电子科技大学信息与软件工程学院

**实 验 报 告**

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 贡献比例 |
| 2015220302008 | 徐智林 | 50% |
| 2016220304031 | 邓巧 | 50% |

学 号 2016220304031

姓 名 邓巧

（实验） 课程名称 软件工程

理论教师 周瑞

实验教师 郝晓青

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名：邓巧 学号：2016220304031 指导教师：郝晓青**

**实验地点：信软楼306 实验时间：2018.5.22**

1. **实验名称：质量保证**
2. **实验学时：4**
3. **实验目的：**

* 掌握软件编码规范
* 掌握单元测试的方法，包括白盒测试和黑盒测试
* 能对实际程序进行单元测试，能够撰写简单的测试报告

1. **实验原理：**
2. 黑盒测试

· 将软件当作内部可不见盒子。

· 行为测试，侧重于软件的功能需求。

· 发现功能、接口、数据访问、行为等错误。

常见黑盒测试：等价类划分，边界值测试。

2. 白盒测试

* 将软件当作内部可见、透明的盒子。
* 查看代码功能和实现逻辑路径。

常见白盒测试：语句覆盖，判定覆盖，条件覆盖，判定-条件覆盖。

1. **实验内容：**

1.输入三个整数a、b和c分别作为三角形的3条边，判断由这3条边构成的三角形类型是：等边三角形、等腰三角形、一般三角形、或非三角形（即不能构成一个三角形）。

* + 编写该程序，注意编程规范
  + 按照等价类划分方法设计测试用例，并进行测试
  + 按照边界值方法设计测试用例，并进行测试

2. 输入不超过50个数（以-1作为输入结束标志），计算其中有效的学生分数的个数，及有效分数的总分和平均值。

* + 编写程序，注意编程规范
  + 导出程序的控制流图，确定独立路径集合（基本路径集合）
  + 为每条路径设计测试用例，并进行测试

3. 细化实验2的系统设计，选择2个关键模块进行编码实现

4. 使用白盒或黑盒方法测试实现的模块

5. 撰写软件测试报告

1. **实验器材（设备、元器件）：**

PC机、编程软件

1. **实验步骤：**

步骤1：输入三个整数a（0<a<100）、b（0<b<100）和c（0<c<100）分别作为三角形的3条边，判断由这3条边构成的三角形类型是：等边三角形、等腰三角形、一般三角形、或非三角形（即不能构成一个三角形）

1. 编写该程序，注意编程规范
2. 按照等价类划分方法设计测试用例，并进行测试
3. 按照边界值方法设计测试用例，并进行测试

总结实验是否实现目标和达到预期效果。

步骤2: 输入不超过50个数（以-1作为输入结束标志），计算其中有效的学生分数的个数，及有效分数的总分和平均值。

1. 编写程序，注意编程规范
2. 导出程序的控制流图，确定独立路径集合（基本路径集合）
3. 为每条路径设计测试用例，并进行测试

步骤3: 细化实验2的系统设计，选择2个关键模块进行编码实现

步骤4: 使用白盒或黑盒方法测试实现的模块

步骤5: 撰写软件测试报告

1. **实验结果与分析（含重要数据结果分析或核心代码流程分析）**

**1.**

|  |
| --- |
| #include<stdio.h>  #define MAX(a,b) ((a) > (b) ? (a) : (b))  int max(int a, int b, int c){  return MAX(a,MAX(b,c));  }  int main(){  int a, b, c;  printf("input three seperated number: ");  scanf("%d%d%d", &a, &b, &c);  if(a+b+c-max(a,b,c) > max(a,b,c)){  if(a == b && b == c){  printf("Equilateral triangle\n");  }  else if(a == b || b == c){  printf("Isosceles triangle\n");  }  else{  printf("Triangle\n");  }  }  else{  printf("Not a triangle\n");  }  return 0;  } |

等价类划分

|  |  |  |
| --- | --- | --- |
| 输入条件 | 有效等价类 | 无效等价类 |
| 输入数据个数 | 3个（1） | 小于3个（2），  大于3个（3） |
| 数据类型 | 整数（4） | 其他类型数据（5） |
| 数据大小 | 1～2147483647（6） | 小于1（7），  大于2147483647（8） |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 输入 | 预期输出 | 覆盖的等价类 |
| 1 | 3 4 5 | Triangle | (1),(4),(6) |
| 2 | 2 3 | Illeagel input | (2),(4),(6) |
| 3 | 2 3 4 5 | Illeagel input | (3),(4),(6) |
| 4 | A b 3 | Illeagel input | (1),(5),(6) |
| 5 | -3 4 5 | Illeagel input | (1),(4),(7) |
| 6 | 2147483670 3 1000000 | Illeagal input | (1),(4),(8) |

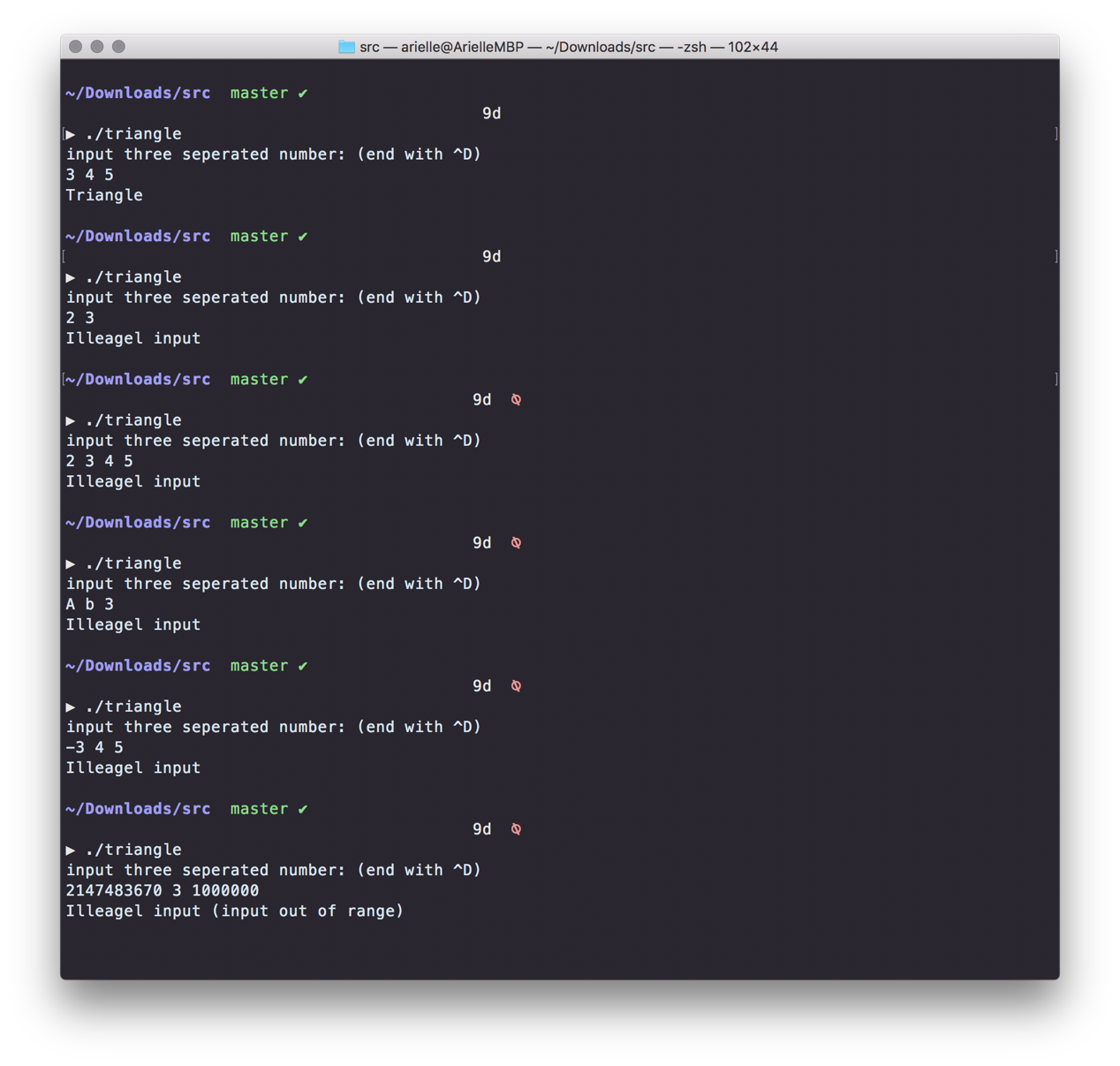
测试结果



经过修改后的代码

|  |
| --- |
| #include<stdio.h>  #define MAX(a,b) ((a) > (b) ? (a) : (b))  int max(int a, int b, int c){  return MAX(a,MAX(b,c));  }  int main(){  long long a = -1, b = -1, c = -1;  printf("input three seperated number: (end with ^D)\n");  if(scanf("%lld%lld%lld", &a, &b, &c) != 3 ||  scanf(" %\*c") != EOF || a <= 0 || b <= 0 || c <= 0){  printf("Illeagel input\n");  exit(EXIT\_FAILURE);  }  if(a > INT\_MAX || b > INT\_MAX || c > INT\_MAX){  printf("Illeagel input (input out of range)\n");  exit(EXIT\_FAILURE);  }  if(a+b+c-max(a,b,c) > max(a,b,c)){  if(a == b && b == c){  printf("Equilateral triangle\n");  }  else if(a == b || b == c){  printf("Isosceles triangle\n");  }  else{  printf("Triangle\n");  }  }  else{  printf("Not a triangle\n");  }  return 0;  } |

经过修改后的程序测试结果

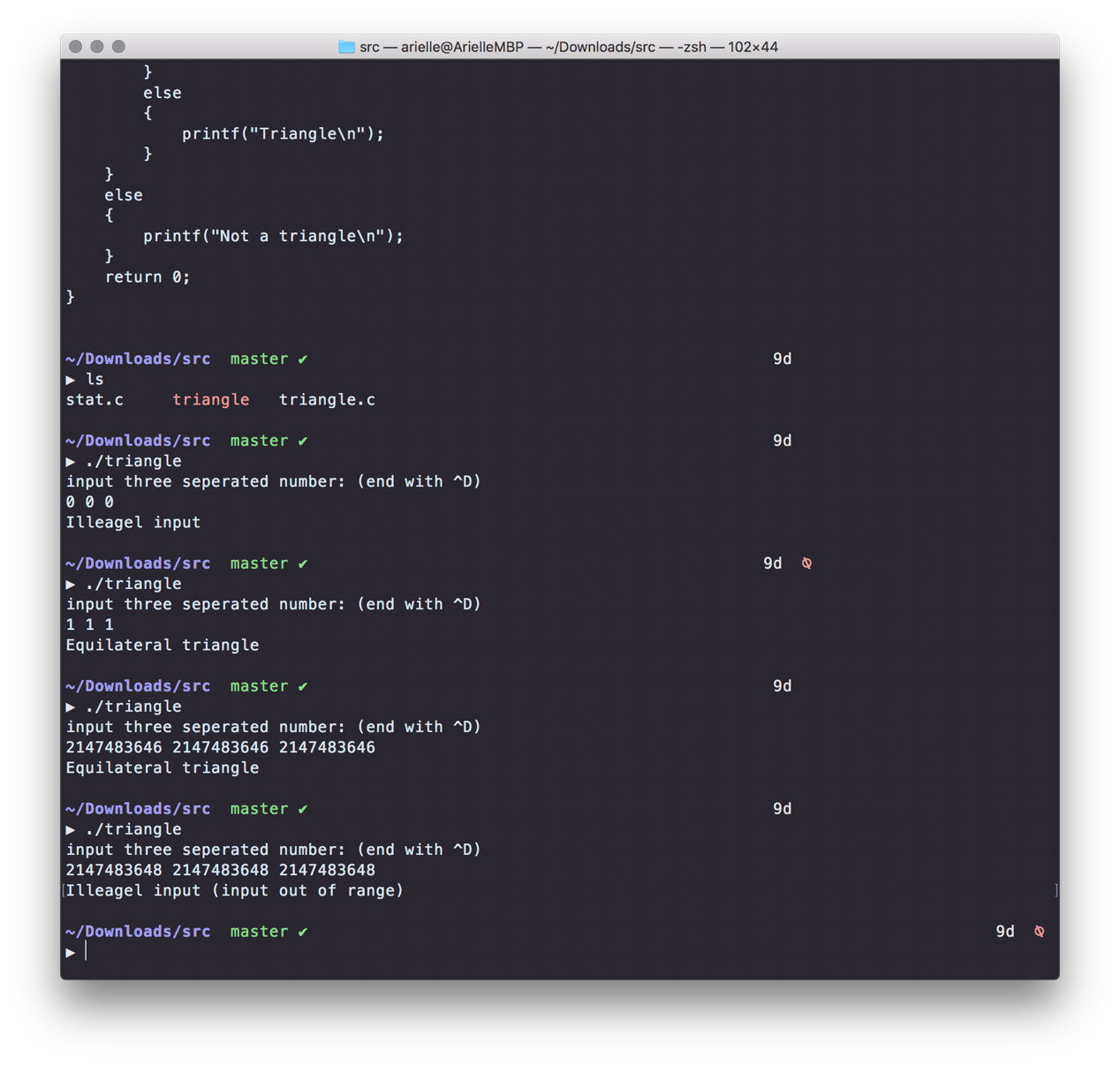


边界值测试:

根据输入条件，可设计测试用例：（0，0，0）（1，1，1）（2147483646，2147483646，2147483646）（2147483648，2147483648，2147483648）。

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 输入 | 预期输出 | 覆盖的等价类 |
| 7 | 0 0 0 | Illegal input | (1),(4),(7) |
| 8 | 1 1 1 | Equilateral triangle | (1),(4),(6) |
| 9 | 2147483646 2147483646 2147483646 | Equilateral triangle | (1),(4),(6) |
| 10 | 2147483648 2147483648 2147483648 | Illeagel input (input out of range) | (1),(4),(7) |

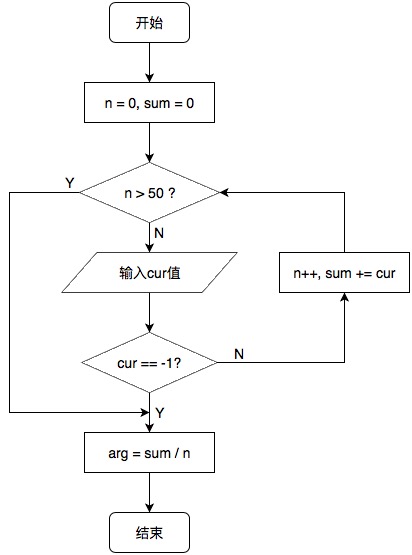
测试结果



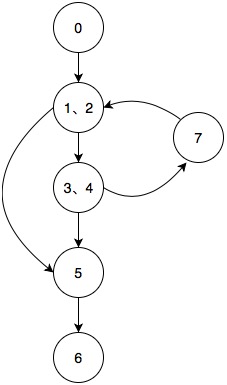
**2.**

|  |
| --- |
| #include<stdio.h>  #include<string.h>  #define MAX\_N 55  int main(){  int n = 0;  int sum = 0;  double avg;  int score[MAX\_N];  int \*cur = score;  memset(score, 0, sizeof(int) \* MAX\_N);  while(scanf("%d", cur)){  if(\*cur == -1){  \*cur = 0;  break;  }  else if(\*cur >=0 && \*cur <= 100){  ++n;  sum += \*cur;  ++cur;  }  }  avg = (double)sum / n;  printf("N=%d\nsum=%d\navg=%.2lf\n", n, sum, avg);  return 0;  } |

程序流图：

****

控制流图：

****

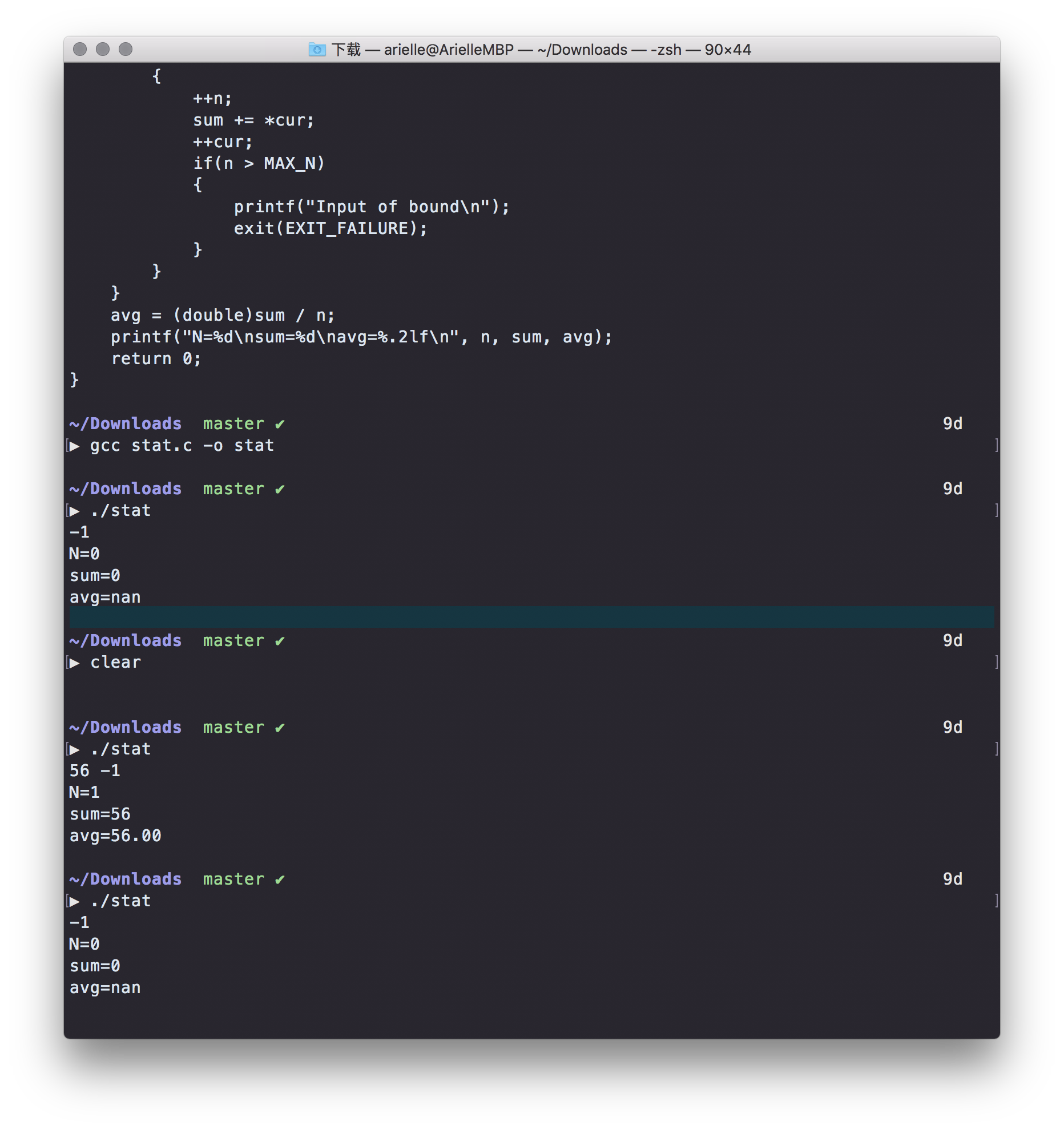
基本路径条数：2

0-1-2-3-4-7-1-2-3-4-5-6

0-1-2-3-4-5-6

设计测试用例

|  |  |  |
| --- | --- | --- |
| 测试编号 | 输入 | 期待输出 |
| 1 | 56 -1 | N = 0, sum = 56,  avg = 56.00 |
| 2 | -1 | N = 0, sum = 0,  Avg = nan |



1. **总结及心得体会：**
2. 测试应该多动脑，并且按照科学的流程来做。
3. 软件测试很重要，能够指导软件的开发
4. 测试驱动开发（TDD）是有效的指导开发健壮软件的开发方法
5. **对本实验过程及方法、手段的改进建议：**
   * 1. 实验任务较重，可以适当减轻一些。

**报告评分：**

**指导教师签字：**

**附录：软件测试报告**

**软件测试报告**

项目名称：学生事务管理系统

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 贡献比例 |
| 2016220304031 | 邓巧 | 50% |
| 2015220302008 | 徐智林 | 50% |

日期：2018年6月3日

# 引言

## 目的

本文档是关于学生个人事务管理系统（Student Essentials）的部分模块的需求的描述，为开发方和测试方提供交流的平台。版本1.0。

## 范围

该产品实现一个学生个人事务管理系统（Student Essentials），通过计算机和网络实现对学生个人事务和课程进度的统筹管理，方便每个学生对个人时间进行管理。

本次测试只对用户模型和用户登录登出模块进行了测试。

## 预期的读者和阅读建议

预期读者为设计人员、开发人员、项目管理人员、测试人员。

## 缩写和术语

控制器：接收浏览器或客户端的请求或提交信息，读取写入模型数据，将生成的对应的视图返回给浏览器或客户端。

视图：根据控制器传来的信息和选项生成浏览器或客户端能够显示的文件。

模型：根据控制器的请求与数据库进行交互，更新/修改数据库，返回请求的信息。

MVC：模型-视图-控制器体系结构。

会话：浏览器中通过Cookies实现的记录当前正在使用的用户的方法。

## 参考文献

[1] 软件工程-理论、方法与实践，孙家广等编著，高等教育出版社，2005年

# 测试时间、地点和人员

2018年6月1日到2018年6月3日

电子科技大学沙河校区

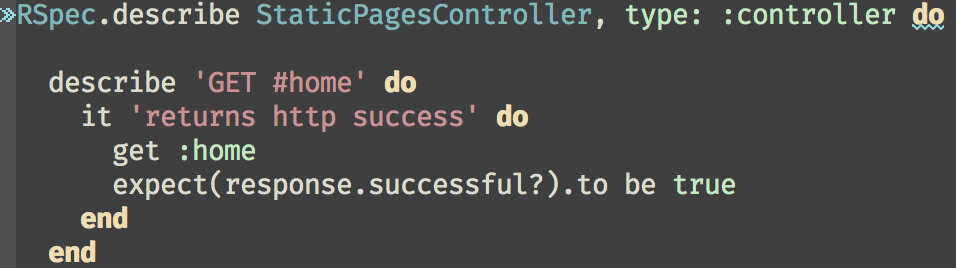
徐智林，邓巧进行黑盒集成测试。

# 测试环境及方法

使用的Rails开发并使用Rspec集成测试工具编写测试用例并进行黑盒集成测试。

# 测试用例执行情况

|  |  |
| --- | --- |
| 测试用例序号 | 1 |
| 测试目的及说明 | 主页能够被访问到 |
| 测试输入 | 访问主页 |
| 期望测试结果 | 可以访问主页 |
| 判断准则 | 回复报文代码为200（正常） |
| 实际测试结果 | 通过 |
| 测试人员 | 徐智林 |
| 测试日期 | 2018年6月3日 |
| 测试评价 | 控制器相关测试 |
| 问题标识 | 无 |
| 问题处理 | 无 |

图 1 主页能够被访问到

|  |  |
| --- | --- |
| 测试用例序号 | 2 |
| 测试目的及说明 | 关于页面能够被访问到 |
| 测试输入 | 访问关于（About） |
| 期望测试结果 | 可以访问关于 |
| 判断准则 | 回复报文代码为200（正常） |
| 实际测试结果 | 通过 |
| 测试人员 | 徐智林 |
| 测试日期 | 2018年6月3日 |
| 测试评价 | 控制器相关测试 |
| 问题标识 | 无 |
| 问题处理 | 无 |

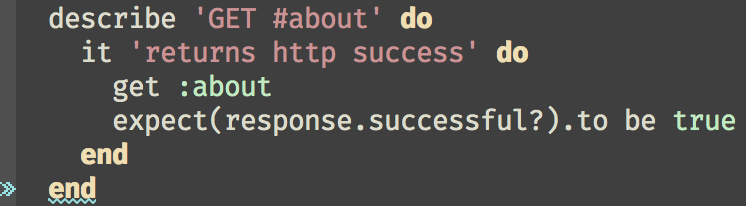


图 2 关于页面能够被访问到

|  |  |
| --- | --- |
| 测试用例序号 | 3 |
| 测试目的及说明 | 登录页面能够被访问到 |
| 测试输入 | 访问登录（Sessions#new）页面 |
| 期望测试结果 | 可以访问登录页面 |
| 判断准则 | 回复报文代码为200（正常） |
| 实际测试结果 | 通过 |
| 测试人员 | 徐智林 |
| 测试日期 | 2018年6月3日 |
| 测试评价 | 控制器相关测试 |
| 问题标识 | 无 |
| 问题处理 | 无 |

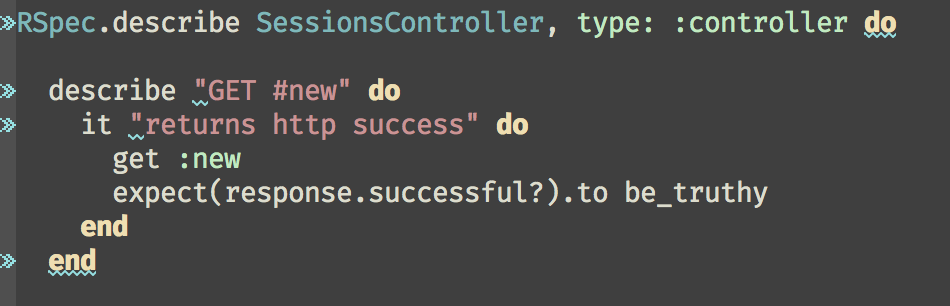
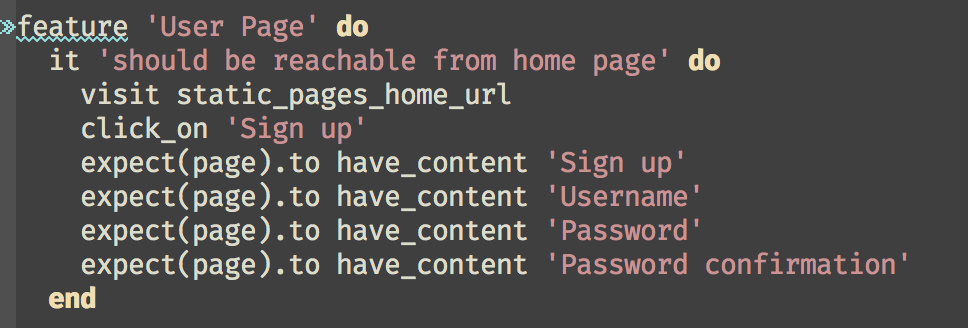
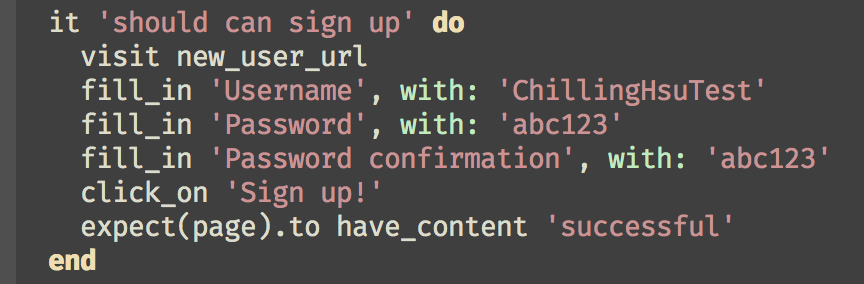


图 3 登录页面能够被访问到

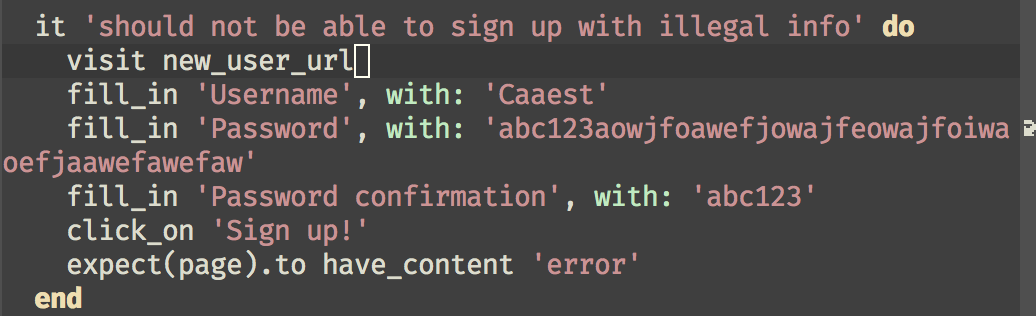
|  |  |
| --- | --- |
| 测试用例序号 | 4 |
| 测试目的及说明 | 用户注册页面能够从主页被正常访问 |
| 测试输入 | 在主页点击注册（Sign up）链接 |
| 期望测试结果 | 出现注册页面内容 |
| 判断准则 | 点击注册链接后页面内容中出现如下内容：   1. 注册（Sign up） 2. 用户名（Username） 3. 密码（Password） 4. 密码确认（Password confirmation） |
| 实际测试结果 | 通过 |
| 测试人员 | 徐智林 |
| 测试日期 | 2018年6月3日 |
| 测试评价 | 黑盒集成测试 |
| 问题标识 | 无 |
| 问题处理 | 无 |



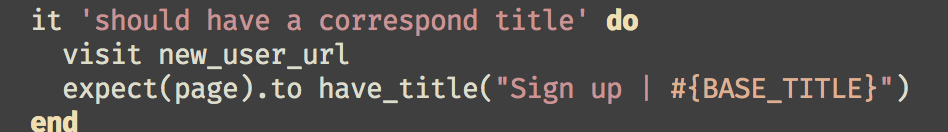
|  |  |
| --- | --- |
| 测试用例序号 | 4 |
| 测试目的及说明 | 用户能够正常注册 |
| 测试输入 | 1. 访问注册页面 2. 在对应输入栏内输入用户名（Username） 3. 在对应输入栏内输入密码（Password） 4. 在对应输入栏内输入密码确认（Password confirmation） 5. 点击注册 |
| 期望测试结果 | 出现注册成功提示信息 |
| 判断准则 | 点击注册提交后，页面出现注册成功的提示信息。（successful） |
| 实际测试结果 | 通过 |
| 测试人员 | 徐智林 |
| 测试日期 | 2018年6月3日 |
| 测试评价 | 黑盒集成测试 |
| 问题标识 | 无 |
| 问题处理 | 无 |



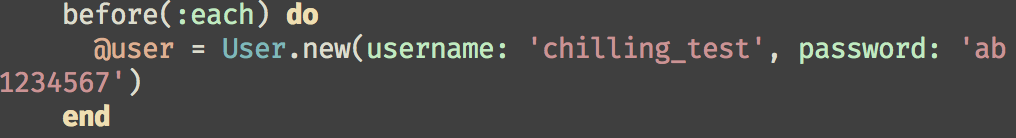
|  |  |
| --- | --- |
| 测试用例序号 | 5 |
| 测试目的及说明 | 当输入信息不合法时用户不能够正常注册 |
| 测试输入 | 1. 访问注册页面 2. 在对应输入栏内输入不合法用户名（Username） 3. 在对应输入栏内输入不合法密码（Password） 4. 在对应输入栏内输入不匹配密码确认（Password confirmation） 5. 点击注册 |
| 期望测试结果 | 出现响应错误信息 |
| 判断准则 | 点击注册提交后，页面出现注册错误的提示信息。（error） |
| 实际测试结果 | 通过 |
| 测试人员 | 徐智林 |
| 测试日期 | 2018年6月3日 |
| 测试评价 | 黑盒集成测试 |
| 问题标识 | 无 |
| 问题处理 | 无 |



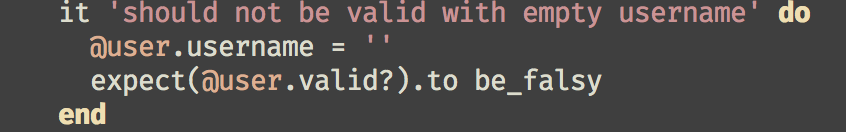
|  |  |
| --- | --- |
| 测试用例序号 | 6 |
| 测试目的及说明 | 注册页面的标题元素应含有正确内容 |
| 测试输入 | 访问注册页面 |
| 期望测试结果 | 标题元素包含Sign up字样 |
| 判断准则 | 点击注册提交后，页面标题包含正确字样（Sign up） |
| 实际测试结果 | 通过 |
| 测试人员 | 徐智林 |
| 测试日期 | 2018年6月3日 |
| 测试评价 | 黑盒集成测试 |
| 问题标识 | 无 |
| 问题处理 | 无 |



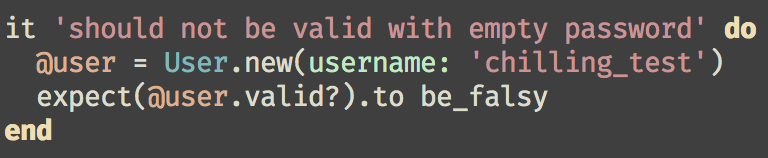
一个有效用户模型数据为，其后的测试以此为基础进行



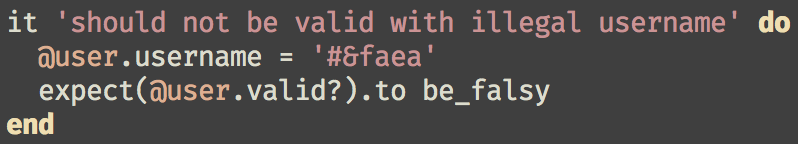
|  |  |
| --- | --- |
| 测试用例序号 | 7 |
| 测试目的及说明 | 用户模型（数据库表单）用户名不应接受空值 |
| 测试输入 | 含空用户名的用户数据 |
| 期望测试结果 | 无效 |
| 判断准则 | 通过valid?函数判断数据是否有效 |
| 实际测试结果 | 通过 |
| 测试人员 | 邓巧 |
| 测试日期 | 2018年6月1日 |
| 测试评价 | 白盒数据模型单元测试 |
| 问题标识 | 无 |
| 问题处理 | 无 |



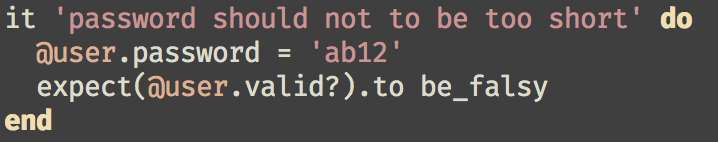
|  |  |
| --- | --- |
| 测试用例序号 | 8 |
| 测试目的及说明 | 用户模型（数据库表单）密码不应接受空值 |
| 测试输入 | 含空密码的用户数据 |
| 期望测试结果 | 无效 |
| 判断准则 | 通过valid?函数判断数据是否有效 |
| 实际测试结果 | 通过 |
| 测试人员 | 邓巧 |
| 测试日期 | 2018年6月1日 |
| 测试评价 | 白盒数据模型单元测试 |
| 问题标识 | 无 |
| 问题处理 | 无 |



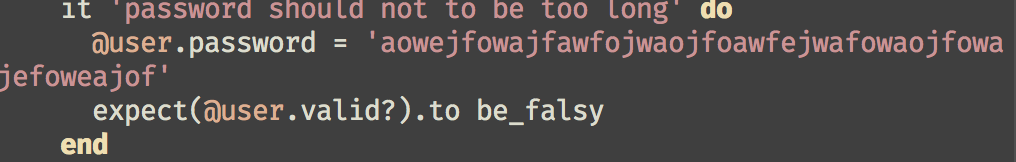
|  |  |
| --- | --- |
| 测试用例序号 | 8 |
| 测试目的及说明 | 用户模型（数据库表单）用户名不应接受含特殊符号的值。（只能包含字母、下划线和链接号） |
| 测试输入 | 用户名含非法字符的用户数据 |
| 期望测试结果 | 无效 |
| 判断准则 | 通过valid?函数判断数据是否有效 |
| 实际测试结果 | 通过 |
| 测试人员 | 邓巧 |
| 测试日期 | 2018年6月1日 |
| 测试评价 | 白盒数据模型单元测试 |
| 问题标识 | 无 |
| 问题处理 | 无 |



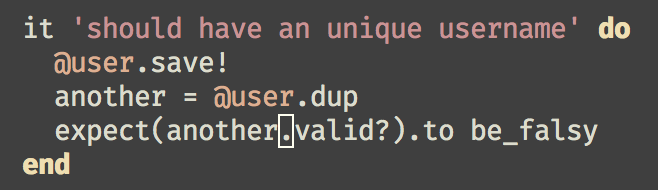
|  |  |
| --- | --- |
| 测试用例序号 | 9 |
| 测试目的及说明 | 用户模型（数据库表单）用户密码长度不应过短 |
| 测试输入 | 长度小于6的用户密码 |
| 期望测试结果 | 无效 |
| 判断准则 | 通过valid?函数判断数据是否有效 |
| 实际测试结果 | 无效 |
| 测试人员 | 邓巧 |
| 测试日期 | 2018年6月1日 |
| 测试评价 | 白盒数据模型单元测试 |
| 问题标识 | 无 |
| 问题处理 | 无 |



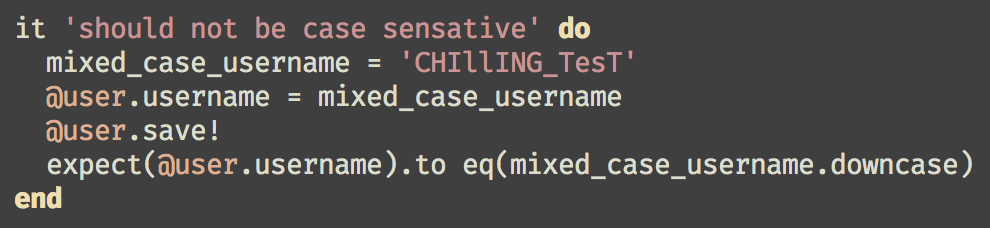
|  |  |
| --- | --- |
| 测试用例序号 | 10 |
| 测试目的及说明 | 用户模型（数据库表单）用户密码长度不应过长 |
| 测试输入 | 长度大于32的用户密码 |
| 期望测试结果 | 无效 |
| 判断准则 | 通过valid?函数判断数据是否有效 |
| 实际测试结果 | 通过 |
| 测试人员 | 邓巧 |
| 测试日期 | 2018年6月1日 |
| 测试评价 | 白盒数据模型单元测试 |
| 问题标识 | 无 |
| 问题处理 | 无 |



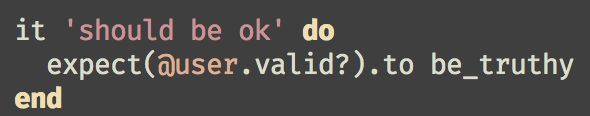
|  |  |
| --- | --- |
| 测试用例序号 | 11 |
| 测试目的及说明 | 用户模型（数据库表单）用户名不应重复 |
| 测试输入 | 数据库中已有的用户名 |
| 期望测试结果 | 无效 |
| 判断准则 | 通过valid?函数判断数据是否有效 |
| 实际测试结果 | 通过 |
| 测试人员 | 邓巧 |
| 测试日期 | 2018年6月1日 |
| 测试评价 | 白盒数据模型单元测试 |
| 问题标识 | 无 |
| 问题处理 | 无 |



|  |  |
| --- | --- |
| 测试用例序号 | 12 |
| 测试目的及说明 | 用户模型（数据库表单）用户名不应大小写敏感（统一转换为小写） |
| 测试输入 | 向数据库插入一个同时包含大小写的用户名的用户 |
| 期望测试结果 | 用户名变为全小写的形式 |
| 判断准则 | 通过从数据库中取出刚刚插入的用户，检查改用户名是否为全小写 |
| 实际测试结果 | 通过 |
| 测试人员 | 邓巧 |
| 测试日期 | 2018年6月1日 |
| 测试评价 | 白盒数据模型单元测试 |
| 问题标识 | 无 |
| 问题处理 | 无 |



|  |  |
| --- | --- |
| 测试用例序号 | 13 |
| 测试目的及说明 | 用户模型（数据库表单）用户名和密码符合规范应成功添加 |
| 测试输入 | 向数据库插入一个合法用户 |
| 期望测试结果 | 有效 |
| 判断准则 | 通过valid?函数判断数据是否有效 |
| 实际测试结果 | 有效 |
| 测试人员 | 邓巧 |
| 测试日期 | 2018年6月1日 |
| 测试评价 | 白盒数据模型单元测试 |
| 问题标识 | 无 |
| 问题处理 | 无 |



# 测试结果分析

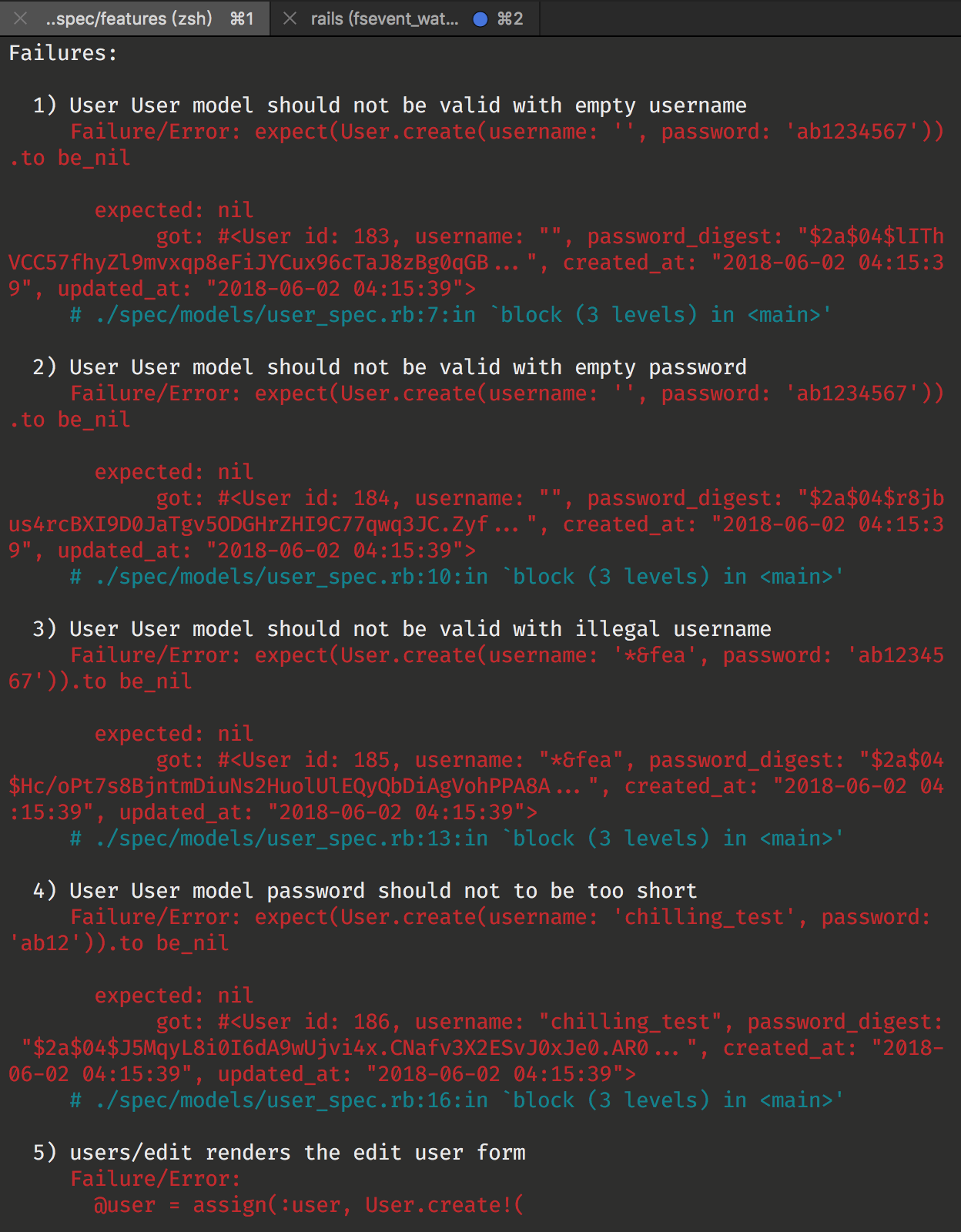
## 测试执行的充分性

在对用户数据信息有效性的测试方面（白盒测试）已比较充分，网页应用的操作逻辑（黑盒测试）也有所测试，但仍然需要充分设计用例。

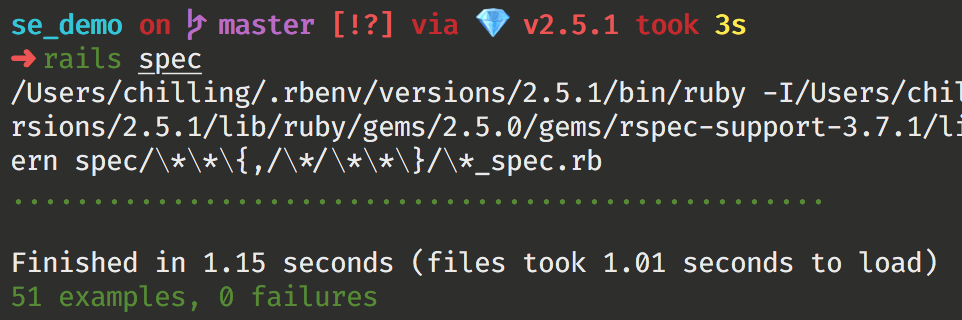
## 缺陷个数、分类及原因分析

使用了测试驱动开发（TDD）的方法，不断的进行，测试-开发循环，从最初的数据库设计，无法通过几乎所有的测试。然后不断的改进整个应用的使用逻辑。比如：

1. 设计更合理的操作逻辑（注册/登录/登出）
2. 设计相应的数据库类型限制和检查逻辑。



通过不断的修改程序，逐渐通过了所有的测试，达到了预期设计的所有效果。



## 遗留问题

没有对使用逻辑进行充分的黑盒测试。数据模型的白盒测试并没有十分完备。

# 测试结论

现有的应用的用户登入/登出/注册功能已经达到了预期设计的要求，数据库表的设计也能够完成所有的测试要求。基本已经完成了黑盒/白盒测试的目标。

**附录：源代码**

* + 1. 集成测试代码

|  |
| --- |
| require 'spec\_helper'  feature 'User Page' do  it 'should be reachable from home page' do  visit static\_pages\_home\_url  click\_on 'Sign up'  expect(page).to have\_content 'Sign up'  expect(page).to have\_content 'Username'  expect(page).to have\_content 'Password'  expect(page).to have\_content 'Password confirmation'  end  it 'should can sign up' do  visit new\_user\_url  fill\_in 'Username', with: 'ChillingHsuTest'  fill\_in 'Password', with: 'abc123'  fill\_in 'Password confirmation', with: 'abc123'  click\_on 'Sign up!'  expect(page).to have\_content 'successful'  end  it 'should not be able to sign up with illegal info' do  visit new\_user\_url  fill\_in 'Username', with: 'Caaest'  fill\_in 'Password', with: 'abc123aowjfoawefjowajfeowajfoiwaoefjaawefawefaw'  fill\_in 'Password confirmation', with: 'abc123'  click\_on 'Sign up!'  expect(page).to have\_content 'error'  end  it 'should have a correspond title' do  visit new\_user\_url  expect(page).to have\_title("Sign up | #{BASE\_TITLE}")  end  end |

* + 1. **单元测试代码**

|  |
| --- |
| require 'rails\_helper'  RSpec.describe User, type: :model do  # pending "add some examples to (or delete) #{\_\_FILE\_\_}"  context 'User model' do  before(:each) do  @user = User.new(username: 'chilling\_test', password: 'ab1234567')  end  it 'should not be valid with empty username' do  @user.username = ''  expect(@user.valid?).to be\_falsy  end  it 'should not be valid with empty password' do  @user = User.new(username: 'chilling\_test')  expect(@user.valid?).to be\_falsy  end  it 'should not be valid with illegal username' do  @user.username = '#&faea'  expect(@user.valid?).to be\_falsy  end  it 'password should not to be too short' do  @user.password = 'ab12'  expect(@user.valid?).to be\_falsy  end  it 'password should not to be too long' do  @user.password = 'aowejfowajfawfojwaojfoawfejwafowaojfowajefoweajof'  expect(@user.valid?).to be\_falsy  end  it 'should have an unique username' do  @user.save!  another = @user.dup  expect(another.valid?).to be\_falsy  end  it 'should not be case sensative' do  mixed\_case\_username = 'CHIllING\_TesT'  @user.username = mixed\_case\_username  @user.save!  expect(@user.username).to eq(mixed\_case\_username.downcase)  end  it 'should be ok' do  expect(@user.valid?).to be\_truthy  end  end  end |

* + 1. 首页应用逻辑测试代码

|  |
| --- |
| require 'spec\_helper'  require 'const\_helper'  feature 'Home Page' do  # before(:each) do  # @base\_title = 'Student Essentials'  # end  it 'should display welcome message' do  visit static\_pages\_home\_url  expect(page).to have\_content 'Student Essentials'  end  it 'should can reach to About Page' do  visit static\_pages\_home\_url  click\_link 'About'  # expect(response.successful?).to be true  expect(page.status\_code).to be 200  end  it 'should have link to login' do  visit static\_pages\_home\_url  expect(page).to have\_content 'Log in'  end  it 'should have a correspond title' do  visit static\_pages\_home\_url  expect(page).to have\_title("Home | #{BASE\_TITLE}")  end  end |

* + 1. 登入登出逻辑测试代码

|  |
| --- |
| require 'spec\_helper'  feature 'Sessions#Log in Page' do  it 'should be reachable from home page' do  visit static\_pages\_home\_url  click\_on 'Log in'  expect(page).to have\_content 'Log in'  expect(page).to have\_content 'Username'  expect(page).to have\_content 'Password'  end  it 'should can log in' do  User.create!(username: 'testtesttest', password: 'hellochilling')  visit login\_url  fill\_in 'Username', with: 'testtesttest'  fill\_in 'Password', with: 'hellochilling'  click\_on 'Log in!'  expect(page).not\_to have\_content 'invalid'  expect(page).not\_to have\_selector 'header .navbar', text: 'Log in'  expect(page).to have\_selector 'header .navbar', text: 'Testtesttest'  end  it 'should can log out' do  u = User.create!(username: 'testtest', password: 'hellochilling')  visit login\_url  fill\_in 'Username', with: 'testtest'  fill\_in 'Password', with: 'hellochilling'  click\_on 'Log in!'  visit root\_path  expect(page).not\_to have\_content 'Log in'  click\_button u.username.capitalize  click\_link 'Log out'  expect(page).to have\_content 'Log in'  expect(page).to have\_content 'Sign up'  end  it 'should not be able to log in with illegal info' do  visit login\_url  fill\_in 'Username', with: 'Caaest'  fill\_in 'Password', with: 'abc123aowjfoawefjowajfeowajfoiwaoefjaawefawefaw'  click\_on 'Log in!'  expect(page).to have\_content 'Invalid'  end  it 'should have a correspond title' do  visit login\_url  expect(page).to have\_title("Log in | #{BASE\_TITLE}")  end  end |

