

手机测试面试题

手机应用和PC应用的区别

1、操作方式不同

PC端：键盘、鼠标 手机端：多点触摸

2、操作系统不同 PC端：Windows、MacOS、Linux、Unix 手机端：早期-塞班、黑莓、Windowsphone

现在-苹果（IOS）、谷歌（Android）

Android：碎片化-小米（MIUI）、华为（EMUI）

OPPO（ColorOS）、Vivo（FuntouchOS）、一加（氢OS）

3、手机屏幕设计不同 手机端：之前-非全面屏

现在-全面屏：异形屏-刘海屏、水滴屏、挖孔屏 真全面屏：弹拉式摄像、屏下摄像头

4、手机屏幕尺寸不同 区别：PC端屏幕一般远远大于手机端 手机端：早期：3寸、4.7寸

中期：5寸 现在：6寸、7寸

PC端：14寸、15.6寸、27寸

5、分辨率不同

手机端：早期-720P（低端） 中期-1080P（中端） 现在-2K、2K+（高端、旗舰）

6、网络类型不同

手机端：移动网络-运营商：移动、电信、联通 网络类型：2G、3G、4G、5G、WiFi

其他网络：2G-3G-GPRS（2.5G）、EDGE（2.75G） 特点--网络多样、相对不稳定

PC端：有线网络-有线水晶头连接、WiFi

特点--固定单一、稳定高速

手机测试与传统测试区别

1、网络类型多样化

2、操作系统多样化：Android（原生（5.0及以上）定制（小米不同版本测试））、IOS（8.0及以上）

3、分辨率多样化：720P、1080P、2K/2K+

手机软件测试定义

在不同的网络、不同的分辨率、不同的系统环境下执行测试软件，提升软件质量的过程。

手机软件测试流程

1、前期工作：测试计划（宏观）-->测试方案（具体）-->测试准备（数据（输入、项目）、环境（项目、工具）、技术）-->测试分析、设计用例<开发过程测试活动>

2、测试执行：不同策略-冒烟测试（monkey）、功能测试、UI测试、兼容性测试、客户端性能测试（CPU、内存、电量、启动速度）、网络类型测试、弱网测试、交叉事件测试、推送测试、稳定性测试（monkey）、服务器性能测试（jmeter/LR）-->迭代更新，回归测试（旧的缺陷、旧的模块、旧的数据）-->上线测试（客户环境下部署展开系统测试）-->对外发布 注意：手机测试重点考虑在于专项测试角度。

冒烟测试

1、目的：确认软件主体功能是否正常，是否可进行测试（可测性-能不能测）

2、方式：之前--一般以2-3资深测试人员进行功能确认--手工测试 现在--以monkey工具替代人工自动化实施测试快速覆盖确认

3、作用：monkey可以模拟用户操作（点击、滑动等）随机的发送到手机设备中进行大量的执行操作，确认软件中不稳定的因素（crash（崩溃）、anr（未响应））

4、环境资源

手机设备--真机、模拟器（夜神模拟器（首推）、手机模拟大师（第三方）、AVD(安卓虚拟设备，可以自定义创建)）。

测试项目--家具酷APP（家具电商APP）

测试工具--adb shell monkey(底层)、易测--monkey（表层）

5、环境部署

1) 手机设备--安装模拟器（夜神模拟器）-->之后--AVD 夜神安装包：nox_setup_v6.3.0.8_full.exe 安装方式：傻瓜式安装（直接操作即可） 注意：通过VT.exe检查是否支持且开启（成功标识：两个绿色对勾）

2) 测试项目--家具酷APP 安装包：测试项目-jiajuk.apk 安装方式：直接将安装包拖拽至模拟器即可

3) 测试工具--adb shell monkey、易测-monkey

易测-monkey：易测APP

易测安装包：手机测试工具-easytest2.2.4.apk

安装方式：直接拖拽至模拟器即可

adb shell monkey资源：

测试工具--Android_Test.rar(压缩包)--解压：adt4_2_32bit.rar/jdk-6-windows-i586.exe

4) 安卓工具部署 adt环境部署 环境资源：adt4_2_32bit.rar

部署方式：将adt解压至PC端盘符中（D:\tools(新建文件夹)） 文件资源：

Eclipse---->java开发工具（等同于pycharm）-->作用：提供AVD的自定义创建及使用；

Eclipse.exe：启动Eclipse开发工具

SDK----->软件开发工具包（安卓）-->包含环境支持的资源及工具

（adb/monkey/monkeyrunner）

platform-tools:平台工具文件夹-->adb.exe、aapt.exe tools:MonkeyRunner（自动化回归测试，现在不适用） SDKManager.exe-->版本更新管理程序

JDK环境部署

环境资源：jdk-6-windows-i586.exe

部署方式：安装配置即可 配置步骤：

- 1) 双击启动安装，进入引导页面，许可协议中选择“接受”；
- 2) 自定义配置：选择开发工具左侧：“第二个选项”，其他默认即可，点击“下一步”；
- 3) 自动按照配置安装完成；
- 4) 自定义配置：选择Java SE RE左侧:"第二个选项", 其他默认即可，点击“下一步”
- 5) 自动按照配置安装完成；
- 6) 结束安装：复选框取消勾选，点击“完成”。

配置环境变量

开启环境变量配置：桌面--计算机/此电脑--右键选择“属性”--系统“高级系统设置”--系统属性--高级选项卡--点击右下角“环境变量”按钮即可进入。

环境变量：系统变量（使用）、用户变量

(1) java变量配置 变量名：Path(路径)

变量值： C:\Program Files (x86)\Java\jdk1.6.0\bin

注意： win7系统添加环境变量调整位置及间隔区分

光标定位到首位： backspace（删除键）-->insert--->home (点击)；笔记本：Fn+方向键<--

间隔区分： 每个变量之间加上英文格式的分号即可。

(2) adt变量配置 变量名：Path（路径） 变量值1： D:\tools\adt4_2_32bit\sdk\platform-tools 变量值2： D:\tools\adt4_2_32bit\sdk\tools

(3) 验证测试 方式：统一cmd命令操作（win+R-->cmd） java环境测试：

java -version:获取当前的jdk版本（1.6.0） javac：编译java代码---->源程序：javac.exe java：执行代码，给出结果-->源程序：java.exe

注意：如果版本号出错，环境变量配置错了（内容、格式、位置）、cmd没有重启、配置环境变量窗口

没有确定关闭；

adt环境测试：

--adb工具测试：

adb version:获取当前adb的版本号（1.0.31）

adb :获取工具的帮助信息

--monkeyrunner工具测试：

monkeyrunner:启动monkeyrunner运行环境，包括环境版本号（jython2.5.0）、命令提示

符(>>>)

启动；

AVD安卓虚拟设备创建及使用

启动Eclipse:文件目录-->eclipse.exe(启动程序)； 首次启动配置：发送消息弹窗口：选择“No”，点击“Finish”；介绍窗口：点击左上角“AndroidIDE”--“X”（关闭后进入工作 窗口）； **操作步骤：**

打开管理窗口：点击工具栏AVDManager（手机图式）； 添加新建AVD：点击右侧按钮“New”；

参数配置：

--AVD Name： AVD名称（见名之意）--First_AVD;

--Device:设备，指的是硬件参数，包括尺寸及分辨率--3.7寸 +480X800

--Target:目标系统版本，选择的系统资源--默认4.2.2

--CPU、前后置摄像头：默认配置

--Memory Options:运行内存，按照系统要求给予配置<=768M

--Internal Storage:内置存储内存，默认即可--200M

--SD Card：存储卡，外置存储，一般不考虑。

--Emulator Options :快照、加速处理器--默认不勾选。

--完成配置：点击“OK”即可

启动AVD：选择已创建的AVD，点击右侧“start”按钮，启动设置的窗口，直接点击“Launch”

启动失败调试：

(1) 针对AVD编辑调低运行内存，760M-->500M

(2) 添加环境变量：

a、变量名： Android_SDK_Home

变量值： D:\tools\adt4_2_32bit\sdk

b、变量名： Anroid_AVD_Home

变量值： C:\Users\tedu\.android\avd

注意：如果实在无法启动，可以使用夜神模拟器操作。

adb 联调测试工具

1) adb 基本概况

本质： 属于Android\sdk\platform-tools\自带的命令控制工具 全称： adb---->Android Debug Bridge： 安卓调试桥

PC-----» adb-----» 手机 作用： PC端借助adb连接访问控制手机端（真机、模拟器（夜神、 AVD））， 进行手机端连接调试。特点：批量的多设备的管理（手机1、手机2、手机3.....）

2) adb运行原理

PC端--» adb程序--» adb服务器进程（端口号： 5037（默认） ） ----» 手机端--» 生成adb

后台进程---» 控制手机设备

3) 常用的命令

adb devices

作用：检测当前环境连接的手机设备

语法：adb devices （没有参数）

结果：

结果1：手机设备，显示设备名称，区分真机和模拟器 真机：随机十六进制数字作为androidID唯一识别，比如 131ab23e 模拟器：Emulator-5554（AVD）或者其他

结果2：连接状态。

成功---device（唯一）：连接畅通

失败---offline（还有其他）：连接离线 补充：连接其他模拟器--> 夜神模拟器（第三方）

主动连接：adb connect 127.0.0.1:62001(或者52001)

注意：adb 服务调试

命令：adb kill-server :杀掉、关闭adb服务

adb start-server:启动adb服务 其他：关闭杀毒软件，保持单一模拟器运行，在platform-tools原始目录中运行adb命令，方

式：Shift+右键菜单--> 选择在此处打开命令窗口

adb push（推送、上传） 作用：通过PC端指定文件上传至手机端指定位置 语法：adb push 参数1 参数2

参数1：PC端指定位置指定名称的文件，包括文件类型（后缀名）， 比如：D:\test1.txt

参数2：手机端指定位置指定名称，比如：/data 案例：adb push D:\1.txt /data 问题1：多于1个设备传输文件 解决：关闭其他设备，保留唯一一个设备 问题2：传输文件不存在

解决：你的文件准备有问题，原本：1.txt -->现在：1.txt.txt,修改文 件名称即可 验证测试：

进入shell命令环境：adb shell 确认当前目录位置：pwd --->/(根目录) 跳转目录：cd /data -->pwd 检查 查看目录文件：ls ---->1.txt 查看文件内容：cat 1.txt --->test 验证通过：文件名+文件内容完全相同。

练习：通过adb命令上传文件到手机 文件：D:\2.txt,内容：test2

手机：/data/data ----> 验证确认，是否上传成功 测试目的：辅助测试人员针对指定的测试数据或者测试配置文件进行 上传配置

adb pull（下载） 作用：指定手机端文件下载至PC端 语法：adb pull 参数1 参数2 参数1：手机端指定位置指定的文件 参数2：PC端指定位置指定的文件

案例：adb pull /data/1.txt D:\3.txt 问题1：无法在PC端创建3.txt进行下载 解决：a、提升权限-搜索cmd，右键选择管理员权限运行

b、提前在指定盘符创建同名的空文件，之后覆盖下载 问题2：无法找到文件或者目录

解决：文件不存在或者指定参数出错

练习：通过adb 命令下载手机端2.txt文件 保存位置：D:\4.txt 测试目的：下载手机端测试结果及运行日志，进行PC端查看分析

adb logcat（日志查看） 作用：针对手机端运行日志进行跟踪查看 形式1：adb logcat 实现：实时持续跟踪日志返回显示 结束：ctrl+C

形式2：adb logcat -d

实现：一次性读取日志返回显示

形式3：adb logcat -d >D:\logcat1.txt

实现：跟踪日志单独保存

形式4：adb logcat -d *:E >D:\logcat2.txt

实现：按照日志等级过滤筛选查看日志

级别：Error（错误级别）--> Warning（警告级别）--> info等

练习：针对计算器进行日志的跟踪查看 要求1：实时操作实时跟踪日志，保存为D:\logcat3.txt；要求2：一次性读取按照错误等级（E）过滤筛选查看日志，保存为D:\logcat4.txt 要求3：根据获取的日志进行问题分析，依据crash（崩溃）、anr（未响应）搜索错误 测试目的：根据日志查看分析测试中隐含或者遗漏的缺陷。

adb install（安装） 作用：通过命令指定安装apk程序到指定的手机设备中 语法：adb install 参数1 参数1：指定位置指定的安卓程序安装包（.apk） 案例：adb install D:\HelloWorld.apk

过程：上传文件（传输速度、文件大小、总耗时）--> 临时保存目 录--> 执行安装，给出反馈：

success（成功），其他（有问题）。

练习：针对禾禾小镇进行自动化的安装 程序：D:\hehe_V1.2.apk（安装时间：40s） 测试目的：针对被测系统进行快速的安装测试

adb uninstall（卸载） 作用：针对程序进行自动化的卸载操作 语法：adb uninstall 参数1 参数1：指定卸载的程序

问题1：指定卸载程序如何确认目标：安装包？中文？包名？

解决：利用包名指定程序--包名？ -->com.xxx.xxx（com.android.browser:浏览器包名）、

cn.xxx.xxx 问题2：如何获取程序的包名？ 方案一：跳转目录，查看文件获取所有程序包名 命令：adb shell-->进入shell命令环境

cd /data/data-->跳转目录

ls---->查看目录文件获取包名 方案二：拆包解析法

命令：aapt dump badging 安装包名称 定位：解析信息首行 package: name=‘包名’ 方案三：获取所有程序包名

命令：adb shell pm list packages

方案四：过滤筛选获取包名

命令1：adb shell pm list packages -s（system） 实现：过滤筛选系统自带的程序包名（系统预装程序）--部分 命令2：adb shell pm list packages -3（第三方） 实现：过滤筛选第三方案程序包名--部分

命令3：adb shell pm list packages hello--> 1行执行

命令4：adb shell -->pm list packages |grep hello --> 2次执行 实现：以关键内容过滤筛选获取唯一包名

补充：adb shell pm list packages |findstr hello-->cmd解析识别 方案五：启动谁获取谁

命令：adb shell-->进入shell命令环境；

dumpsys window |grep mCurrent;

定位：window{包名/活动窗口}

案例：卸载helloworld程序 包名：com.day01.helloworld

命令：adb uninstall com.day01.helloworld

测试目的：针对程序进行快速的自动化的卸载测试

练习：针对禾禾小镇进行自动化卸载 要求1：利用5种方式获取包名 要求2：针对包名指定程序进行卸载

adb shell monkey 测试

1) 冒烟测试之前与现在不同与相同之处

不同之处：

测试方式不同：之前--手工的测试；现在--自动化工具的测试 测试目标侧重点不同：之前--人工方式主要关注业务点；

现在--自动化方式主要关注软件的不稳定因素：crash

（崩溃）、anr（未响应）等 相同之处：测试方向相同：都是要确认软件的可测性，是否能够进入具体测试 阶段

2) monkey工具概况

本质：属于Android\sdk\platform-tools下自带的命令行控制工具 作用：monkey可以模拟用户操作（点击、滑动、拖拽等）随机的 发送到手机设备，大量操作执行，确认软件的稳定性 目的：确认软件的稳定性寻找不稳定因素：crash、anr等 阶段：确认软件的可测性。

3) monkey配置选项类型

-1、基本配置选项，比如：运行次数 1000

-2、运行约束选项，比如：指定包名

-3、运行事件及频率，比如：点击操作 100

-4、调试选项，比如：忽略崩溃

4) monkey常用的配置选项

-p

含义：p--->package:包 作用：指定包名，确认被测系统

语法：-p 参数1

参数1：指定程序的包名

案例1：-p com.android.browser

案例2：adb shell monkey -p com.android.browser 10

形式1：单个程序指定测试

命令：adb shell monkey -p 包名1 10

形式2：多个程序执行测试

命令：adb shell monkey -p 包名1 -p 包名2 -p 包名3 10

形式3：整个系统级执行测试 （不指定包名） 命令：adb shell monkey 10 练习：针对计算器进行monkey测试

-v

作用：控制monkey测试按照不同的等级进行日志输出，区别就在 详细程度的不同。 语法：-v

形式：

(1) -v (1个)：第一个等级，数字0标识

(2) -v -v (2个)：第二个等级，数字1标识

(3) -v -v -v (3个)：第三个等级，数字2标识 特点：数字越大，等级越高，内容越详细。

案例1：adb shell monkey -p 包名 -v 10 level 0：

命令：adb shell monkey -p com.android.calculator2 -v 10

跟踪内容：日志内容跟踪较少，只有运行启动、结束、完成等少量 信息

level 1：

命令：adb shell monkey -p com.android.calculator2 -v -v 10

跟踪内容：日志内容相对详细，跟踪到被测系统activity（活动窗口）

level 2：

命令：adb shell monkey -p com.android.calculator2 -v -v -v 10 跟踪内容：日志内容最为详细，包括选中或者未选中activity（活动 窗口） 注意：一般测试人员分析日志以第二等级及以上等级进行日志跟踪。

日志保存：adb shell monkey -p 包名 -v -v 1000 >D:\monkeylog1.txt

总结：

1) 日志内容：

a、通用信息统计，包括运行随机种子数（seed值）、运行次数（count）、运行程序（包名）、活动窗 口、操作事件及运行比重

b、运行信息统计,在第一个switch位置，向后查看，包括启动程序、操作事件、运行异常信息（有则补 充跟踪）、运行结束、运行完成。

2) 运行次数：运行次数不是指操作事件多少次（点击（down/up）、滑动），其实际所指的是操作的动 作次数，比如：事件5次，动作操作10次；其次运行次数包含启动程序。

3) 日志分析：根据关键字（crash、anr）查看定位，整理错误信息反馈

--throttle（间隔）

作用：针对操作事件进行运行时时间的间隔控制，比如点击--» 等待 1s---» 滑动 语法：--throttle 时间 单位：ms 1000ms =1s

案例：adb shell monkey -p 包名 --throttle 300 -v -v -v 20

>D:\monkeylog2.txt 练习：针对日历程序进行monkey测试，并跟踪日志。 思路：获取包名--» 设计monkey命令--» 执行测试，并分析日志

-s

含义：-s代表seed,指代随机种子数 作用：seed值可以控制多次随机（伪随机）运行保持相同的逻辑 目的：保证测试中缺陷重现及返测

>D:\monkeylog4.txt

区分：

第一次：adb shell monkey -p com.android.browser -s 10 --throttle 500

-v -v -v 40 >D:\monkeylog4.txt

执行逻辑：光标向下移动--» 曲线滑动--» 直线滑动--» 曲线滑动

第二次：adb shell monkey -p com.android.browser -s 10 --throttle

500 -v -v -v 40 >D:\monkeylog5.txt

执行逻辑：光标向下移动--» 曲线滑动--» 直线滑动--» 曲线滑动

第三次：adb shell monkey -p com.android.browser -s 11 --throttle

500 -v -v -v 40 >D:\monkeylog6.txt 执行逻辑：点击菜单键--» 直线滑动--» 直线滑动--» 点击操作 总结：随机种子数相同，则运行逻辑相同，反之则不同。

--ignore (调试选项) 作用：针对monkey测试中遇到的异常情况（anr、crash）进行规避，保证完整有效的测试执行。选项：--ignore-crashes ---->忽略崩溃

--ignore-timeouts--->忽略超时（未响应）

--ignore-security-exceptions--->忽略安全异常

--ignore-native-crashes-->忽略本地崩溃

--monitor-native-crashes-->监控本地崩溃代码

案例：adb shell monkey -p com.android.browser -s 15 --throttle 600 -- ignore-crashes --ignore-timeouts --ignore-security-exceptions --ignore- native-crashes --monitor-native-crashes -v -v -v 50 >D:\monkeylog7.txt

--pct 作用：指定测试的操作事件及运行频率（百分比） 语法：--pct-事件类型 运行频率

案例1：--pct-touch 50 （点击操作运行比重是50%）

案例2：adb shell monkey -p 包名 --pct-touch 60 --pct- motion 20 -s 13

--throttle 500 -v -v -v 500 >D: \monkeylog8.txt

练习：针对家具酷进行monkey冒烟测试

要求1：测试机---» 夜神模拟器，连接设备

命令：adb connect 127.0.0.1:62001(或52001)

要求2：测试项目--» 获取家具酷包名（3种） 命令：adb install D:\jiajuk.apk

要求3：设计monkey命令：-p --pct -s --throttle --ignore -v 运行次数、另存>

要求4：运行次数10000

问题：家具酷程序的包名？ ---» adb shell pm list packages :包名-com.ulang.jjk

命令：adb shell monkey -p com.ulang.jjk --pct-touch 60 --pct-motion 20 -- pct-majornav 10 --pct-anyevent 10 -s 12 --throttle 500 --ignore-crashes -

--ignore-timeouts --ignore-security-exceptions --ignore-native-crashes -- monitor-native-crashes -v -v -v 10000 >D:\jiajuklog.txt

要求5：跟踪日志后分析问题（anr、crash）

总结：

1) 根据关键字查找分析日志，关键字包括：anr、crash、exception（异常）、GC（内存异常：内存溢出--GC_Error）、force_closed（强制退出）

2) 异常情况分析

陌匠APP异常分析

思路：

a、根据logcat.log，以关键字查找错误位置，确认基本信息情况，包括：PID、包名、错误基本信息、 基本原因。

陌匠APP：问题类型--anr：

--PID-2866、包名-cn.idcby.jiajubang；

--错误信息-Input dispatching timed out

--基本原因-窗口没有加载出来，（比如：在空白屏中点击没有任何业务效果，因为系统前台操作的限制时间是5s。）

b、根据traces.txt（内存跟踪信息）

简单--根据PID（2866）确认错误信息区间，整体反馈开发即可；

开始：PID 2866--» 中间：信息区间--» 结束：end 2886（end：结束）

复杂--从end 2866结束的位置向上查找，依据错误关键字：waiting（正在等待）、locked（锁定、死 锁）

本质：窗口加载不出--» 就是有其他进程的干扰，很多进程排队等待，处理超时了。 错误信息：

"ReferenceQueueDaemon" daemon prio=5 tid=4 Waiting

| group="system" sCount=1 dsCount=0 obj=0x12c07100 self=0xb3c32800

| sysTid=2878 nice=0 cgrp=default sched=0/0 handle=0xb3c52800

| state=S schedstat=(10216019 6686311 100) utm=1 stm=0 core=0 HZ=100

| stack=0xaf1da000-0xaf1dc000 stackSize=1036KB

| held mutexes=

at java.lang.Object.wait!(Native method)

- waiting on <0x28cd1964> (a java.lang.Class<java.lang.ref.ReferenceQueue>)

at java.lang.Daemons\$ReferenceQueueDaemon.run(Daemons.java:134)

- locked <0x28cd1964> (a java.lang.Class<java.lang.ref.ReferenceQueue>)

at java.lang.Thread.run(Thread.java:818)

===== 反馈结果： 形式：以Excel表反馈结果，包括：测试阶段、测试项目（中文、包名）、测试工具、测试命令、测试事

件 及运行频率、测试次数、测试结果（Passed、Failed）、缺陷提交（问题及频率）、备注说明（告知 错误信息）

主流手机操作系统

1、常见的操作系统：Android、iOS、塞班、黑莓、windowsphone

2、主流系统：Android、iOS

3、Android：

（1）公司：谷歌-07年11月5日发布

（2）系统内核：linux系统

（3）系统架构：操作系统、用户界面、中间件、应用软件

（4）特点：免费开源、系统碎片化

（5）主流手机厂商： 台湾：HTC 韩国：三星

国内：华为、OPPO、Vivo、魅族、酷派、小辣椒、美图、小米

补充：以糖果名字对每一个版本命名：8.0-奥利奥 9.0-pie 派， 最新版本：9.0 10.0（研发最新版本）

4、IOS

（1）公司：苹果，专门针对iPhone手机

（2）其它设备：iPad、ipod iTouch、iwatch、appleTV

（3）ios版本：IOS12、12.3.1、公测版（ios12.4）、开发者预览版

手机测试的分类

1、传统的手机测试 a、关注对象：手机硬件本身 b、测试方式：抗摔、抗压、抗疲劳、抗高温测试，包括手机系统的基本的功能和性能的测试

2、应用软件的测试 a、针对对象包括手机系统及应用软件 b、针对软件的功能及性能的测试考虑，包括：兼容测试、功能测试、性能测试、内存占用测试等

手机测试与传统测试的区别

1、网络的多样化：2G-4G

2、系统的多样化：Android、IOS、Windowsphone

3、分辨率的多样化： a、低端：720P b、中端：1080P c、高端：2K、4K

手机软件测试的定义

定义：在不同网络环境、不同系统环境、不同的硬件分辨率支持下执行软件的过程，最终发现缺陷提升软件质量。

量。

手机测试流程

开发人员：程序打包-发布测试版本

测试人员：冒烟测试--» monkey--随机测试--确定软件稳定性。

测试人员：UI测试（界面）--RP/效果图，其次针对性做更多扩展的UI测试（界面美观性、界面规范性等） 测试人员： 功能测试（业务）--需求文档;需求分析--» 场景罗列--» 用例设计--» 用例执行 测试人员：兼容性测试（软件、硬件）、性能压力测试（服务器端、客户端） 测试人员：正式环境测试（试运行测试）：真实用户

环境、真实用户数据等 测试人员：a、后台用户数据统计测试 b、用户行为统计测试

测试人员：回归测试-- a、回归旧的模块功能 b、回归缺陷 c、回归旧的数据

发送上线报告--项目负责人--» 签字确认后对外发布。

手机软件安装测试如何进行

（1）不同渠道包下载后安装是否正常，包括：第三方应用商店（应用宝、百度手机助手、91）、本地应用商店（小米、华为）

（2）在不同系统下安装程序是否正常 注意：不同的定制系统（EMUI、MIUI、ColorOS等）、不同Android（5.0左右）系统

（3）安装后是否正常启动，安装后是否按照指定目录写入文件。

（4）安装过程中各个选项组合是否符合概要设计 注意：选项组合排版布局、按钮默认状态，开启关闭后的状态实现。

（5）安装过程中是否支持取消，取消后是否正常实现，回滚业务及数据。 注意：取消安装时关注业务逻辑实现以及数据实现情况。

（6）安装过程异常情况测试，确认是否正常 a、内部：程序安装过程中是否有崩溃、闪退、未响应等异常 b、外部：外部环境未响应（微信卡死）、断电、重启

（7）安装过程中存储空间不足是否及时给出信息提示 注意：信息提示是否正常，信息提示后续处理是否正常

（8）安装后配置文件是否有多余或者遗漏的文件结构及文件。

（9）安装手册测试，确认是否逻辑及数据同步。 注意：如果有问题，确认是否是软件问题，还是文档问题。

（10）安装过程中需要网络验证，断开网络后确认反馈是否正常。

安装流程：双击安装--流程引导--安装页面（取消、安装）--安装流程页面-完成后（结果页面：完成、打 开）

手机卸载测试测试点有哪些

本质：删除数据、图标、软件主体。

（1）删除安装后生成配置文件是否有相关的提示。

（2）卸载触发后是否给出卸载确认的提示。

（3）卸载后相关配置文件和缓存数据是否删除

（4）卸载时异常情况测试是否正常 a、内部：软件本身在卸载时是否有崩溃、闪退、重启的情况。 b、外部：外部软件及系统异常、异常环境卸载被测系统是否正

常。系统或软件：未响应、崩溃

（5）卸载时是否支持取消，取消后是否正常（业务、数据）

（6）卸载过程中UI测试是否正常 卸载开始-卸载过程-卸载结束

注意： UI中图片、文字、按钮状态、图标等是否与当前的业务逻辑一致。

(7) 不同系统卸载

(8) 功能按键与卸载逻辑交互正常 a、back键取消卸载 b、卸载时菜单键前后台切换是否正常 c、卸载时后台直接kill进程 缺陷：搜狗听写卸载时前后切换后软件管理进程崩溃

手机UI测试测试点有哪些

定义：针对用户界面（窗口、对话框、控件（按钮、下拉列表））、文字、图片等对象进行测试，考虑界面的美观、布局的协调。

(1) 导航测试 导航的分类：标签式导航（音乐类型软件）、抽屉式导航（QQ）、宫格式导航（美图秀秀）、列表式导航

（微信-我的模块（个人中心））、悬浮图标导航（好奇心日报） 测试点：

a、各个页面衔接是否需要导航 b、导航是否方便直接，结构直观明了。 c、导航系统是否需要加入搜索引擎 d、导航是否准确，各连接页之间是否在切换后保持统一页面风格。

(2) 图形测试

a、横向比较，各个控件操作方式是否保持统一 包括滑动、点击 注意：操作后页面显示形式也要保持统一，比如统一向左翻转显示。 b、自适应界面设计，确认横竖屏切换内容是否同步协调展示

(1) 自适应设计模式：一对一设计实现，比如：4.7寸，直接设计4.7模板；如果5.1寸，直接设计5.1模板。

(2) 数据加载设计模式，一堆多，比如：4.7寸--» 少加载一些内容；5.1寸--» 多加载一些内容

c、页面标签风格是否保持统一 注意：颜色不能过分使用，重点考虑页面的一致性。 d、页面是否美观 e、页面的图片应有其实际的意义，且整体排布有序合理 f、页面的图片质量及尺寸符合要求的情况下尽量选择小一些。 g、页面颜色不宜使用过多

注意：页面颜色对比度不要过高，否则非常刺眼。其次结合业务去考虑颜色使用，来区分业务层次

(3) 内容测试 本质：针对即是软件中的文字内容 a、文本框中说明文字是否与其业务保持一致。

b、文字长度是否加以限制 c、文字表意是否有不明。 d、文字中是否有错别字。 e、文字是否以中文显示

注意：区分对待不同软件定位的市场区域，包括不同国家、不同民族等等，其次还应注意语言语法，比如英文复数，注意书写顺序，藏文（从上到下），阿拉伯文字（从右到左）

补充：国际化测试--针对海外发布软件，需要做国际化语言转化；

本地化测试--其他国家软件进入本国区域发行，就需要针对本地语言进行转化，包括语言习惯用语等。 f、文字中是否有敏感性词汇，包括：血腥、暴力等 g、图片中是否有敏感性内容，包括涉及版权、涉及专利、隐私等。

手机功能测试策略

a、运行APP

(1) 安装后试运行，确认启动是否正常 启动逻辑：点击开始--启动页--进入主页 启动页：官方默认（logo显示及文化标签）、活动、节日主题推广、广告推广（默认逻辑（自动跳转主 页）、手动逻辑（手动跳过进入主页）） b、打开测试中对于有数据加载情况，确认是否有加载状态及进度提示。 注意：默认数据加载状态，更新数据加载状态。

c、启动被测系统，确认启动速度是否正常。 注意：是否有过慢、甚至黑屏情况。 d、APP页面交互逻辑及流畅性是否正常 逻辑：业务加载页面是否正确，比如：

小说--未登录（没有用户）--书城页面（推荐数据） 小说--已登录（有用户数据）

--书架页面（供用户继续浏览上一次数据） 流畅：页面切换速度是否较快。 e、注册是否正常

1) 注册方式：手机号注册、账号密码注册、第三方软件授权注册

2) 注册时同表单编辑是否正常：数据错误的情况以及数据全部正确的情况

3) 账号密码的长度测试 补充：账号密码内容类型，比如：必须包含大写字母、小写字母、数字、下划线等等

4) 前台注册后是否给予正常反馈(信息提示)

5) 注册后前后台数据是否保持统一

• 比如：前台：张三，后台：张三

• 缺陷：前台：李四，后台：李四 二次注册：李四，把第一次注册用户给覆盖

6) 前台注册后，后台是否正常给出信息提示

7) 登录测试

8) 注销测试

f、前后台交互对于软件功能的影响

1) APP切换前后台操作后，确认软件的停留页面是否上一次操作页面

2) APP前后台切换后，确认是否保持历史的功能及状态的正常实现

3) APP前后台切换后，确认是否有崩溃、闪退或者功能异常。

4) 手机锁屏后解锁，确定被测系统是否功能及状态正常，是否有崩溃的情况

5) 手机接听电话后挂断，回到APP，确认是否功能及状态正常，且无崩溃等异常缺陷

6) APP切换至后台kill(杀掉、关闭)进程，确认APP重启后数据、业务是否正常实现。

7) 信息提示框弹出后，确认前后台切换是否影响功能实现及状态。异常情况包括：消失（跳过）、位置发生 偏离等。

8) APP数据加载过程中，前后台切换确认是否出现崩溃、闪退、空白页、重启等异常情况 注意：数据加载过程中页面非常不稳定，极易发生缺陷，而且一旦问题产生严重程度非常高。 g、免登录

定义：指的是登录一次后，不主动退出时，后续一直保持登录状态。

1) 免登录功能确定在不同版本中是否正常

(1) 新版本新功能新结构对原始的免登录功能影响。主要考虑版本差异

(2) 不同版本之间覆盖更新后用户数据及状态的传递是否正常

2) 无网后确认免登录功能及状态是否正常 注意：网络的状态是独立，最终不会影响免登录的状态，需要分别考虑，关注断网后APP的反馈，及断网重连 后用户相关的信息及数据的同步是否正常

- 3) 切换用户登录，确认免登录功能是否以最新用户为准，同步更新用户信息及业务数据，其次原用户正常退出。
- 4) 免登录时，确认异地登录多台手机，是否正常给予反馈，强制退出且给出信息提示。注意：异地登录限制
- 5) 免登录时，确认前后台切换安全校验是否正常
- 6) 免登录时，确认前后台切换数据同步是否正常
- 7) 更改用户密码，确认免登录处理逻辑是否正常。
- 8) 自动登录与免登录功能的交互是否正常，自动登录后，是否由免登录逻辑正常处理。
- 9) 退出登录后，确认免登录是否正常，重启APP后确认是否停留在登录页面

手机数据更新测试

- 1) 确定被测系统哪些模块是自动刷新，哪些模块是手动刷新，或者自动+手动。
- 2) 确定哪些模块需要后台数据更新（前后台切换）且实现是否正常。
- 3) 确定被测系统哪些模块是否定时更新，哪些模块是实时更新，或者定时+实时
- 4) 确定被测系统如何获取数据，数据获取的交互逻辑是否正常 a、客户端（发起请求）--本地缓存（有数据）--无需获取数据（服务器） b、客户端（发起请求）--本地缓存（没有数据）--获取数据（服务器） c、客户端（发起请求）--本地缓存（部分数据）--获取另外的数据（服务器）

- 5) 确定被测系统任意模块有数据加载的情况，其交互逻辑及状态是否正常。注意：数据更新过程，页面不稳定，需要特别关注。

手机离线浏览

- 1) 无网络时确定离线浏览是否正常，包括数据内容及界面排布
- 2) back键退出后重启APP，确认离线浏览是否正常
- 3) APP切换后台后，返回前台，确认离线数据加载是否正常
- 4) 锁屏后解锁，确定APP离线浏览是否正常 注意：锁屏后解锁，手机状态变化，交互逻辑较多，重点关注。
- 5) 离线数据有更新时，确认是否及时推送更新提示且可正常更新使用

APP更新测试

- 1) APP有新版本时，确认是否及时推送更新提示
- 2) APP非强制更新时，确认取消后，可正常使用旧版本，且继续推送更新提示。
- 3) APP强制更新时，确认是否不更新不能使用，重新启动后继续推送更新提示。
- 4) APP覆盖更新（不删除本地客户端），确认更新是否正常
- 5) APP覆盖更新后，确认功能实现是否以最新版本需求考虑。
- 6) APP更新后对同名资源文件（文件及图片等）是否能够以最新版本业务逻辑及布局结构正常调用展示。扩展：更新时与系统交互，包括第三方或者本地软件的交互，其次包括系统及三大主键交互

相机、定位服务测试

- 1) 相机、定位服务在不同版本差异变化中，确认功能实现是否正常。注意：版本的变化中会涉及模块层次、逻辑关系。
 - 2) 相机、定位服务使用过程中，确认前后台切换功能是否正常 前台：相机（软件+硬件）正常工作 后台：相机（软件+硬件）暂停状态
- 底层交互：应用--系统--硬件驱动--硬件
- 3) 若定位服务未开启时，确认程序是否给出友好性提示，且能通过其设置开启服务，返回后功能正常实现
 - 4) 定位、照相机服务，须采用真机进行测试。

时间测试

主要考虑是手机时区、时间的改变对APP的影响

案例：比如发布一条微博 中国：10:00--发布微博 美国：22:00--显示该条微博

push（推送）测试

- 1) 设置不接收推送，确认用户是否无法正常不接收消息

注意：软件内部的控制--新消息通知设置：关闭 系统及的外部控制--通知中心：关闭“允许通知”

- 2) 手机免打扰设置时，确定在免打扰时间段是否正常屏蔽消息，反之是否正常接收消息
 - 3) 若消息推送有用户身份的考虑，确认消息推送是否与用户身份相匹配，其次发送消息以最后登录的用户为准，确认是否正常
- 总结：身份与消息是否对等匹配

- 4) 推送测试须进行真机测试。

手机性能测试

a、极限测试

注意：主要考虑是环境（硬件、网络）极端压力情况下APP是否能够正常处理。包括：存储、电量、网络等情况处于极端压力范围

比如：内存满了安装APP 使用APP时突然断电 使用APP时突然断网

b、响应能力测试 针对的是客户端的各种操作行为最终反馈时间是否符合用户的要求。比如：APP的安装、卸载是否时间消耗符合要求

APP的各项操作符合要求：搜狗听写-记事写作的查看速度。c、压力测试 指的是长时间的、反复大量对软件进行操作确认是否正常运行 比如：反复安装、卸载确认是否异常

反复操作各项功能，确认是否异常。d、性能评估 主要通过性能工具包括：LR、jmeter等对用户进行大量模拟，最终以此用户量访问服务器，确定服务器压力 反应是否符合要求。

补充：系统架构==>就一个客户端就可以干成任何事情？==>不对。前端：客户端-->通过各种语言渲染有序准确的显示数据 中间：中间件（应用服务器）

==>主要是数据交互控制，包括：Apache、tomcat. 后端：服务器（数据库管理系统）==>SQLserver/mysql/oracle/db2等 e、基准测试

主要是通过对比市场中同类软件，来确定其开发标准。

- 1) 业务逻辑路径的复杂度 软件1：登录：3步 软件2：登录：2步或者1步就完成

2) 安全性是否覆盖全面 密码是否暗文显示。

monkey测试是什么?

主要是Android系统自带的自动化测试工具

monkey测试的实现方式

通过模拟用户在手机中的操作行为：滑动、触摸、系统操作等，对被测系统进行随机的操作

monkey的测试结果

主要是发现软件中隐含的不稳定的因素，包括crash（崩溃）、ANR（未响应）等情况

如何处理monkey测试结果

主要是获取并整理崩溃的相关错误信息，通过测试获取的运行日志搜索定位崩溃的错误信息，整理后反馈给开发

发人员即可

monkey测试的测试目的

主要测试软件的稳定性，暴露其中崩溃的情况。--又叫做稳定测试、健壮性测试。

monkey的指令类型

- (1) 基本操作选项：比如运行次数。
- (2) 约束选项：比如包名指定，事件之间时间缓冲
- (3) 事件类型和运行频率：比如touch（触摸、点击），motion（滑动），syskeys(系统级事件，菜单、功能、home等)
- (4) 调试选项：比如ignore（忽略），ignore_crashes/ignore_timeout

monkey常用的基本命令

-p：指定运行程序包名（packge） 语法：-p 包名

案例：-p com.baidu.browser

实际：

adb shell monkey -p com.baidu.browser 100

-v:日志详细记录等级 等级：3种等级：

第一个等级-编号是0，书写形式：-v,记录基本的运行开始及过程和结果的信息。第二个等级-编号是1，

书写形式：-v-v，记录指定程序响应的所有内容 第三个等级-编号是2，

书写形式：-v-v-v，记录指定响应的或未响应的所有的activity(活动)

jdk配置

- (1) JAVA_HOME

变量值：jdk安装路径；

- (2) Path 变量值：jdk安装目录\bin； 注意：变量值最后添加英文格式“;”作用：提供java开发工具支持比如：java/javac

- (3) CLASSPATH

变量值：.jdk安装目录\lib\dt.jar;jdk安装目录下\lib\tools.jar;作用：寻找定位编译后的类文件，先在Windows本地目录查找，找不到再分别去jdk资源包查看，包括dt.jar 以及tools.jar.

(4)验证环境变量配置

1)java -version ,可以显示Java版本

2) java ,正常执行反馈结果

3) javac,编译工具

AVD创建

1) 基于Eclipse创建管理，注意如果启动有问题，主要是环境配置原因造成

2) 创建步骤 a、在SDK\eclipse目录下双击“Eclipse.exe”程序启动 b、弹出提示框选择“Cancle”关闭，其次点击“Android IDE”-“X”号关闭窗口即可进入主窗口。 c、工具栏上点击选择“Android Virtual Device Manager” d、进入管理页面点击“New”按钮。

e、进入配置页面进行一下配置

1) AVD Name:虚拟系统的名称--见名知意

2) device：选择系统的硬件标准，包括设备尺寸及分辨率

3) Tagert：指定系统版本，可离线下载或在线下载

4) 其他配置：包括前置摄像头，后置摄像头，存储内存，运行内存，外置存储（SD card）等 注意：谷歌对部分系统要求的运行内存不能操作1GB，若超过后可能无法启动，故选择默认运行内存配置启动。

f、点击“确定”后创建成功，选择虚拟系统，点击“start”按钮启动虚拟系统 变量值1：D:\tools\adt4_2_32bit\sdk\platform-tools;

变量值2:

D:\tools\adt4_2_32bit\sdk\tools;

Android环境变量配置

(1)sdk下platform-tools文件夹：提供adb相关的工具及资源的支持。

(2) sdk下tools文件夹：提供monkeyrunner工具及资源支持。

(3) 配置变量 变量名：path 变量值：D:\tools\adt4_2_32bit\sdk\platform-tools;

D:\tools\adt4_2_32bit\sdk\tools;

(4) 验证配置

1) adb：会获取版本信息及执行信息

2) adb version:获取adb版本信息

adb工具

- 1) adb是通过访问、控制手机、模拟器进行指定命令下自动化的关联调试==调试测试
- 2) adb==Android debug bridge（安卓调试桥）
- 3) 本质：Android sdk下platform-tools文件夹下支持工具。
- 4) 基本操作：adb devices ==获取连接的设备及状态 深层操作：adb shell ,进入Linux命令环境。
- 5) 常用操作命令

（1）adb devices:获取连接设备及状态，包括模拟器、物理真机 获取模拟器：

名称：emulate+端口（5554） 手机真机：

名称：显示Android ID

设备状态：device=已就绪；offline=离线状态。

（2）adb push:从PC端推送、上传文件到指定手机端

语法：adb push 指定PC端目录下的文件 指定手机端保存文件目录、位置

案例：adb push D:\1.txt /data/

检查是否上传成功：

- 1) adb shell :进入Linux命令环境
 - 2) 当前所在位置：pwd=/(根目录下)
 - 3) 跳转目录：cd /data/
 - 4) 当前所在位置：pwd =/data
 - 5)检查文件夹所属文件：
- ls ==1.txt(说明上传成功)

（3）adb pull:从手机端指定目录下指定文件下载到保存到PC端指定目录指定文件。注意：主要用于日志文件的下载，进行日志记录及分析。

语法：adb pull 手机端指定文件 PC保存目录及文件名称 案例：adb pull /data/1.txt d:\2.txt 检查确认：在指定目录打开查看确认。

adb logcat:查看当前系统日志。

a、adb logcat >D:\app1.log

持续跟踪记录日志且重定向保存日志内容。 b、adb logcat -d >D:\app2.log 一次读取完日志并重定向保存日志内容。

adb install:从PC端指定目录下指定文件上传安装至手机端。

语法：adb install PC端指定位置指定文件名称+文件格式

adb uninstall

作用：指定手机设备中某一程序进行卸载。

语法：adb uninstall 程序的包名

aapt dump badging 程序安装包

作用：通过拆包解析获取包名，识别位置：package:name='包名'

获取HelloWorld.apk的包名：

aapt dump badging D:\HelloWorld.apk

包名即：com.day01.helloworld

对禾禾小镇程序进行自动化卸载。

- 1) 获取包名
- aapt dump badging D:\ hehe_v1.2.apk
- 包名：com.zmhj.hehe
- 2) 卸载程序
- adb uninstall com.zmhj.hehe

3) 检查确认

错误情况：

adb shell

作用：直接进入访问手机底层命令环境（Linux命令环境），可以直接使用Linux命令对底层进行各种操作，包括基本文件的操作、系统信息查看等等 语法：adb shell 回车即可

环境提示：

root:超级用户、管理员

@Android：是由Android系统进入shell命令窗口

\#：标识当前是Linux命令环境 环境退出：ctrl+c 注意：区别monkeyrunner结束环境的操作：ctrl+d

adb shell常用的命令

- 1) pwd:获取当前所在位置或目录
 - 2) ls:获取文件夹中所有的文件内容
- ls -l:获取文件内容及所属权限、所属用户、文件目录结构等信息。
- 3) date:获取手机设备的系统日期和时间。
 - 4) cat /proc/meminfo：
- 获取手机设备的内存信息
- 5) cat /proc/cpuinfo:
- 获取手机设备的CPU信息
- 6) ps :获取系统进程信息，包括进程名、PID等

7) |grep:过滤筛选查询内容

monkey常用命令

-s

(1)作用：指的是monkey运行的随机种子数，以此数据可以控制每次运行的事件逻辑。

(2)语法：-s 随机种子数（任意的指定数值，不要用小数数据。）

(3)案例：

第一次测试：adb shell monkey -p com.day01.helloworld -s 15 100

结果：1、点击；2、滑动，3、系统事件：back.....

第二次运行：

adb shell monkey -p com.day01.helloworld -s 15 100

结果：1、点击，2、滑动，

--throttle

(1)作用：主要是为事件操作进行时间的间隔缓冲

(2)--throttle 时间（单位：毫秒）

(3)adb shell monkey -p com.day01.helloworld -s 14 --throttle 300 -v -v 50

--pct

(1)作用：指定被测系统运行的事件类型以及运行频率（百分比）

(2)语法：--pct 事件类型 运行频率

(3)案例：

adb shell monkey -p

com.day01.helloworld --pct touch 40 --pct motion 30 --pct syskeys 20 -s 25 -- throttle 350 -v -v 1000

注意：

-s 随机种子数可以不指定，系统会默认给定，可通过日志查看；--pct 事件类型不指定时系统会自动分配

--ignore-crashes

(1)作用：忽略崩溃情况，继续保持完整的、准确。稳定的随机测试

(2)语法：--ignore-异常情况，包括：crashes/timeout/exception

(3)案例：

adb shell monkey -p

com.day01.helloworld --pct touch 40 --pct motion 30 --pct syskeys 20 -s 25 -- throttle 350

--ignore-crashes --ignore-timeouts -v -v 1000 >D:\helloworld.log adb shell monkey -p

com.day01.helloworld --pct touch 40 -s 25 --throttle 350

--ignore-crashes --ignore-timeouts -v -v 1000 >D:\helloworld.log

包名：com.zmhj.hehe

日志分析

a、日志内容类型：

- (1)随机种子和运行总次数
- (2)运行程序包名
- (3)运行过程中事件操作后的日志
- (4)事件的内容及覆盖比重

b、错误信息定位

- (1)ctrl+F 打开搜索
- (2)定位关键字：分别输入：ANR、Exception、crash、GC进行相关内容搜索 注意：ANR--未响应的缺陷
- crash--崩溃的缺陷
- GC--内存异常情况
- exception--异常情况，比如：空指针异常--NullPointerException
- (3)定位关键字后上下文分析定位错误信息，整理该信息反馈给开发人员。

手机真机测试

环境配置

- (1)通过手机数据线连接手机端和PC端
- (2)匹配下载合适的手机驱动进行安装
- (3)通过adb客户端启动后借助adb服务器（5037端口指定进程）向手机端启动adb后台进程进行访问通讯控制。

monkeyrunner

- (1)本质：Android-sdk目录下自带的工具文件
- (2)运行原理：手动测试：操作动作+逻辑==》测试人员 自动化测试

操作动作（monkeyrunner）+逻辑（编程语言、脚本语言==Python）

- (3)测试框架 案例：修理工-修理家具

修理工--工具箱--monkeyrunner(三大组件) 修理工--python编程语言 修理工利用工具按照有序逻辑完成修改--》

Python有序调取工具：MonkeyRunner-->MonkeyDevice-->MonkeyImage

- (4)运行方式 方式一：逐行解释执行 方式二：python脚本批量执行 案例：monkeyrunner

D:\first_py.py 注意：实际工作中为了自动化持续实施和更新维护，一般采用单独Python文件进行编辑管理，最终整体调用 执行。

(5) 常用API（工具方法）

MonkeyRunner(工具类)

waitForConnection

返回类型：MonkeyDevice

方法名：waitForConnection 注意：方法名首个单词的首字母小写，后面的单词的首字母大写 参数：确认方法输入参数，包括参数的个数、参数类型、参数顺序。

float timeout:指定连接设备的超时时间，float:数据类型--浮点类型（小数类型） String deviceId:连接设备的ID，String：字符串类型 作用：连接指定的某一设备，超过超时时间后，终止连接。

sleep

返回值类型：void

-无需返回结果值（无返回值类型） 方法名：sleep

参数：float--浮点类型（小数），时间单位：s(秒) 作用：等待一定的时间后再继续执行。

MonkeyDevice

1)常用的常量

-DOWN: 触摸（按下） 数据类型：String（字符串类型）

-UP: 抬起 数据类型：String

-DOWN_AND_UP:按下后抬起 注意：对于常量所有的字母都需要大写。

2) 常用的方法

-wake:点亮屏幕 返回值类型：void（无返回值类型） 方法名：wake

参数：空参--（） 作用：对手机设备点亮屏幕的操作

-type: 类型--> 输入 返回值类型：void(无返回值类型) 方法名：type

参数：类型-String,message:代指输入字符串 作用：对手机设备文本框输入指定内容

案例：device.type('10086')

结果：对手机设备device指定文本框输入（动作）10086（参数）

-touch: 点击、触摸 返回值类型：void(无返回值) 方法名：touch

参数：坐标位置-X, Y, 类型：integer（整数类型） 其次指定行为逻辑-‘DOWN_AND_UP’指定常量

案例：

device.touch(155,235,'DOWN_AND_UP')

结果：对指定手机设置中x像素坐标：155, Y像素坐标：235位置点击一下，逻辑形式：“按下后抬起”

-drag:滑动、拖拽 返回值类型：void(无返回值类型) 方法名：drag

参数：坐标位置开始和结束，类型整形（integer），duration:间隔时间，默认1s,步骤插值：默认10 案例：device.drag(155,235,155,245)

注意：滑动形式-向上、向下（y轴变化），向左、向右（X轴变化）

长按操作如何实现？

-解决方案：基于drag方法实现。只要不改变开始和结束坐标位置，即可实现长按。结果：device.drag(155,255,155,255,2,10)

-press:主要考虑手机系统级操作形式，包括back/home/menu/拨号键/光标移动。返回值类型：void(无返回值类型)

方法名：press 参数：1、操作的按键名称，2、操作逻辑形式 作用：通过press方法实现系统级操作行为。操作形式：

-KEYCODE_HOME:

指代的就是home键的操作，具体形式：device.press('KEYCODE_HOME', 'DOWN_AND_UP');

-KEYCODE_MENU:指代是功能键或菜单键的操作，具体形式：device.press('KEYCODE_MENU','DOWN_AND_UP');

-KEYCODE_BACK:指代是手机中back键，返回上一级页面或者退出系统。具体形式：device.press('KEYCODE_BACK','DOWN_AND_UP');

-KEYCODE_CALL:指代就是手机中拨号键，具体形式：

device.press('KEYCODE_CALL','DOWN_AND_UP') -KEYCODE_DPAD_UP:

指代就是光标向上移动

-KEYCODE_DPAD_DOWN:指代就是光标向下移动

-KEYCODE_DPAD_RIGHT:指代就是光标向右移动

-KEYCODE_DPAD_LEFT:指代就是光标向左移动 具体形式：

device.press('KEYCODE_DPAD_UP','DOWN_AND_UP')

-takeSnapshot:获取截图 返回值类型：MokeyImage-图片类型 方法名：takeSnapshot 参数：空参的形式。（）

作用：通过此方法对执行过程中APP的操作界面进行截图处理，最终返回指向某一变量。具体形式：

device.takeSnapshot()

writeToFile:写入文件（保存文件）

返回值类型：void(无返回值) 方法名：writeToFile 参数：path-保存路径 类型：String（字符串类型）； format-保存格式，类型：String（字符串类型）

作用：通过指定截图文件对其进行保存本地的处理。具体形式：result.writeToFile('D:\testImage.png','png')

sameAs: 相似或者对比相同的情况

返回值类型：boolean(布尔类型)，逻辑判断的类型：False/Ture

方法名：sameAs

参数：比较的图片，类型：MokeyImage;百分比差异值-相似的百分比程度，类型：float。

MonkeyRecorder(录制工具类型)

a、固定代码形式：

第一行：导入MonkeyRunner工具类型 代码形式：

```
from com.android.monkeyrunner import MonkeyRunner
```

第二行：导入MonkeyRecorder

代码形式：

```
from com.android.monkeyrunner.recorder import MonkeyRecorder
```

第三行：连接设备 代码形式：

```
device= MonkeyRunner.waitForconection()
```

第四行：启动录制

```
MonkeyRecorder.start(device)
```

微信发红包的脚本达内讲师群发送1个红包，拆分成10个，总金额是：10元。

完成第一步：

思考微信发红包业务逻辑。完成第二步： 整理思路，对每一步进行代码转化

（1）导入三大组件

```
from com.android.monkeyrunner import MonkeyRunner from com.android.monkeyrunner import MonkeyDevice from com.android.monkeyrunner import MonkeyImage
```

（2）控制台打印‘开始’

```
print 'start'
```

（3）连接手机设备 device=MonkeyRunner.waitForConnection() MonkeyRunner.sleep(2)

（4）启动微信APP程序

```
device.shell("monkey -p com.tencent.mm -v 1") MonkeyRunner.sleep(6)
```

（5）触摸达内专家微信群进入聊天界面

```
device.touch(100,350,'DOWN_AND_UP') MonkeyRunner.sleep(2)
```

（6）触摸+显示红包选项页 device.touch(125,300,'DOWN_AND_UP') MonkeyRunner.sleep(2)

（7）触摸红包选项 device.touch(300,450,'DOWN_AND_UP') MonkeyRunner.sleep(2)

（8）触摸红包个数文本框

```
device.touch(560,1000,'DOWN_AND_UP')
```

（9）对红包个数文本框中输入10个

```
device.type('10') MonkeyRunner.sleep(2)
```

（10）触摸红包总金额文本框

```
device.touch(450,800,'DOWN_AND_UP')
```

（11）对红包总金额文本框输入10元 device.type('10') MonkeyRunner.sleep(2)

（12）收起数字键盘，触摸V(向下箭头)

```
device.touch(670,900,'DOWN_AND_UP')
```

（13）触摸“塞钱进红包”按钮 device.touch(780,460,'DOWN_AND_UP') MonkeyRunner.sleep(6)

（14）支付确认，分别输入6位密码，通过数字键盘点击形式输入的 a、输入第一位密码 device.touch(200,500,'DOWN_AND_UP') b、输入第二位密码 device.touch(300,450,'DOWN_AND_UP')

c、输入第三位密码 device.touch(850,760,'DOWN_AND_UP')

后续密码输入同理

```
MonkeyRunner.sleep(3)
```

（15）触摸back,取消红包选项页的展示

```
device.press('KEYCODE_BACK','DOWN_AND_UP')
```

（16）对当前测试结果进行截图操作

```
image=device.takeSnapshot()
```

（17）对截图的图片保存到本地

```
device.writeToFile('D:\test_image.png','png')
```

（18）控制台打印输入“end”

```
print 'End'
```

ADB常用命令

简要APP测试流程