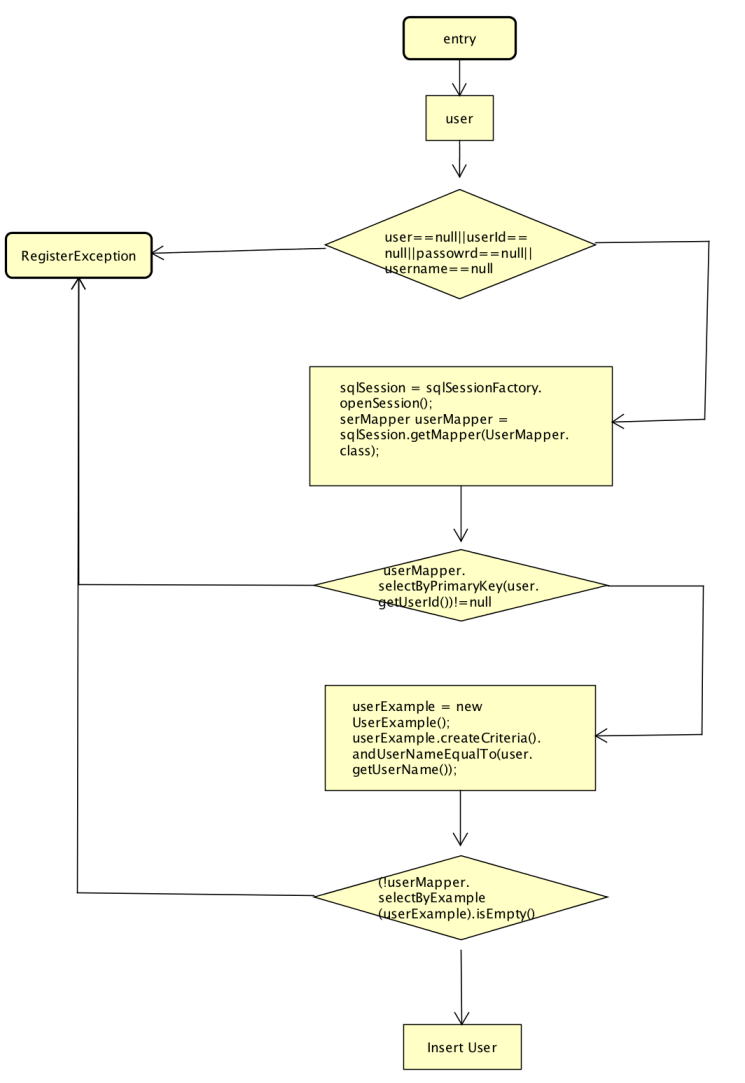
**测试文档**

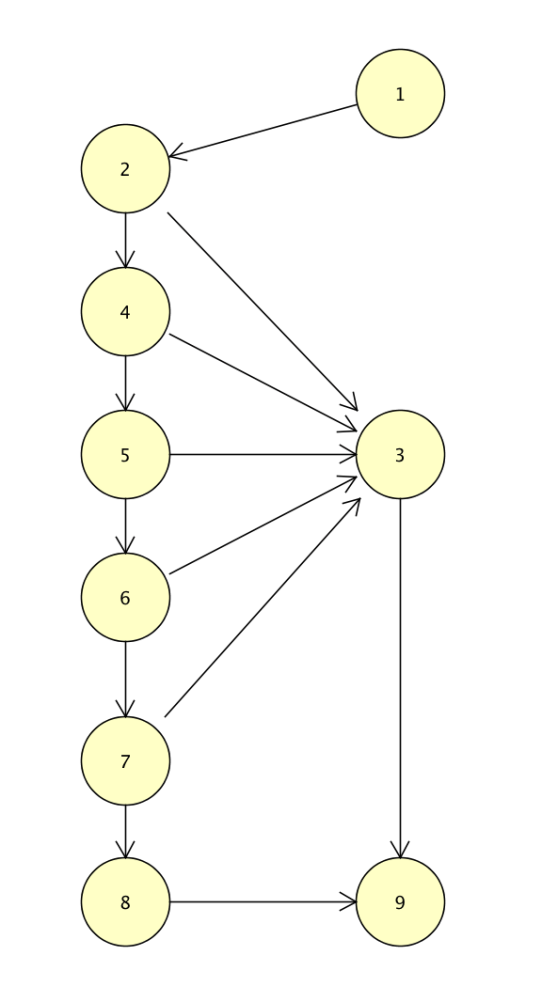
**1.白盒测试**

（1）用户注册流程测试

1. //如果传入user.user\_id为空
2. **if** (**null** == user || **null** == user.getUserId()||**null**==user.getUserPassword()||**null**==user.getUserName()){
4. **throw** **new** RegisterException(RegisterException.INVALID\_INPUT\_MESSAGE,RegisterException.INVALID\_INPUT);
6. }
8. SqlSession sqlSession = sqlSessionFactory.openSession();
9. UserMapper userMapper = sqlSession.getMapper(UserMapper.**class**);
11. //看看用户email是否已经存在,如果已经存在
12. **if** (**null** != userMapper.selectByPrimaryKey(user.getUserId())) {
14. **throw** **new** RegisterException(RegisterException.EMAIL\_HAS\_EXIST\_MESSAGE,RegisterException.EMAIL\_HAS\_EXIST);
15. }
17. //看看userName是否已经存在
18. UserExample userExample = **new** UserExample();
19. userExample.createCriteria().andUserNameEqualTo(user.getUserName());
20. **if** (!userMapper.selectByExample(userExample).isEmpty()){
21. **throw** **new** RegisterException(RegisterException.USERNAME\_HAS\_EXIST\_MESSAGE,RegisterException.USERNAME\_HAS\_EXIST);
22. }
24. //设置角色为普通用户
25. user.setRole(**false**);
26. //设置头像路径为
27. user.setHeadImg("boy.jpg");
28. //将用户密码进行加密后，保存到数据库中
29. user.setUserPassword(passwordEncoder.encode(user.getUserPassword()));
30. userMapper.insert(user);

33. sqlSession.close();
35. **return**;
36. 由程序流程图导出程序流图





程序流程图 程序流图

1. 计算需测试的基本路径数，通过计算环路复杂性来达到：

V(G)=E-N+2=13-9+2=6

1. 确定基本路径的集合

路径1: 1-2-3-9

路径2:1-2-4-3-9

路径3:1-2-4-5-3-9

路径4:1-2-4-5-6-3-9

路径5:1-2-4-5-6-7-3-9

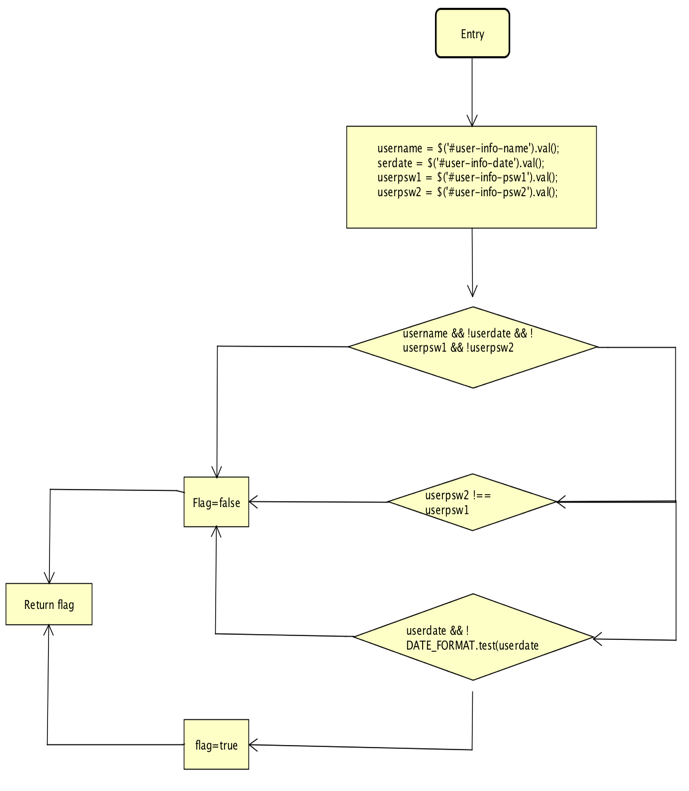
路径6:1-2-4-5-6-7-8-9

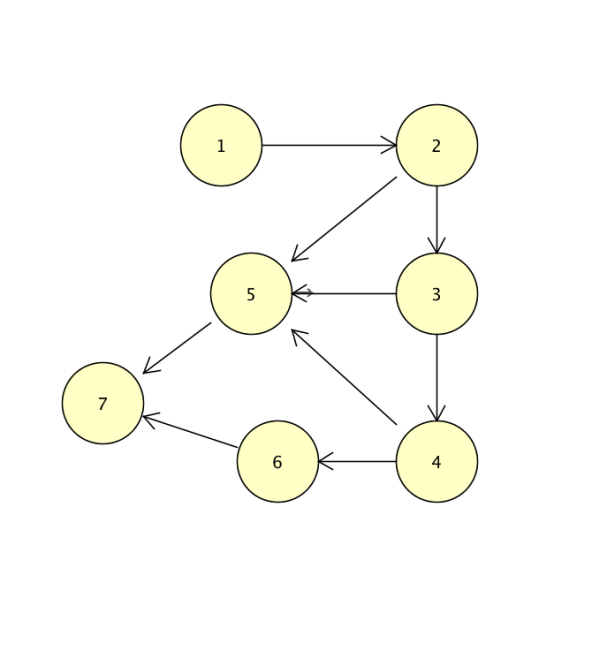
1. 根据基本路径设计测试用例

|  |  |  |
| --- | --- | --- |
| 路径 | 条件 | 结果 |
| 1 | user==null | RegisterException “无效输入” |
| 2 | user!=null&&userId==null | RegisterException “无效输入” |
| 3 | user!=null&&userId!=null  &&uername==null | RegisterException “无效输入” |
| 4 | user!=null&&userId!=null  &&uername!=null&&  email!=null | RegisterException “用户邮箱已经存在” |
| 5 | user!=null&&userId!=null&&uername  !=null&&email==null&&  userExample!=null | RegisterException “用户已经存在” |
| 6 | user!=null&&userId!=null&&uername  !=null&&email==null&&  userExample==null | 注册成功，用户设置默认信息，角色为user |

（2）检查用户信息

1. **function** check() {
2. **var** username = $('#user-info-name').val();
3. **var** userdate = $('#user-info-date').val();
4. **var** userpsw1 = $('#user-info-psw1').val();
5. **var** userpsw2 = $('#user-info-psw2').val();
6. **var** DATE\_FORMAT = /^[0-9]{4}-[0-1]?[0-9]{1}-[0-3]?[0-9]{1}$/; //日期的正则表达式
7. //对输入内容的判断
8. //至少得输入一项
9. **if** (!username && !userdate && !userpsw1 && !userpsw2) {
10. swal("", "请至少修改一项内容", "warning");
11. **return** **false**;
12. }
13. //密码得一样
14. **else** **if** (userpsw2 !== userpsw1) {
15. swal("", "两次密码输入不一致，请重试", "error");
16. **return** **false**;
17. }
18. //输入的格式不能错
19. **else** **if** (userdate && !DATE\_FORMAT.test(userdate)) {
20. swal("", "输入的日期格式有误，请按照xxxx-xx-xx的格式输入", "error");
21. **return** **false**;
22. }
24. }
25. 由程序流程图导出程序流图





程序流程图 程序流图

1. 计算需测试的基本路径数，通过计算环路复杂性来达到：

V(G)=E-N+2=9-7+2=4

1. 确定基本路径的集合

路径1: 1-2-5-7

路径2:1-2-3-5-7

路径3:1-2-3-4-5-7

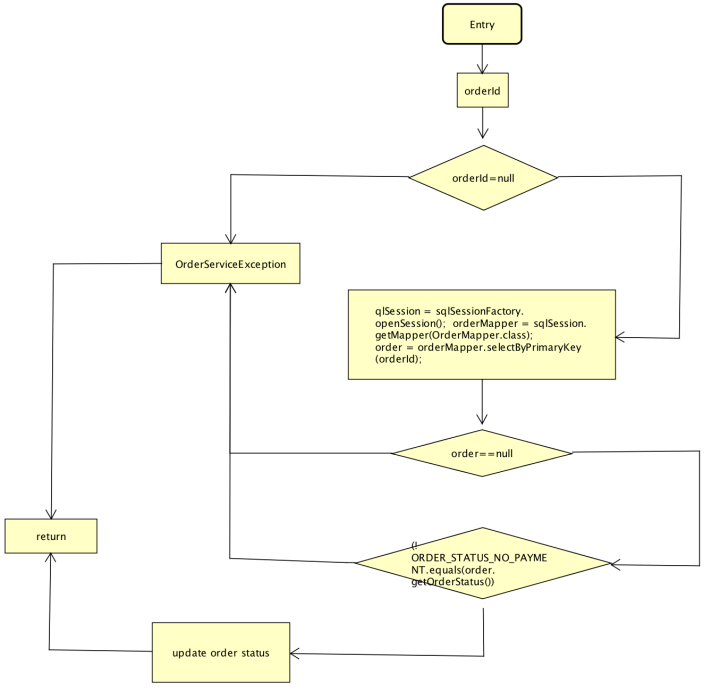
路径4:1-2-3-4-6-7

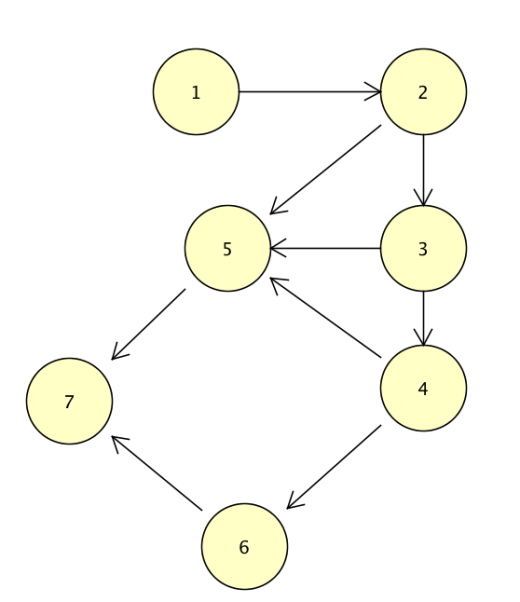
1. 根据基本路径设计测试用例

|  |  |  |
| --- | --- | --- |
| 路径 | 条件 | 结果 |
| 1 | username、userdate、passwd、passwd2至少有一项为null | false |
| 2 | username、userdate、passwd、passwd2均不null&&passwd!=passwd2 | false |
| 3 | username、userdate、passwd、passwd2均不null&&passwd==passwd2&&email格式不正确 | false |
| 4 | username、userdate、passwd、passwd2均不null&&passwd==passwd2&&email格式正确 | true |

（3）订单支付状态管理

1. **public** **void** payOrderByOrderId(String orderId) {
3. //检查输入
4. **if** (**null** == orderId) {
6. **throw** **new** OrderServiceException(INVALID\_INPUT\_MESSAGE, INVALID\_INPUT);
7. }
9. SqlSession sqlSession = sqlSessionFactory.openSession();
10. OrderMapper orderMapper = sqlSession.getMapper(OrderMapper.**class**);
12. Order order = orderMapper.selectByPrimaryKey(orderId);
14. //判断订单是否存在
15. **if** (**null** == order) {
16. **throw** **new** OrderServiceException(OrderServiceException.NO\_ORDERS\_MESSAGE, OrderServiceException.NO\_ORDERS);
17. }
19. //判断订单是否未支付
20. **if** (!ORDER\_STATUS\_NO\_PAYMENT.equals(order.getOrderStatus())){
21. **throw** **new** OrderServiceException(OrderServiceException.NO\_UNPAY\_MESSAGE,OrderServiceException.NO\_UNPAY);
22. }
24. //更新订单状态为已支付
25. order.setOrderStatus(ORDER\_STATUS\_HAS\_PAYMENT);
26. orderMapper.updateByPrimaryKey(order);
28. sqlSession.close();
30. **return**;
31. }
32. 由程序流程图导出程序流图





程序流程图 程序流图

1. 计算需测试的基本路径数，通过计算环路复杂性来达到：

V(G)=E-N+2=9-7+2=4

1. 确定基本路径的集合

路径1: 1-2-5-7

路径2:1-2-3-5-7

路径3:1-2-3-4-5-7

路径4:1-2-3-4-6-7

1. 根据基本路径设计测试用例

|  |  |  |
| --- | --- | --- |
| 路径 | 条件 | 结果 |
| 1 | orderId==null | OrderServiceException “订单编号不存在” |
| 2 | orderId!=null&&order==null | OrderServiceException “订单不存在” |
| 3 | orderId!=null&&order!=null&&  orderStatus!=  ORDER\_STATUS\_NO\_PAYMENT | OrderServiceException “订单状态已支付” |
| 4 | orderId!=null&&order!=null&&  orderStatus==  ORDER\_STATUS\_NO\_PAYMENT | 改变订单状态成功 |

**2.黑盒测试**

**（1）注册**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 需求 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 用户名 | 非空 | (1) | 空 | (5) |
| 邮箱地址 | 邮箱格式 | (2) | 非邮箱格式 | (6） |
| 空 | (7) |
| 密码 | 非空 | (3) | 空 | (8) |
| 确认密码 | 与前一次输入一致 | (4) | 与前一次输入不一致 | (9) |
| 空 | (10) |

**测试用例**

1、用户名

|  |  |  |
| --- | --- | --- |
| 测试数据 | 期望结果 | 覆盖范围 |
| 李白 | 有效 | 等价类（1） |
| 空 | 无效 | 等价类（5） |

1. 邮箱地址

|  |  |  |
| --- | --- | --- |
| 测试数据 | 期望结果 | 覆盖范围 |
| 123@.com | 有效 | 等价类（2） |
| 空 | 无效 | 等价类（7） |
| 123321 | 无效 | 等价类（6） |

1. 密码

|  |  |  |
| --- | --- | --- |
| 测试数据 | 期望结果 | 覆盖范围 |
| 123 | 有效 | 等价类（3） |
| 空 | 无效 | 等价类（8） |

1. 确认密码

|  |  |  |
| --- | --- | --- |
| 测试数据 | 期望结果 | 覆盖范围 |
| 123 | 有效 | 等价类（4） |
| 空 | 无效 | 等价类（10） |
| 321 | 无效 | 等价类（9） |

**（2）结算**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 需求 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 商品种类 | 选中商品 | (1) | 无选中 | (5) |
| 商品数量 | 非空 | (2) | 0个 | (6） |
| 超出库存 | (7) |
| 地址 | 非空 | (3) | 空 | (8) |

**测试用例**

1、商品种类

|  |  |  |
| --- | --- | --- |
| 测试数据 | 期望结果 | 覆盖范围 |
| 选中格力空调 | 结算 | 等价类（1） |
| 无选中 | 无法结算 | 等价类（5） |

2、商品数量

|  |  |  |
| --- | --- | --- |
| 测试数据 | 期望结果 | 覆盖范围 |
| 添加格力空调23件 | 结算 | 等价类（2） |
| 删除空调为0件 | 无法结算 | 等价类（6） |
| 添加空调为999件 | 超出库存 | 等价类（7） |

3、地址

|  |  |  |
| --- | --- | --- |
| 测试数据 | 期望结果 | 覆盖范围 |
| 选中 | 结算 | 等价类（3） |
| 无选中 | 无法结算 | 等价类（8） |

**（3）用户信息修改**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 需求 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 头像 | 图片格式 | (1) | 非图片格式 | (5) |
| 生日 | 指定格式 | (2) | 非指定格式 | (6） |
| 空 | (7) |
| 确认密码 | 与前一次输入一致 | (3) | 与前一次输入不一致 | (9) |
| 空 | (10) |
| 修改项目 | 至少一项 | (4) | 未修改项目 | (11) |

**测试用例**

1、头像

|  |  |  |
| --- | --- | --- |
| 测试数据 | 期望结果 | 覆盖范围 |
| 头像.Jpeg | 成功 | 等价类（1） |
| 文档.txt | 错误 | 等价类（5） |

2、生日

|  |  |  |
| --- | --- | --- |
| 测试数据 | 期望结果 | 覆盖范围 |
| 2009-09-09 | 成功 | 等价类（2） |
| 123444 | 格式有误 | 等价类（6） |

3、确认密码

|  |  |  |
| --- | --- | --- |
| 测试数据 | 期望结果 | 覆盖范围 |
| 输入一致123 | 成功 | 等价类（3） |
| 1234 | 错误 | 等价类（9） |
| 空 | 错误 | 等价类（10） |

4、修改项目

|  |  |  |
| --- | --- | --- |
| 测试数据 | 期望结果 | 覆盖范围 |
| 修改头像 | 成功 | 等价类（4） |
| 未修改项目 | 错误 | 等价类（11） |