition for FishJoy that extends BaseActivity and implements JOnSceneTouchListener.

ID	Name	Line Count	Character Count	File Path
0	duplication_11_98_0	11	98	src/fishjoy/control/game/operation/scenemonitor.java
1	duplication_8_82_1	8	82	src/fishjoy/control/game/operation/scenemonitor.java
2	duplication_8_68_2	8	68	src/fishjoy/control/about/about.java
3	duplication_12_62_3	12	62	src/fishjoy/control/billboard/billboard.java
4	duplication_2_60_4	2	60	src/fishjoy/control/game/fishjoy.java
5	duplication_16_58_5	16	58	src/fishjoy/control/game/gameconstants.java
6	duplication_5_58_6	5	58	src/fishjoy/control/game/factory/fishfactory.java
7	duplication_11_97_7	11	97	src/fishjoy/control/game/factory/userfactory.java

File Path	Line Count	Start Line	End Line	Start Column	End Column	Rule
src/fishjoy/control/GameEnum.java	10	1	295	1	1	ExcessiveIn
src/fishjoy/control/GameSound.java	36	50	295	8	1	AtLeastOne
src/fishjoy/control/about/about.java	23	50	295	8	1	Cyclomatic
src/fishjoy/control/billboard/billboard.java	166	50	295	8	1	Modified
src/fishjoy/control/game/fishjoy.java	264	50	295	8	1	StdCycloma
src/fishjoy/control/game/GameConstants.java	38	50	295	95	1	TooManyM
src/fishjoy/control/game/factory/fishfactory.java	197	50	295	8	1	TooManyFi
src/fishjoy/control/game/factory/userfactory.java	2	50	295	8	1	Comment
src/fishjoy/control/game/operation/ModelInfo...	79	51	51	27	35	BeanMend
src/fishjoy/control/game/operation/TextureRegl...	224	51	51	9	16	DefaultPack
src/fishjoy/control/game/operation/artilleryope...	75	51	51	9	16	Comment

```
fishCollided.add(movingFishLi
}
if(!fishCollided.isEmpty()){
    for(int j = 0; j < fishCollided.size(); j++){
        if(fishCollided.get(j).canBeC
        fishCollided.get(j).
        all
        mScore = scoringOper
        all
        movingFishList.remove
    }
}
```

```
public class FishJoy extends BaseActivity implements JOnSceneTouchListener, GameConstant
    MyDatabaseAdapter mDatabase;
    private Camera mCamera;
    private Scene mScene, mPauseScene;

    private TextureRegion mBackgroundTextureRegion, returnButtonTextureRegion, retryBut
    private TiledTextureRegion artilleryTextureRegion, pauseButtonTextureRegion, sound
    private HashMap<FishName, TiledTextureRegion> allMovingFishTextureRegionMap = new
    private HashMap<FishName, TiledTextureRegion> allArtilleryFishTextureRegionMap = new
```

视图-测试用例的聚类分析

为了能更好的展示聚类分析这个功能，我预先做了上百个测试用例，选中星云工具企业项---聚类结果，输入分类数，发现星云工具已经把我们的测试用例通过其测试用例与代码的追溯剖面，分成了我们之前输入的分类数，选中其中一个分类，我们可以看到下图，该功能把支付能的测试用例自动化分成了一类，并显示出了类之间的距离，这样的功能有利于我们对测试用例的评审与BUG的查找。当支付中某个测试用例存在BUG时，那相邻的测试用例都有可能会有被波及的风险。



视图-测试用例的聚类分析

星云在线测试安卓客户端-Horn

30

版本Id : 177 聚类个数 : 30 聚类参数 : 欧氏距离 类别: 开始聚类

类别	测试用例中心点ID	类内个数	类内最大距离	类内平均距离	类内直径	类间距离	测试用例中心点ID	测试用例名称
1	77	6	4.90	3.52	5.48	6.40	17	退款订单返回订单列表
2	149	1	0.00	0.00	0.00	8.60	81	微信支付
3	155	39	5.00	1.27	6.24	2.45	82	支付宝支付
4	52	16	4.90	2.81	6.48	2.45	83	支付成功
5	10	3	7.42	4.73	9.75	7.94		
6	146	3	4.69	2.94	5.74	5.92		
7	81	4	6.08	3.94	6.86	4.90		
8	32	8	6.40	3.85	8.77	8.19		
9	102	11	5.00	2.10	6.16	3.74		
10	21	5	7.68	4.91	9.33	6.71		
11	104	3	5.00	2.52	5.48	5.48		
12	93	3	5.00	3.15	6.71	5.48		
13	56	5	4.90	3.41	6.16	3.74		
14	108	8	5.00	3.22	5.92	5.83		
15	23	3	7.42	3.96	8.54	8.66		
16	28	1	0.00	0.00	0.00	8.25		
17	29	1	0.00	0.00	0.00	8.06		
18	124	1	4.26	2.22	4.47	8.27		

视图-报表-项目报表

星云提供了一份网页的报表，我们通过发放的账号和密码进入，发现我们每天的运行的测试用例的覆盖率都已经BUG，测试用例的运行数等输入都已经绘制了很好的展示报表。（不要用IE）

主页 星云在线测试 星云测试概述 企业级产品 解决方案

敬爱的用户，测试账号会在每个工作日的下午6点-8点发放。

测试账号信息

购买的时间	购买的云测类型
2016-06-06 15:03:03	Horn-Android版本

处理状态：已受理



已受理

时间：2016-06-06



已开通

时间：2

客户端管理账号：

初始密码：

星云测试web端访问地址：120.131.81.105

星云测试web端访问端口：3000

高级企业功能：否



视图-报表-项目报表

星云提供了一份网页的报表，我们通过发放的账号和密码进入，发现我们每天的运行的测试用例的覆盖率都已经BUG，测试用例的运行数等输入都已经绘制了很好的展示报表。（不要用360，IE，腾讯）

主頁 星云在线测试 星云测试概述 企业级产品 解决方案

敬爱的用户，测试账号会在每个工作日的下午6点-8点发放。

测试账号信息

购买的时间	购买的云测类型
2016-06-06 15:03:03	Horn-Android版本

处理状态：已受理

已受理

已开通

客户端管理账号：

初始密码：

星云测试web端访问地址：120.131.81.105

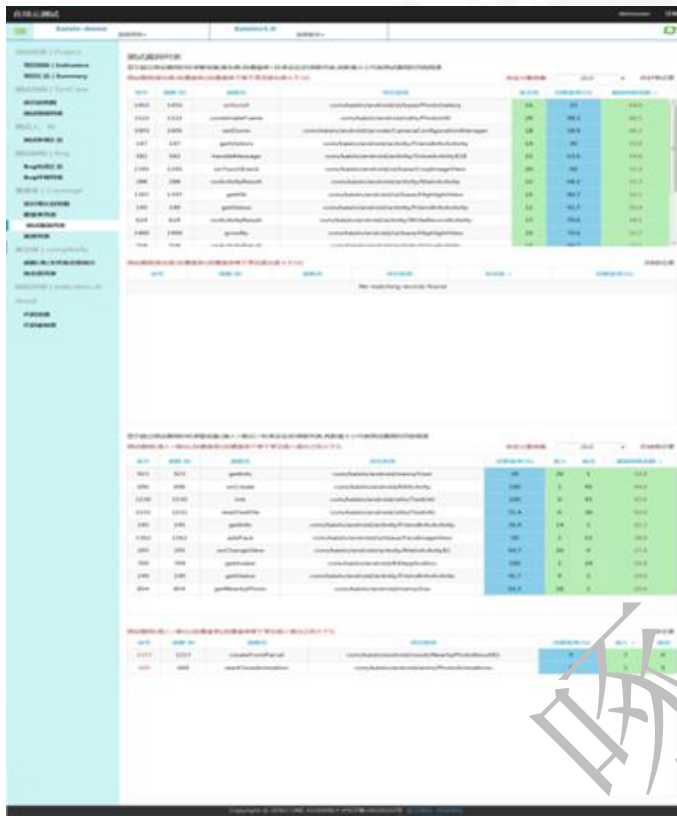
星云测试web端访问端口：3000

高级企业功能：否



视图-报表-漏洞分析

在报表中有一张报表挺值得我们注意观察的，我们发现之前做了测试用例后，该报表会按照函数复杂度与函数调用关系结合覆盖率告诉我们哪些模块测试的不足，一般这些高复杂度与多调用的模块都为核心模块，这样大大方便了我们去找那些没有测试过的高危点，可以在短期内检出更多缺陷



The screenshot displays a software testing report interface. It features a sidebar on the left with navigation options like '测试用例管理', '测试用例执行', '测试用例统计', '测试用例分析', '测试用例导出', '测试用例导入', '测试用例删除', '测试用例修改', '测试用例新增', '测试用例查询', '测试用例过滤', '测试用例排序', '测试用例导出', '测试用例导入', '测试用例删除', '测试用例修改', '测试用例新增', '测试用例查询', '测试用例过滤', '测试用例排序'. The main content area shows a table with columns for '函数名称', '函数复杂度', '函数调用关系', '覆盖率', and '测试用例数量'. The table lists various functions and their associated metrics, with some rows highlighted in green and others in red. Below the table, there are several charts and graphs, including a bar chart and a line graph, which provide visual representations of the data. The interface is designed to be user-friendly, with clear labels and intuitive navigation.

函数名称	函数复杂度	函数调用关系	覆盖率	测试用例数量
main	1000	main -> init -> process -> cleanup	100%	10
init	500	init -> process -> cleanup	100%	5
process	2000	process -> cleanup	100%	20
cleanup	100	cleanup	100%	2
test_case	100	test_case	100%	1
test_case_manager	100	test_case_manager	100%	1
test_case_executor	100	test_case_executor	100%	1
test_case_statistics	100	test_case_statistics	100%	1
test_case_analysis	100	test_case_analysis	100%	1
test_case_export	100	test_case_export	100%	1
test_case_import	100	test_case_import	100%	1
test_case_delete	100	test_case_delete	100%	1
test_case_modify	100	test_case_modify	100%	1
test_case_add	100	test_case_add	100%	1
test_case_query	100	test_case_query	100%	1
test_case_filter	100	test_case_filter	100%	1
test_case_sort	100	test_case_sort	100%	1
test_case_export	100	test_case_export	100%	1
test_case_import	100	test_case_import	100%	1
test_case_delete	100	test_case_delete	100%	1
test_case_modify	100	test_case_modify	100%	1
test_case_add	100	test_case_add	100%	1
test_case_query	100	test_case_query	100%	1
test_case_filter	100	test_case_filter	100%	1
test_case_sort	100	test_case_sort	100%	1

结束

啄木鸟软件测试培训网



Thank You

顾翔

啄木鸟软件测试培训网

QQ: 2025344

微信号: xianggu0625

Email: xianggu625@126.com

网站: www.3testing.com

微信公众号: 见二维码

