# 《停指测试技术》实验报告书

学生姓名：杨溢

学 号：31601372

班 级：软件工程1601

指导老师：杨枨

专 业：软件工程

提交时间：2018年6月2日

**白盒测试实验**  
  
  
**实验原理:**  
  
白盒测试原理：分析程序的内部逻辑结构，选择适当的覆盖标准，设计测试用例，对主要路径进行尽可能多的测试。白盒测试测试用例一般采用逻辑覆盖法进行设计。  
  
语句覆盖：选择足够的测试用例，使得程序中每个语句至少都能被执行一次。  
  
判定覆盖：执行足够的测试用例，使得程序中每个判定至少都获得一次“真”值和“假”值。  
  
条件覆盖：执行足够的测试用例，使得所有判定中的每个条件至少都获得一次“真”值和“假”值。  
  
判定/条件覆盖：执行足够的测试用例，使得判定中每个条件取到各种可能的值，并使每个判定取到各种可能的结果。  
  
条件组合覆盖：执行足够的例子，使得每个判定中条件的各种可能组合都至少出现一次。  
  
路径覆盖：路径覆盖是相当强的逻辑覆盖，它保证程序中每条可能的路径都至少执行一次。  
  
  
**实验步骤：**  
  
1、测试程序源代码

package com.example.a79014.activitytest;

import android.content.pm.ApplicationInfo;

import android.content.pm.PackageInfo;

import android.content.pm.PackageManager;

import android.content.pm.ResolveInfo;

import android.graphics.drawable.Drawable;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.content.Intent;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.BaseAdapter;

import android.widget.ImageView;

import android.widget.ListView;

import android.widget.TextView;

import java.util.ArrayList;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

import java.util.logging.Handler;

public class ThirdActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.third\_layout);

}

private void queryFilterAppInfo() {

PackageManager pm = this.getPackageManager();

// 查询所有已经安装的应用程序

List<ApplicationInfo> appInfos= pm.getInstalledApplications(PackageManager.GET\_UNINSTALLED\_PACKAGES);// GET\_UNINSTALLED\_PACKAGES代表已删除，但还有安装目录的

List<ApplicationInfo> applicationInfos=new ArrayList<>();

// 创建一个类别为CATEGORY\_LAUNCHER的该包名的Intent

Intent resolveIntent = new Intent(Intent.ACTION\_MAIN, null);

resolveIntent.addCategory(Intent.CATEGORY\_LAUNCHER);

// 通过getPackageManager()的queryIntentActivities方法遍历,得到所有能打开的app的packageName

List<ResolveInfo> resolveinfoList = getPackageManager()

.queryIntentActivities(resolveIntent, 0);

Set<String> allowPackages=new HashSet();

for (ResolveInfo resolveInfo:resolveinfoList){

allowPackages.add(resolveInfo.activityInfo.packageName);

}

for (ApplicationInfo app:appInfos) {

// if((app.flags & ApplicationInfo.FLAG\_SYSTEM) <= 0)//通过flag排除系统应用，会将电话、短信也排除掉

// {

//

吕煜杰:

package com.example.a79014.activitytest;

import android.content.Intent;

import android.net.Uri;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

public class FirstActivity extends AppCompatActivity {

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.main,menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()){

case R.id.add\_item:

Toast.makeText(this,"You clicked Add",Toast.LENGTH\_SHORT).show();

break;

case R.id.remove\_item:

Toast.makeText(this,"You clicked Remove",Toast.LENGTH\_SHORT).show();

break;

default:

}

return true;

}

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.first\_layout);

Button button1 = (Button) findViewById(R.id.button\_1);

button1.setOnClickListener(new View.OnClickListener(){

@Override

public void onClick(View v){

Intent intent = new Intent(FirstActivity.this,SecondActivity.class);

startActivity(intent);

}

}

);

}

}

package com.example.a79014.activitytest;

import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

public class SecondActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.second\_layout);

Button button1 = (Button) findViewById(R.id.button\_1);

button1.setOnClickListener(new View.OnClickListener(){

@Override

public void onClick(View v){

Intent intent = new Intent(SecondActivity.this,ThirdActivity.class);

startActivity(intent);

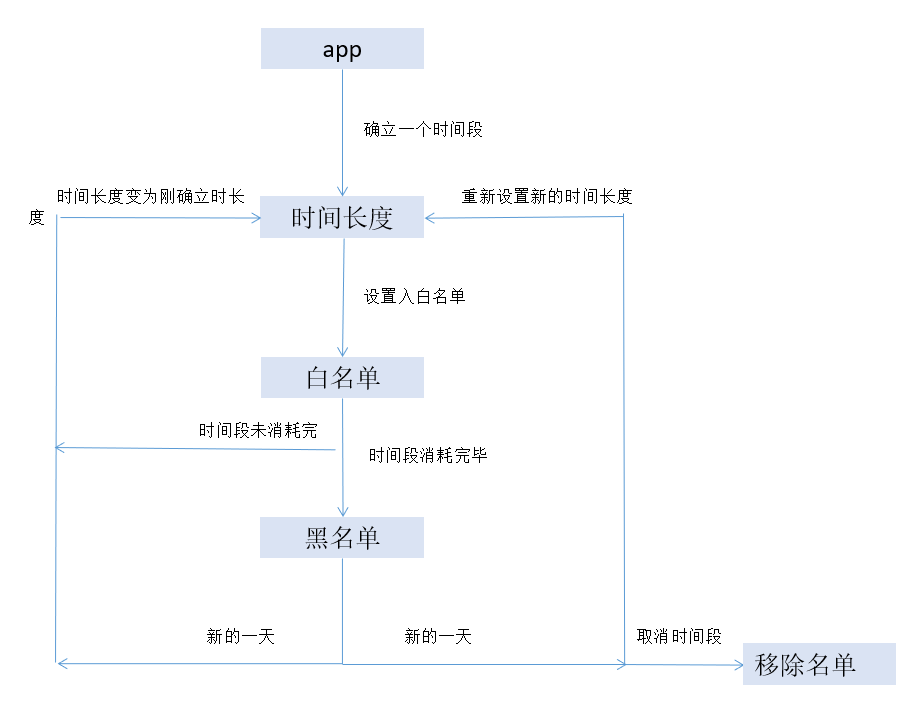
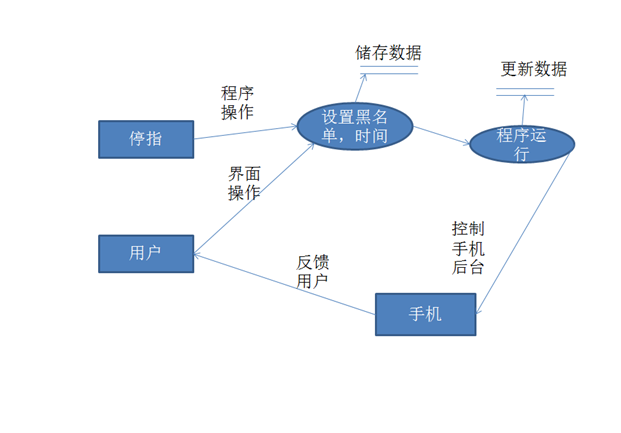
}

});

}

}

2、测试程序流程图

  
  
  
3、测试用例设计  
  
输入“已经注册的帐号”，测试判断语句输出的是true，测试数据库给出的反馈是true

输入“未注册的帐号”，测试判断语句输出的是false，测试数据库给出的反馈是false  
输入“已经注册的帐号,正确的密码”，测试判断密码语句输出的是true，数据库给出的反馈是true  
输入“已经注册的帐号，错误的密码”，测试判断密码语句输出的是false, 测试数据库给出的反馈是false  
输入“未注册的帐号，任意密码”，测试判断密码语句输出的是false, 测试数据库给出的反馈是false  
  
  
  
  
 **黑盒测试实验报告**  
  
  
  
  
**实验原理:**

* 1. 黑盒测试原理：不考虑程序的内部结构与特性，只根据程序功能或程序的外部特性设计测试用例。  
       
     等价分类法：根据程序的I/O特性，将程序的定义域划分为有限个等价区段 —“等价类”，从等价类中选择出的用例，具有“代表性”。 应按照输入条件（如输入值的范围，值的个数，值的\*\*，输入条件必须如何）划分为有效等价类和无效等价类。有效等价类，对于程序的规格说明是合理的、有意义的输入数据构成的\*\*。无效等价类，对于程序的规格说明，是不合理的，是没有意义的输入数据构成的\*\*。  
       
     边值分析法：选择等价类的边缘值作为测试用例，让每个等价类的边界都得到测试，选择测试用例既考虑输入亦考虑输出。  
       
     决策表：在一些数据处理问题当中，某些操作的实施依赖于多个逻辑条件的组合，即：针对不同逻辑条件的组合值，分别执行不同的操作。最适合描述在多逻辑条件取值的组合所构成的复杂情况下，分别执行哪些不同的动作。  
       
     因果图法：一些程序的功能可以用判定表（或称决策表）的形式来表示，并根据输入条件的组合情况规定相应的操作。它是一种利用图解法分析输入的各种组合情况，从而设计测试用例的方法，它适合于检查程序输入条件的各种组合情况。  
       
       
       
       
     **实验步骤：**  
       
     1、测试用例设计  
       
     等价划分和测试用例设计

等价划分：

* + - 1. 有效等价类：（1）已经注册的帐号
      2. 无效等价类：（1）未注册的帐号 （2）错误的密码

|  |  |  |  |
| --- | --- | --- | --- |
| 用例 | 帐号和密码 | 预期输出 | 有效/无效等价类 |
| 1 | 已经注册过的帐号和正确密码 | 登录成功，到达FirstActivity | 有效 |
| 2 | 未注册过的帐号和任意密码 | 登录失败，帐号不存在 | 无效 |
| 3 | 已经注册过的帐号和错误密码 | 登录失败，密码错误 | 无效 |
| 4 | 空帐号和空密码 | 登录失败，  请输入帐号和密码 | 无效  无效 |
| 5 | 已经注册过的帐号和空密码 | 登录失败，请输入密码 | 无效 |
| 6 | 空帐号和密码 | 登录失败，输入帐号 | 无效 |