

单元测试用例



课程名称 软件工程

题目名称 单元测试用例

任课教师 杨 枨

组 号： G18

组 长： 谢子文

组 员： 梁泽生

组 员： 黄 馨

2020 年12月 29 日

**目录**

[一、 引言： 3](#_Toc31401)

[1.1编写目的 3](#_Toc12028)

[1.2背景 3](#_Toc4665)

[1.3定义 4](#_Toc7637)

[1.4参考资料 4](#_Toc31036)

[1.5版本历史 5](#_Toc20934)

[二、 单元测试用例（黑盒测试）： 5](#_Toc6040)

[2.1专注计时模块 5](#_Toc26819)

[2.2统计总览模块 6](#_Toc6798)

[2.3排行榜模块 7](#_Toc11707)

[2.4历史时间轴模块 7](#_Toc28987)

[2.5商店模块 8](#_Toc2968)

[2.6设置模块 8](#_Toc14873)

[三、 单元测试用例（白盒测试）： 10](#_Toc28989)

[3.1单元测试1（登录界面） 10](#_Toc4490)

[3.2单元测试2（主界面计时） 11](#_Toc30310)

[3.2.1倒计时 11](#_Toc29107)

[3.2.2定向计时 14](#_Toc20634)

[3.3单元测试3（蒸笼） 18](#_Toc5679)

[3.4单元测试4（排行榜） 20](#_Toc24711)

[3.5单元测试5（历史时间轴） 23](#_Toc10852)

[3.6单元测试6（用户反馈） 26](#_Toc9102)

[3.7单元测试7（用户个人信息） 28](#_Toc32643)

[3.8单元测试8（开发者界面） 31](#_Toc29245)

[3.9单元测试9（购买蒸品） 34](#_Toc12149)

# 引言：

## 1.1编写目的

单元测试是代码正确性验证的最重要的工具，也是系统测试当中最重要的环节。也是唯一需要编写代码才能进行测试的一种测试方法。在标准的开发过程中，单元测试的代码与实际程序的代码具有同等的重要性。每一个单元测试，都是用来定向测试其所对应的一个单元的数据是否正确。

单元测试是由程序员自己来完成，最终受益的也是程序员自己。可以这么说，程序员有责任编写功能代码，同时也就有责任为自己的代码编写单元测试。执行单元测试，就是为了证明这段代码的行为和我们期望的一致。

## 1.2背景

说明：

1. 待开发的软件系统的名称：

《专注包子》（又名《FocusBuns》）

1. 本项目的任务提出者、开发者、用户及实现该软件的计算中心或计算机网络；

提出者：

任务发布者：杨枨教授

G18小组组长：谢子文

开发者：

G18小组所有成员：谢子文、黄馨、梁泽生

用户：（分为目标用户和其他用户）

面向自制力不强，需要外力帮助监督不玩手机、或者想要专注学习和工作的人们。

目标用户：建筑1801王宇键、法学1802王新、软工1802陈骁

其他用户：为随机用户，因这部分用户的采访是不固定的，并且这部分用户不想透露个人信息，所以为其他用户。

实现该微信小程序的计算中心：

小组三人个人的计算机：

Huawei matebook14、联想威6 2020、戴尔轻薄笔记本

操作系统：Windows® 10 32 / 64bit / 7 32 / 64bit / XP

处理器：

Intel® Core™ 2 Duo E6600 or AMD Phenom™ X3 8750 processor or better

内存：2 GB RAM or more

显卡：Video card must be 256MB or more

实现该微信小程序的计算机网络：任何网络。

1. 该软件系统同其他系统或其他机构的基本的相互来往关系。

（1）微信小程序官方管理系统

（2）所使用开发软件的所有者机构

（3）github版本管理远程库

## 1.3定义

[1] 白盒测试：盒测试又称结构测试、透明盒测试、逻辑驱动测试或基于代码的测试。白盒测试是一种测试用例设计方法，盒子指的是被测试的软件，白盒指的是盒子是可视的，即清楚盒子内部的东西以及里面是如何运作的。"白盒"法全面了解程序内部逻辑结构、对所有逻辑路径进行测试。"白盒"法是穷举路径测试。在使用这一方案时，测试者必须检查程序的内部结构，从检查程序的逻辑着手，得出测试数据。贯穿程序的独立路径数是天文数字。

## 1.4参考资料

[1] 《GB856T——88》国标文档

[2] 《软件工程导论》清华大学出版社，张海藩、牟永敏编著

[3] 单元测试的目的及使用

<https://blog.csdn.net/weixin_43885834/article/details/108627476>

## 1.5版本历史

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 版本修订日期 | 版本修改人 | 是否通过审核 |
| 单元测试用例v0.1.0 | 2020/12/29 | 黄馨 | 是 |
| 单元测试用例v0.1.1 | 2020/12/29 | 黄馨 | 是 |
| 单元测试用例v0.1.2 | 2020/12/31 | 黄馨 | 是 |

# 单元测试用例（黑盒测试）：

## 2.1专注计时模块

|  |  |  |  |
| --- | --- | --- | --- |
| 测试模块：专注计时 | | | |
| 标识符：2.0 | | | |
| 测试项：专注计时分为两个单元，一个倒计时，一个定向计时，测试者两种模式下，设定的时间是否能正常显示并且能正常运行计时功能，专注记录是否能保存至数据库。 | | | |
| 测试环境：微信web开发者工具 | | | |
| 前置条件：数据库和其他功能运行环境正常 | | | |
| 操作步骤：  倒计时模块：  1.设定专注时间，例如10分钟，然后开始专注计时。  2.关闭手机或者退出微信app。  3.打开手机回到微信发现计时仍在进行。  4.继续计时下去知道时间结束或者提前结束这个番茄钟。  定向计时模块：  1.设定定点时间，例如现在的时间是23:04，我设为23:07结束，然后开始专注计时。  2.关闭手机或者退出微信app。  3.打开手机回到微信发现计时仍在进行。  4.继续计时下去知道时间结束或者提前结束这个番茄钟。 | | | |
| 子用例编码 | 输入数据 | 期望数据 | 实际结果 |
| 2.1 | 设定专注时间为10分钟 | 显示10分钟 | 显示10分钟 |
| 2.2 | 设定定时时间为23:07 | 显示23:07 | 显示23:07 |
| 2.1/2.2 | 点击开始按钮 | 开始专注计时 | 专注计时进行 |

## 2.2统计总览模块

|  |  |  |  |
| --- | --- | --- | --- |
| 测试模块：统计总览（蒸笼） | | | |
| 标识符：3.0 | | | |
| 测试项：是否能成功进入到统计总览页面，是否能成功显示出已经成功的包子和失败的包子数量，专注时间总长，按住小球可以进行拖动。 | | | |
| 测试环境：微信web开发者工具 | | | |
| 前置条件：数据库和其他功能运行环境正常 | | | |
| 操作步骤：  1.点击菜单栏蒸笼按钮。  2.进入页面后，切换月份，显示不同时间的成功失败包子数量和专注时长。  3.按住包子可以进行拖动。 | | | |
| 子用例编号 | 输入数据 | 期望输出 | 实际结果 |
| 3.1 | 点击进入蒸笼界面，切换时间显示 | 该月的成功包子数量 | 显示了该月的成功包子数量 |
| 3.2 | 点击进入蒸笼界面，切换时间显示 | 该月的时间总长 | 该月的时间总长 |
| 3.3 | 点击进入蒸笼界面，切换时间显示 | 该月的失败包子数量 | 该月的失败包子数量 |
| 3.4 | 拖动包子 | 包子能被拖动 | 包子被拖动 |

## 2.3排行榜模块

|  |  |  |  |
| --- | --- | --- | --- |
| 测试模块：排行榜（蒸笼） | | | |
| 标识符：4.0 | | | |
| 测试项：是否能成功进入到排行榜页面，是否能显示出用户的专注时间排行榜，并能看到用户自己的名次，和本回合剩余时间（也就是距离这一次排名结束的时间）。 | | | |
| 测试环境：微信web开发者工具 | | | |
| 前置条件：数据库和其他功能运行环境正常 | | | |
| 操作步骤：  1.点击菜单栏排行榜按钮。  2.进入页面后，显示各个用户的排行榜，还有用户自己的排行榜名次等信息。 | | | |
| 子用例编号 | 输入数据 | 期望输出 | 实际结果 |
| 4.1 | 点击进入排行榜界面 | 距离本回合结束还有多久时间 | 距离本回合结束还有多久时间 |
| 4.2 | 点击进入排行榜界面 | 所有用户的排名 | 所有用户的排名 |
| 4.3 | 点击进入排行榜界面 | 我的排名信息 | 我的排名信息 |

## 2.4历史时间轴模块

|  |  |  |  |
| --- | --- | --- | --- |
| 测试模块：历史时间轴 | | | |
| 标识符：5.0 | | | |
| 测试项：是否能成功进入到历史时间轴页面，是否能显示出日历，能否在切换日历日期的时候，下方显示当天的专注记录。 | | | |
| 测试环境：微信web开发者工具 | | | |
| 前置条件：数据库和其他功能运行环境正常 | | | |
| 操作步骤：  1.点击菜单栏历史时间轴按钮。  2.进入页面后，显示出一个很大的日历表，当天可以查看当天的专注记录。  3.用户点击日历表上其他天，不同的日期下方会显示该日期当天的专注记录。 | | | |
| 子用例编号 | 输入数据 | 期望输出 | 实际结果 |
| 5.1 | 点击进入历史时间轴页面 | 日历和当天的专注记录 | 日历和当天的专注记录 |
| 5.2 | 点击日里表上其他日期 | 该日期当天的专注记录 | 该日期当天的专注记录 |

## 2.5商店模块

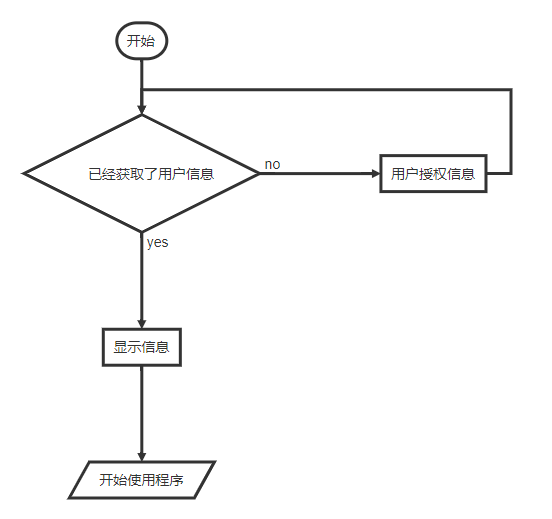
|  |  |  |  |
| --- | --- | --- | --- |
| 测试模块：商店 | | | |
| 标识符：6.0 | | | |
| 测试项：是否能成功进入到商店页面，用户是否能成功买到商品（蒸品）。 | | | |
| 测试环境：微信web开发者工具 | | | |
| 前置条件：数据库和其他功能运行环境正常 | | | |
| 操作步骤：  1.点击菜单栏商店按钮。  2.进入页面后，显示各个商品（蒸品）的信息和价格。  3.点击某个商品，可进行购买。 | | | |
| 子用例编号 | 输入数据 | 期望输出 | 实际结果 |
| 6.1 | 购买商品，点击确认 | 购买成功，能在专注计时前更换已拥有的蒸品。 | 购买成功，能在专注计时前更换已拥有的蒸品。 |

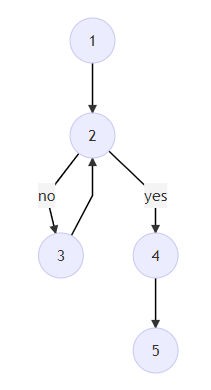
## 2.6设置模块

|  |  |  |  |
| --- | --- | --- | --- |
| 测试模块：设置 | | | |
| 标识符：7.0 | | | |
| 测试项：是否能成功进入到设置页面，是否能成功进入到个人中心，是否能点击相关按钮实现用户反馈，是否能实现开发者界面。 | | | |
| 测试环境：微信web开发者工具 | | | |
| 前置条件：数据库和其他功能运行环境正常 | | | |
| 操作步骤：  1.点击菜单栏设置按钮。  2.进入页面后，会显示个人信息、用户反馈等事情。  3.点击自己头像那一栏进入个人中心，出现个人中心页面。  4.点击用户反馈按钮，可进行用户反馈。  5.实现开发者界面。 | | | |
| 子用例编号 | 输入数据 | 期望输出 | 实际结果 |
| 7.1 | 点击进入个人中心界面 | 个人中心的信息：专注时长、完成数等 | 个人中心的信息：专注时长、完成数等 |
| 7.2 | 点击进入用户反馈 | 用户可以反馈的渠道 | 用户可以反馈的渠道 |
| 7.3 | 点击进入开发者界面 | 开发者管理员界面 | 开发者管理员界面 |

# 单元测试用例（白盒测试）：

## 3.1单元测试1（登录界面）





1：进入小程序

2：是否已经获取了用户信息

3：用户授权信息

4：显示信息

5：开始使用程序

首先根据 V(G) = E-N+2 = 5-5+2 = 2 计算出环形复杂度是2

然后再根据 V(G) = P+1 = 1+1 = 2 计算出环形复杂度是2，那么流图是正确的

语句覆盖：

路径1：1-2-3-4-5

路径2：1-2-4-5

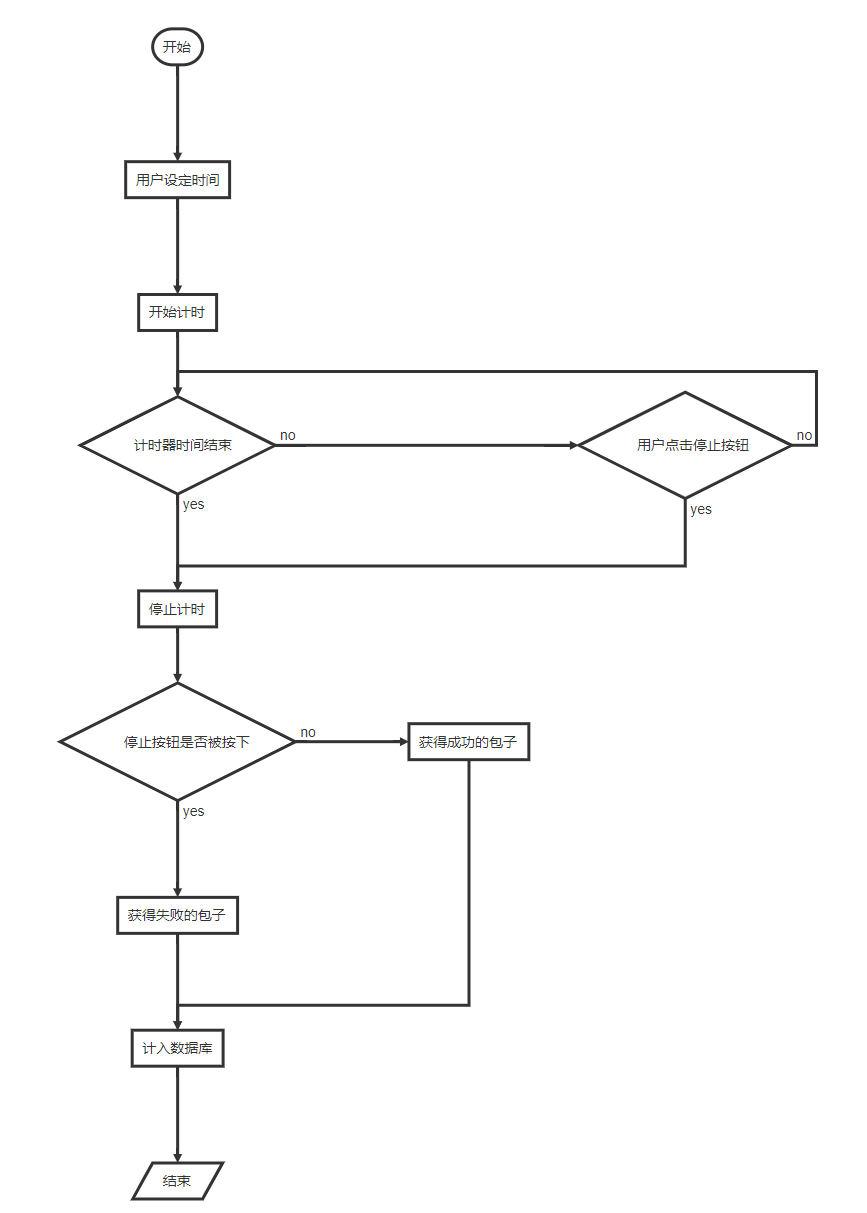
输入数据：用户进入小程序，获取用户信息，开始使用

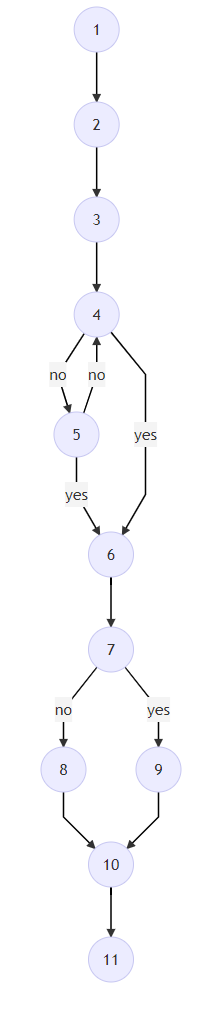
预期结果：用户能够进入小程序，系统能够获取用户信息，用户可以开始使用

小程序实现：

## 3.2单元测试2（主界面计时）

### 3.2.1倒计时





1：开始

2：用户设定时间

3：开始计时

4：计时时间结束

5：用户点击停止按钮

6：停止计时

7：计时按钮是否被按下

8：获得成功的包子

9：获得失败的包子

10：计入数据库

11：结束

首先根据 V(G) = E-N+2 = 13-11+2 = 4 计算出环形复杂度是4

然后再根据 V(G) = P+1 = 3+1 = 4 计算出环形复杂度是4，那么流图是正确的

语句覆盖：

路径1：1-2-3-4-5-6-7-9-10-11

路径2：1-2-3-4-5-4-6-7-9-10-11

路径4：1-2-3-4-6-7-8-10-11

路径5：1-2-3-4-6-7-9-10-11

路径1：1-2-3-4-5-6-7-9-10-11

输入数据：用户开始计时后点击停止按钮

预期结果：计时正常停止，用户获得失败的包子

路径2：1-2-3-4-5-4-6-7-8-10-11

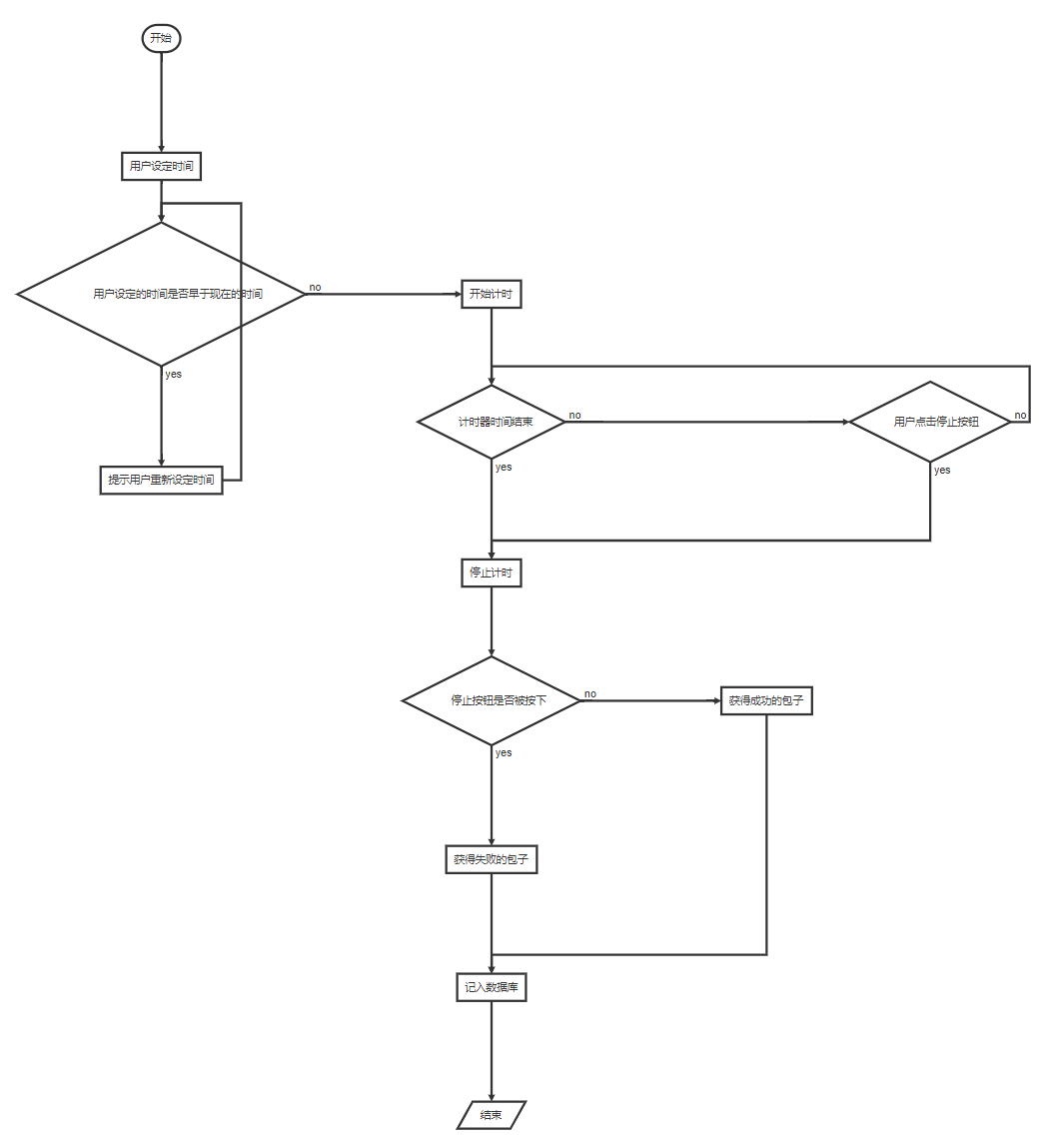
输入数据：用户开始计时后，计时器正常计时结束

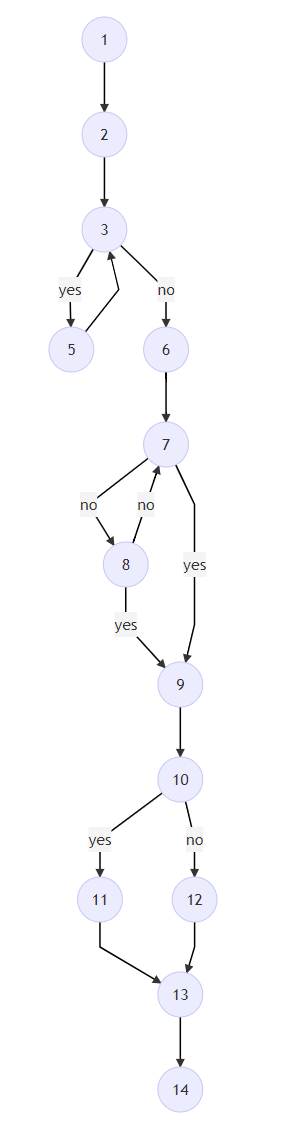
预期结果：计时正常停止，用户获得成功的包子

小程序实现：



### 3.2.2定向计时





1：开始

2：用户设定时间

3：设定的时间是否早于现在的时间

5：用户重新设定时间

6：开始计时

7：计时器时间结束

8：用户点击停止按钮

9：停止计时

10：停止按钮是否被按下

11：获得失败的包子

12：获得成功的包子

13：计入数据库

14：结束

首先根据 V(G) = E-N+2 = 16-13+2 = 5 计算出环形复杂度是5

然后再根据 V(G) = P+1 = 4+1 = 5 计算出环形复杂度是5，那么流图是正确的

语句覆盖：

路径1：1-2-3-4-5-6-7-9-10-11

路径2：1-2-3-4-5-4-6-7-9-10-11

路径4：1-2-3-4-6-7-8-10-11

路径5：1-2-3-4-6-7-9-10-11

路径1：1-2-3-4-5-6-7-9-10-11

输入数据：用户前设置的时间早于目前的时间

预期结果：提示用户重新设定时间

路径2：1-2-3-4-5-6-7-9-10-11

输入数据：用户开始计时后点击停止按钮

预期结果：计时正常停止，用户获得失败的包子

路径3：1-2-3-4-5-4-6-7-8-10-11

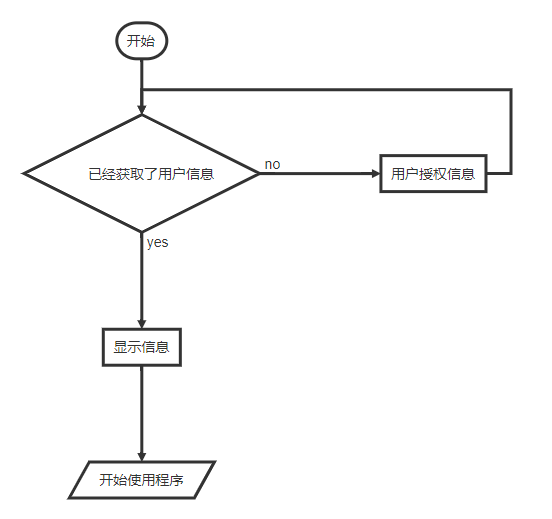
输入数据：用户开始计时后，计时器正常计时结束

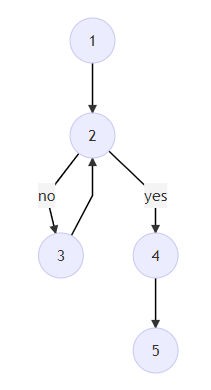
预期结果：计时正常停止，用户获得成功的包子

小程序实现：



## 3.3单元测试3（蒸笼）





1：进入小程序

2：是否已经获取了用户信息

3：用户授权信息

4：显示信息

5：开始使用程序

首先根据 V(G) = E-N+2 = 5-5+2 = 2 计算出环形复杂度是2

然后再根据 V(G) = P+1 = 1+1 = 2 计算出环形复杂度是2，那么流图是正确的

语句覆盖：

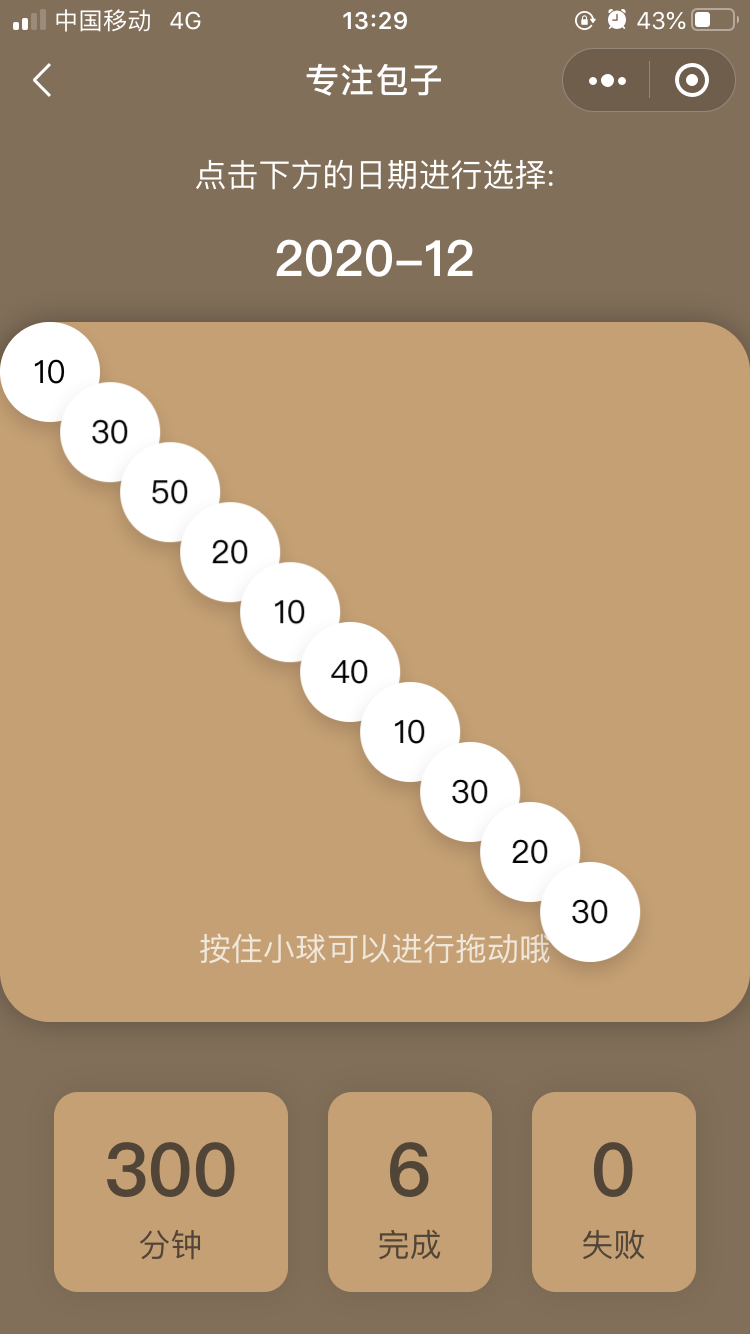
路径1：1-2-3-4-5

路径2：1-2-4-5

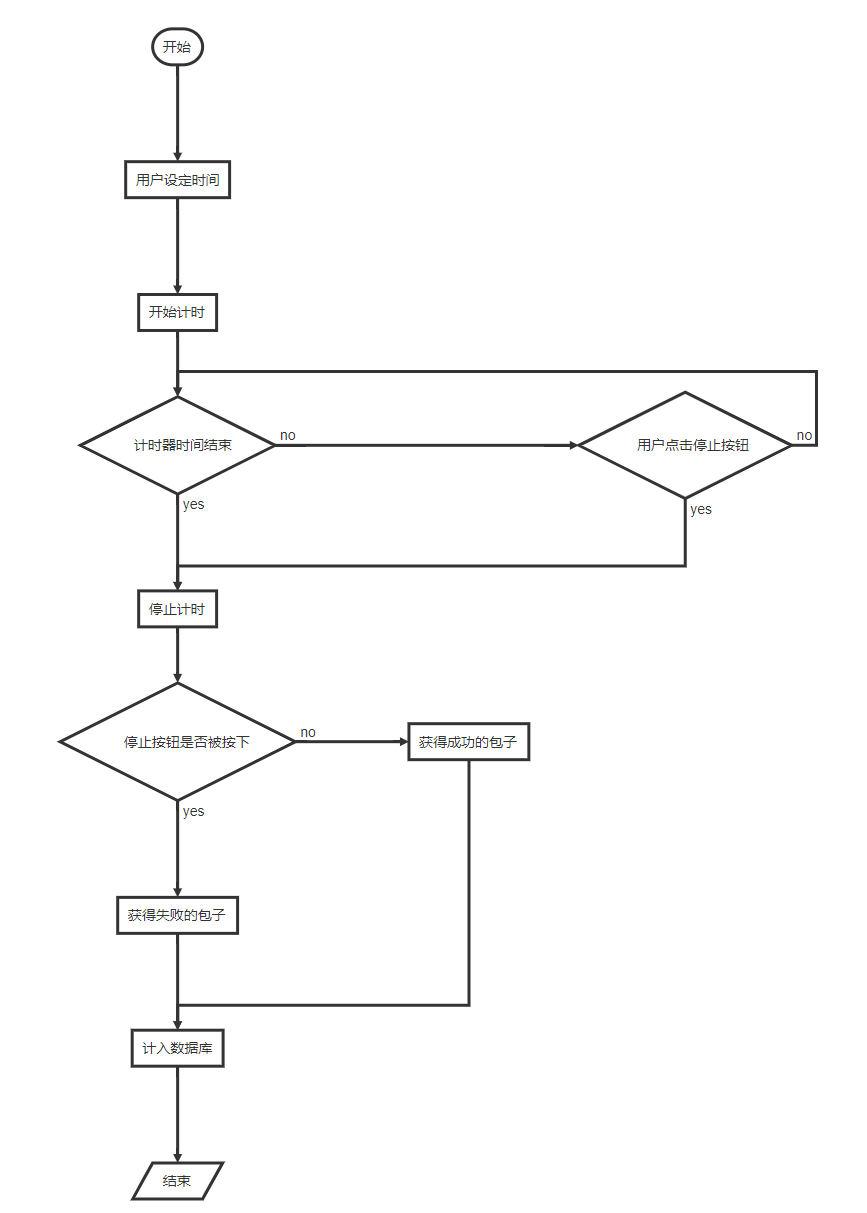
输入数据：用户进入小程序，获取用户信息，开始使用

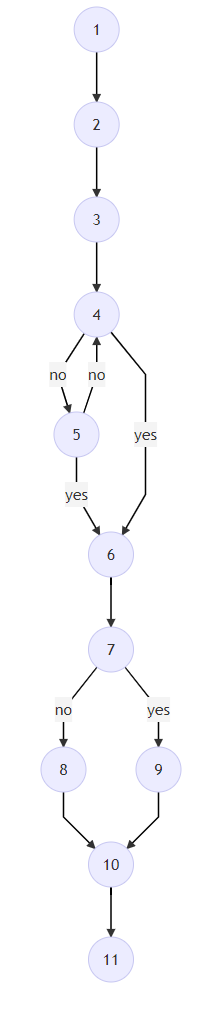
预期结果：用户能够进入小程序，系统能够获取用户信息，用户可以开始使用

小程序实现：



## 3.4单元测试4（排行榜）





1：开始

2：用户设定时间

3：开始计时

4：计时时间结束

5：用户点击停止按钮

6：停止计时

7：计时按钮是否被按下

8：获得成功的包子

9：获得失败的包子

10：计入数据库

11：结束

首先根据 V(G) = E-N+2 = 13-11+2 = 4 计算出环形复杂度是4

然后再根据 V(G) = P+1 = 3+1 = 4 计算出环形复杂度是4，那么流图是正确的

语句覆盖：

路径1：1-2-3-4-5-6-7-9-10-11

路径2：1-2-3-4-5-4-6-7-9-10-11

路径4：1-2-3-4-6-7-8-10-11

路径5：1-2-3-4-6-7-9-10-11

路径1：1-2-3-4-5-6-7-9-10-11

输入数据：用户开始计时后点击停止按钮

预期结果：计时正常停止，用户获得失败的包子

路径2：1-2-3-4-5-4-6-7-8-10-11

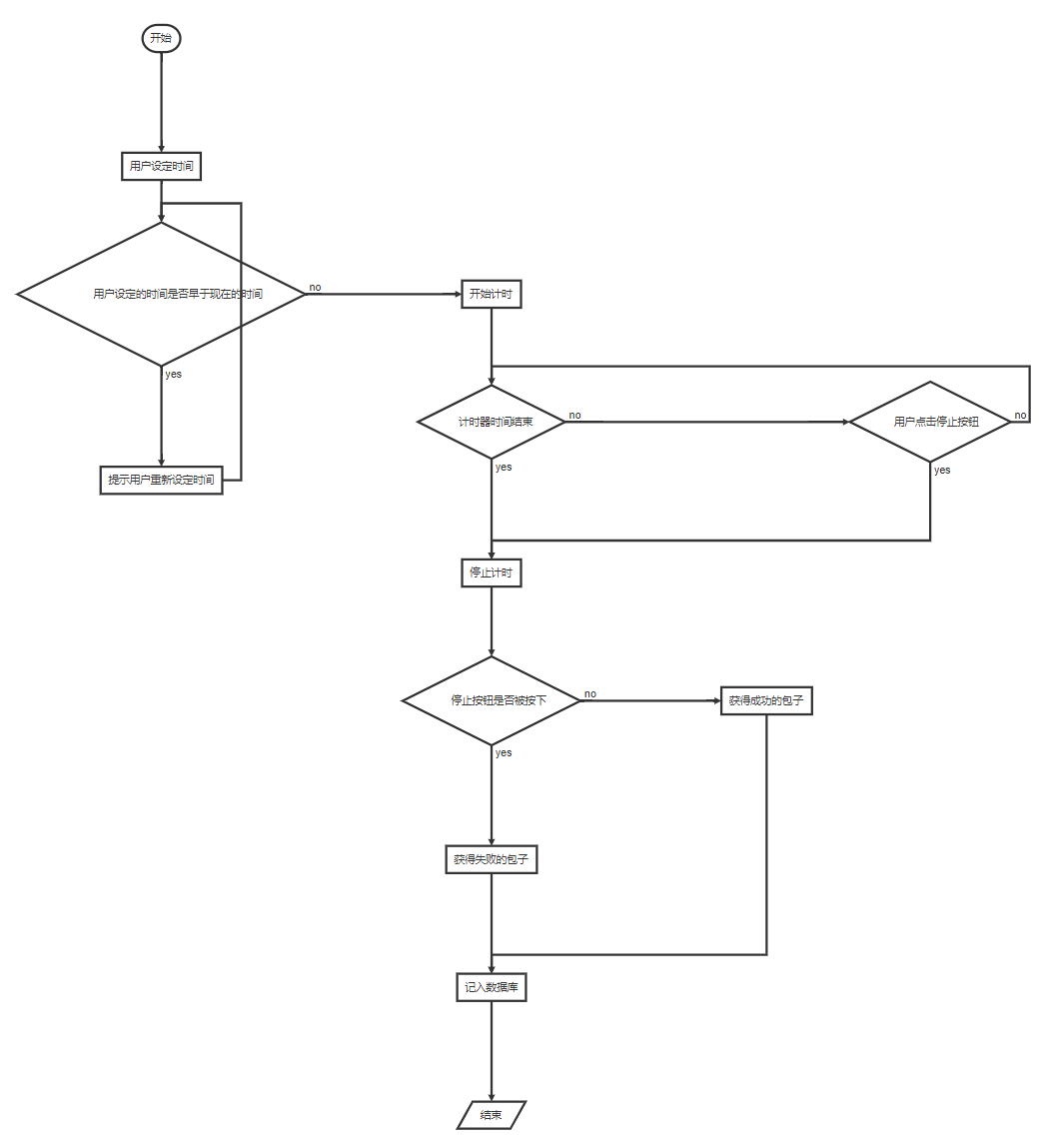
输入数据：用户开始计时后，计时器正常计时结束

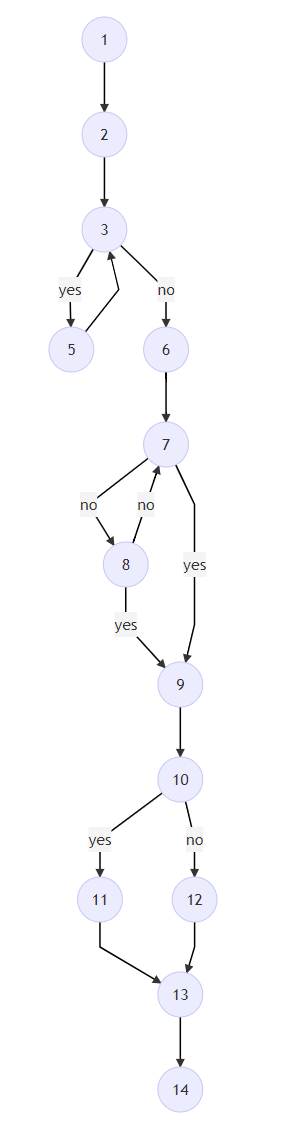
预期结果：计时正常停止，用户获得成功的包子

小程序显示：



## 3.5单元测试5（历史时间轴）





1：开始

2：用户设定时间

3：设定的时间是否早于现在的时间

5：用户重新设定时间

6：开始计时

7：计时器时间结束

8：用户点击停止按钮

9：停止计时

10：停止按钮是否被按下

11：获得失败的包子

12：获得成功的包子

13：计入数据库

14：结束

首先根据 V(G) = E-N+2 = 16-13+2 = 5 计算出环形复杂度是5

然后再根据 V(G) = P+1 = 4+1 = 5 计算出环形复杂度是5，那么流图是正确的

语句覆盖：

路径1：1-2-3-4-5-6-7-9-10-11

路径2：1-2-3-4-5-4-6-7-9-10-11

路径4：1-2-3-4-6-7-8-10-11

路径5：1-2-3-4-6-7-9-10-11

路径1：1-2-3-4-5-6-7-9-10-11

输入数据：用户前设置的时间早于目前的时间

预期结果：提示用户重新设定时间

路径2：1-2-3-4-5-6-7-9-10-11

输入数据：用户开始计时后点击停止按钮

预期结果：计时正常停止，用户获得失败的包子

路径3：1-2-3-4-5-4-6-7-8-10-11

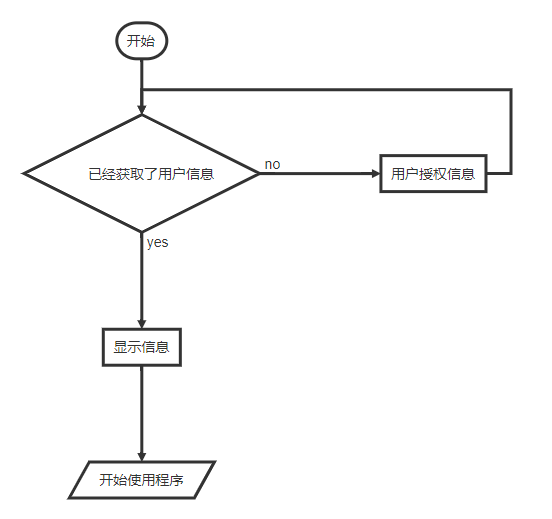
输入数据：用户开始计时后，计时器正常计时结束

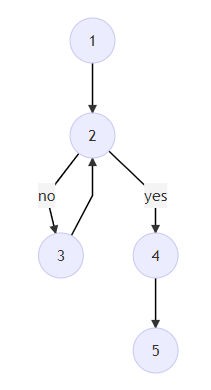
预期结果：计时正常停止，用户获得成功的包子

小程序显示：



## 3.6单元测试6（用户反馈）





1：进入小程序

2：是否已经获取了用户信息

3：用户授权信息

4：显示信息

5：开始使用程序

首先根据 V(G) = E-N+2 = 5-5+2 = 2 计算出环形复杂度是2

然后再根据 V(G) = P+1 = 1+1 = 2 计算出环形复杂度是2，那么流图是正确的

语句覆盖：

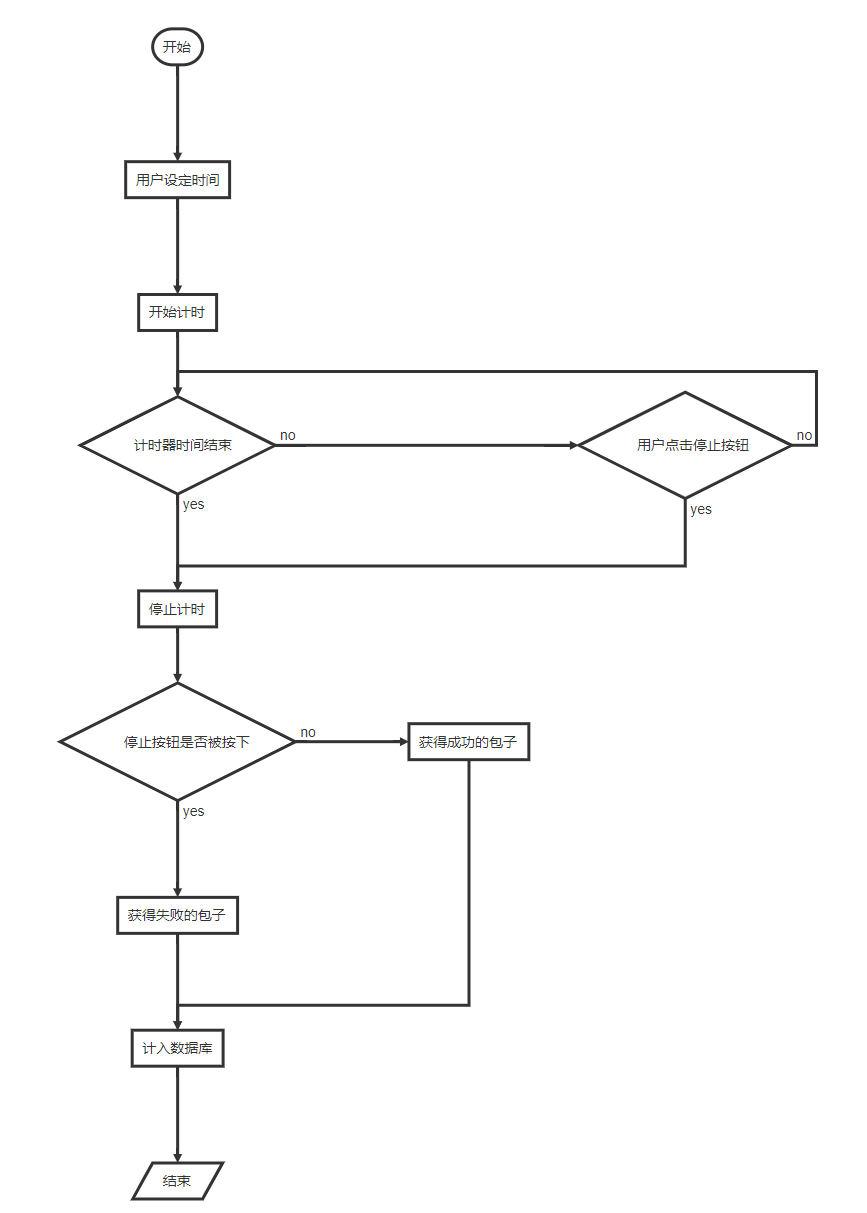
路径1：1-2-3-4-5

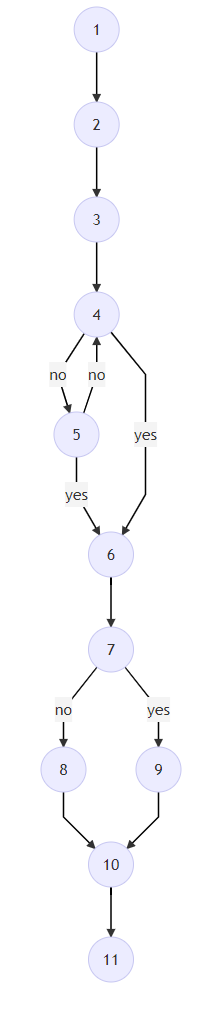
路径2：1-2-4-5

输入数据：用户进入小程序，获取用户信息，开始使用

预期结果：用户能够进入小程序，系统能够获取用户信息，用户可以开始使用

## 3.7单元测试7（用户个人信息）





1：开始

2：用户设定时间

3：开始计时

4：计时时间结束

5：用户点击停止按钮

6：停止计时

7：计时按钮是否被按下

8：获得成功的包子

9：获得失败的包子

10：计入数据库

11：结束

首先根据 V(G) = E-N+2 = 13-11+2 = 4 计算出环形复杂度是4

然后再根据 V(G) = P+1 = 3+1 = 4 计算出环形复杂度是4，那么流图是正确的

语句覆盖：

路径1：1-2-3-4-5-6-7-9-10-11

路径2：1-2-3-4-5-4-6-7-9-10-11

路径4：1-2-3-4-6-7-8-10-11

路径5：1-2-3-4-6-7-9-10-11

路径1：1-2-3-4-5-6-7-9-10-11

输入数据：用户开始计时后点击停止按钮

预期结果：计时正常停止，用户获得失败的包子

路径2：1-2-3-4-5-4-6-7-8-10-11

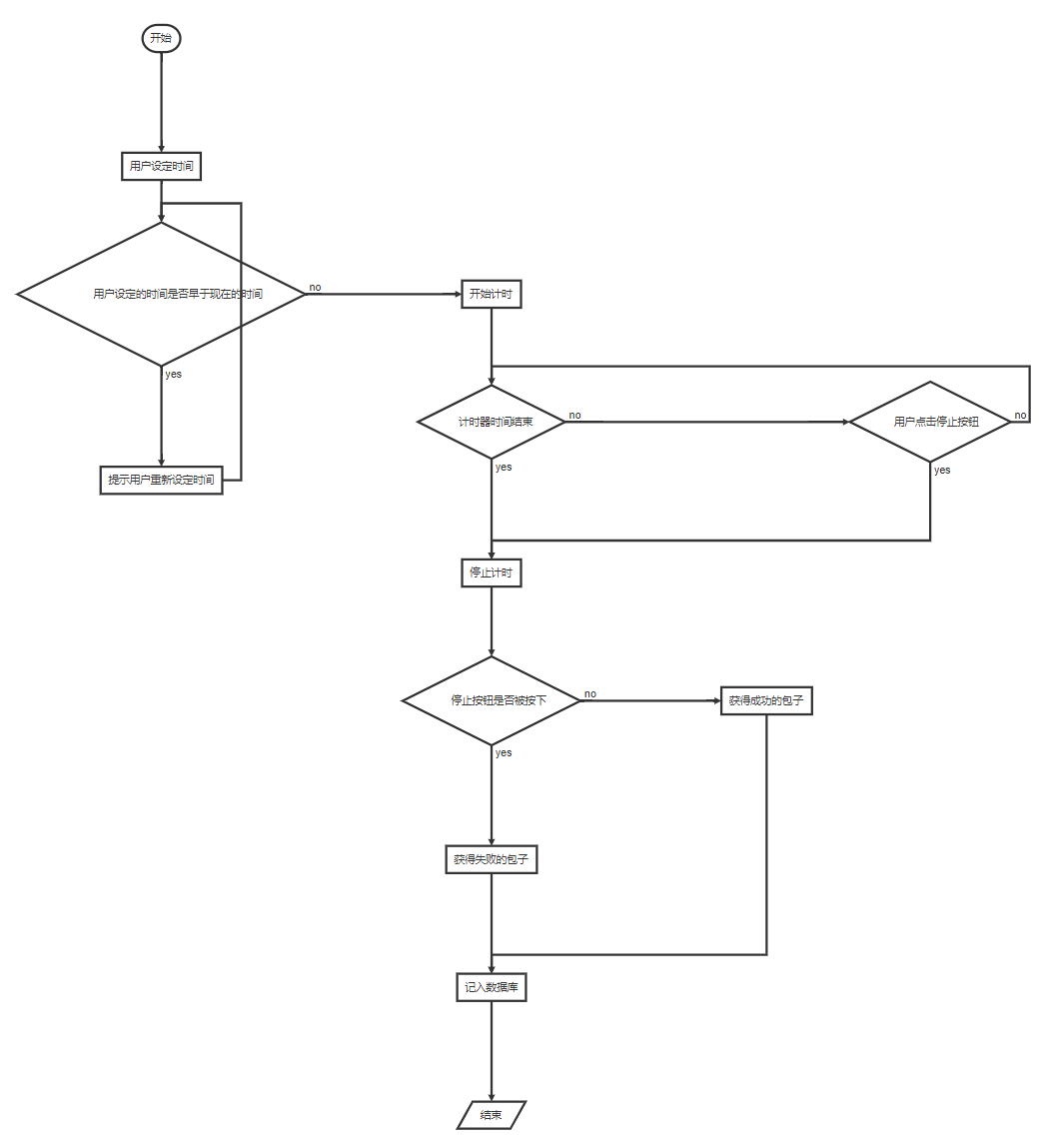
输入数据：用户开始计时后，计时器正常计时结束

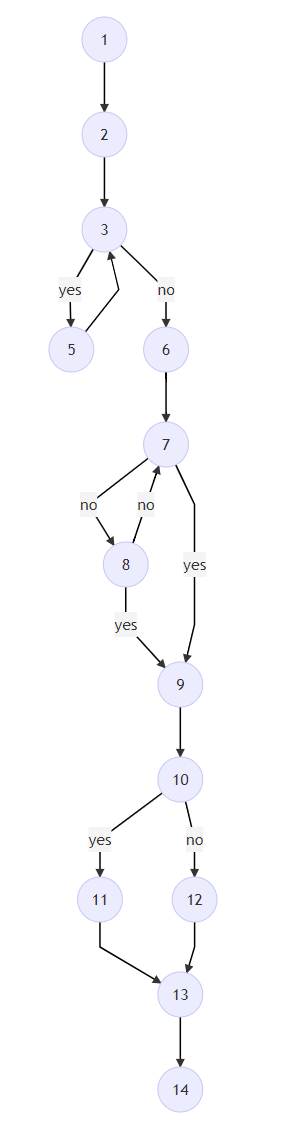
预期结果：计时正常停止，用户获得成功的包子

小程序显示：



## 3.8单元测试8（开发者界面）





1：开始

2：用户设定时间

3：设定的时间是否早于现在的时间

5：用户重新设定时间

6：开始计时

7：计时器时间结束

8：用户点击停止按钮

9：停止计时

10：停止按钮是否被按下

11：获得失败的包子

12：获得成功的包子

13：计入数据库

14：结束

首先根据 V(G) = E-N+2 = 16-13+2 = 5 计算出环形复杂度是5

然后再根据 V(G) = P+1 = 4+1 = 5 计算出环形复杂度是5，那么流图是正确的

语句覆盖：

路径1：1-2-3-4-5-6-7-9-10-11

路径2：1-2-3-4-5-4-6-7-9-10-11

路径4：1-2-3-4-6-7-8-10-11

路径5：1-2-3-4-6-7-9-10-11

路径1：1-2-3-4-5-6-7-9-10-11

输入数据：用户前设置的时间早于目前的时间

预期结果：提示用户重新设定时间

路径2：1-2-3-4-5-6-7-9-10-11

输入数据：用户开始计时后点击停止按钮

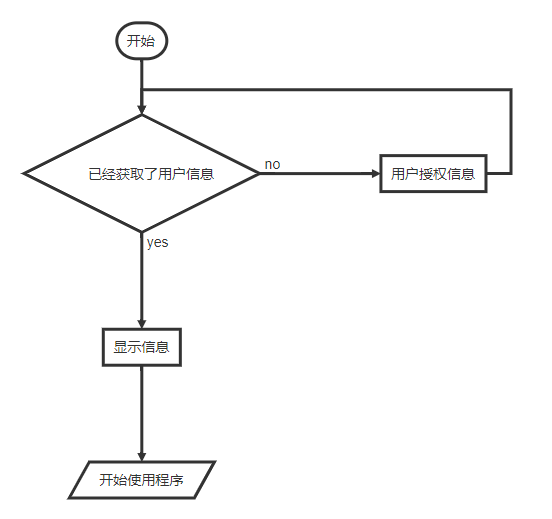
预期结果：计时正常停止，用户获得失败的包子

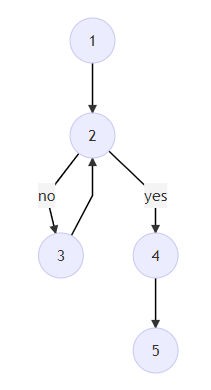
路径3：1-2-3-4-5-4-6-7-8-10-11

输入数据：用户开始计时后，计时器正常计时结束

预期结果：计时正常停止，用户获得成功的包子

## 3.9单元测试9（购买蒸品）





1：进入小程序

2：是否已经获取了用户信息

3：用户授权信息

4：显示信息

5：开始使用程序

首先根据 V(G) = E-N+2 = 5-5+2 = 2 计算出环形复杂度是2

然后再根据 V(G) = P+1 = 1+1 = 2 计算出环形复杂度是2，那么流图是正确的

语句覆盖：

路径1：1-2-3-4-5

路径2：1-2-4-5

输入数据：用户进入小程序，获取用户信息，开始使用

预期结果：用户能够进入小程序，系统能够获取用户信息，用户可以开始使用

小程序显示：

