**实 验 报 告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **课程名称** | 软件测试 | | | | | |
| **实验项目名称** | **软件测试试验报告1** | | | | | |
| **实验时间**  **（日期及节次）** | **2024.06.05 7-8节** | | | | | |
| **专业** | **软件工程** | | **学生所在学院** | | **软件学院** | |
| **年级** | **2022** | | **学号** | | **20225958** | |
| **姓名** | **李嘉富** | | **指导教师** | | **王楠** | |
| **实验室名称** | **4-505** | | | | | |
| **实验成绩** | **预习情况** | **操作技术** | **实验报告** | **附加：综合创新能力** | | **实验**  **综合成绩** |
|  |  |  |  | |  |
| **教师签字** |  | | | | | |

**黑龙江大学教务处**

## 程序流程图

程序流程图如图1

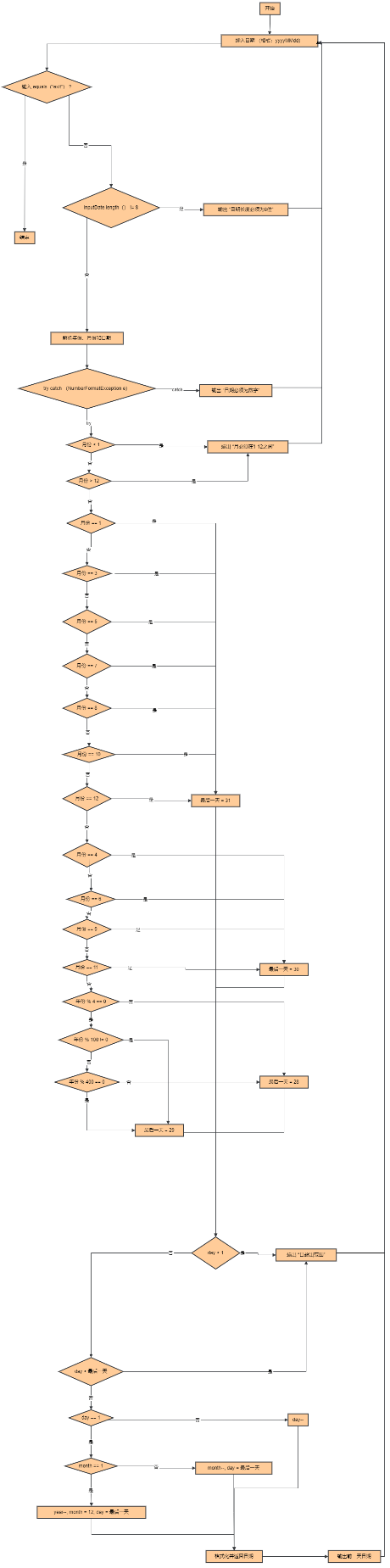


图 1

## 程序控制流图

程序控制流图如图2

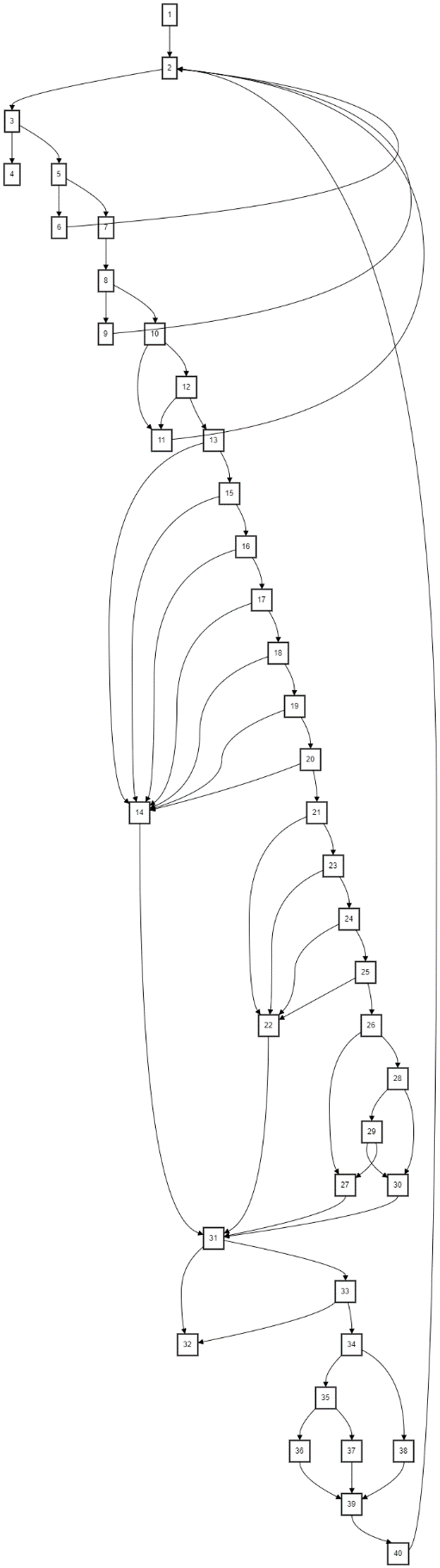


图 2

## 计算圈复杂度

根据上述控制流图，圈复杂度 V(G) 计算如下：

* 节点数 N = 40
* 边数 E = 61
* 判断节点数P=22

所以圈复杂度 V(G) 为：

V(G)=E−N+2=61−40+2=23

或V(G)=P+1=22+1=23

## 独立路径

* 路径1: 1 → 2 → 3 → 4
* 路径2: 1 → 2 → 3 → 5 → 6 → 2
* 路径3: 1 → 2 → 3 → 5 → 7 → 8 → 9 → 2
* 路径4: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 11 → 2
* 路径5: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 12 → 11 → 2
* 路径6: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 12 → 13 → 14 → 31 → 32 → 2
* 路径7: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 12 → 13 → 15 → 14 → 31 → 32 → 2
* 路径8: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 12 → 13 → 15 → 16 → 14 → 31 → 32 → 2
* 路径9: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 12 → 13 → 15 → 16 → 17 → 14 → 31 → 32 → 2
* 路径10: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 12 → 13 → 15 → 16 → 17 → 18 → 14 → 31 → 32 → 2
* 路径11: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 12 → 13 → 15 → 16 → 17 → 18 → 19 → 14 → 31 → 32 → 2
* 路径12: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 12 → 13 → 15 → 16 → 17 → 18 → 19 → 20 → 14 → 31 → 32 → 2
* 路径13: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 12 → 13 → 15 → 16 → 17 → 18 → 19 → 20 → 21 → 22 → 31 → 32 → 2
* 路径14: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 12 → 13 → 15 → 16 → 17 → 18 → 19 → 20 → 21 → 23 → 22 → 31 → 32 → 2
* 路径15: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 12 → 13 → 15 → 16 → 17 → 18 → 19 → 20 → 21 → 23 → 24 → 22 → 31 → 32 → 2
* 路径16: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 12 → 13 → 15 → 16 → 17 → 18 → 19 → 20 → 21 → 23 → 24 → 25 → 22 → 31 → 32 → 2
* 路径17: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 12 → 13 → 15 → 16 → 17 → 18 → 19 → 20 → 21 → 23 → 24 → 25 → 26 → 27 → 31 → 32 → 2
* 路径18: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 12 → 13 → 15 → 16 → 17 → 18 → 19 → 20 → 21 → 23 → 24 → 25 → 26 → 28 → 29 → 30 → 31 → 32 → 2
* 路径19: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 12 → 13 → 15 → 16 → 17 → 18 → 19 → 20 → 21 → 23 → 24 → 25 → 26 → 28 → 30 → 31 → 32 → 2
* 路径20: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 12 → 13 → 15 → 16 → 17 → 18 → 19 → 20 → 21 → 23 → 24 → 25 → 26 → 29 → 30 → 31 → 32 → 2
* 路径21: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 12 → 13 → 15 → 16 → 17 → 18 → 19 → 20 → 21 → 23 → 24 → 25 → 26 → 29 → 27 → 31 → 32 → 2
* 路径22: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 12 → 13 → 15 → 16 → 17 → 18 → 19 → 20 → 21 → 23 → 24 → 25 → 26 → 27 → 31 → 32 → 2
* 路径23: 1 → 2 → 3 → 5 → 7 → 8 → 10 → 12 → 13 → 15 → 16 → 17 → 18 → 19 → 20 → 21 → 23 → 24 → 25 → 26 → 28 → 29 → 30 → 31 → 33 → 34 → 35 → 36 → 39 → 40 → 2

## 测试用例

测试用例表如表1

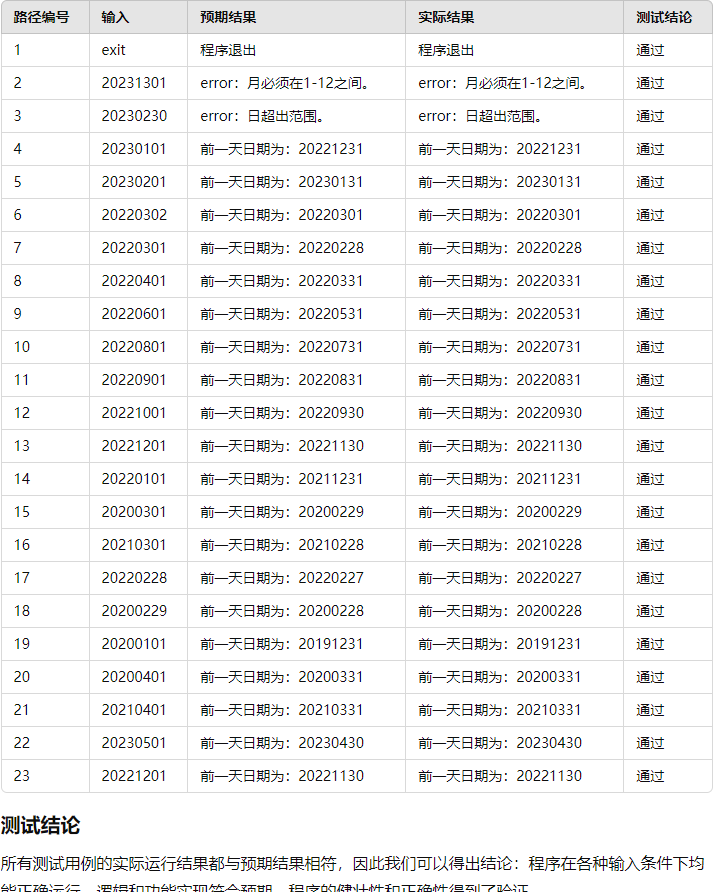
表1 测试用例表



## 测试结果和结论

测试结果如表2

表2 测试结果

测试结论

所有测试用例的实际运行结果都与预期结果相符，因此我们可以得出结论：程序在各种输入条件下均能正确运行，逻辑和功能实现符合预期。程序的健壮性和正确性得到了验证。

附录1源代码（李嘉富20225958）：

import java.util.Scanner;

/\*\*

\* 程序一

\*/

public class PreDate {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.print("请输入日期（格式：yyyyMMdd）：");

String inputDate = scanner.nextLine();

if ("exit".equals(inputDate)) {

break;

}

if (isValidDate(inputDate)) {

String preDate = getPreDate(inputDate);

System.out.println("前一天日期为：" + preDate);

} else {

System.out.println("日期格式错误，请重新输入。");

}

}

scanner.close();

}

/\*\*

\* 判断输入的日期是否合法

\*

\* @param inputDate 输入的日期

\* @return 是否合法

\*/

public static boolean isValidDate(String inputDate) {

if (inputDate.length() != 8) {

System.out.println("error：日期长度必须为8位。");

return false;

}

try {

int year = Integer.parseInt(inputDate.substring(0, 4));

int month = Integer.parseInt(inputDate.substring(4, 6));

int day = Integer.parseInt(inputDate.substring(6));

if (month < 1 || month > 12) {

System.out.println("error：月必须在1-12之间。");

}

if (day < 1 || day > getLastDayOfMonth(year, month)) {

System.out.println("error：日超出范围。");

}

if (month < 1 || month > 12 || day < 1 || day > getLastDayOfMonth(year, month)) {

return false;

}

} catch (NumberFormatException e) {

System.out.println("error：日期必须为数字。");

return false;

}

return true;

}

/\*\*

\* 获取输入日期的前一天日期

\*

\* @param inputDate 输入日期

\* @return 前一天日期

\*/

public static String getPreDate(String inputDate) {

int year = Integer.parseInt(inputDate.substring(0, 4));

int month = Integer.parseInt(inputDate.substring(4, 6));

int day = Integer.parseInt(inputDate.substring(6));

if (day == 1) {

if (month == 1) {

year--;

month = 12;

} else {

month--;

}

day = getLastDayOfMonth(year, month);

} else {

day--;

}

return String.format("%04d%02d%02d", year, month, day);

}

/\*\*

\* 获取指定年月的最后一天

\*

\* @param year 年

\* @param month 月

\* @return 最后一天

\*/

private static int getLastDayOfMonth(int year, int month) {

switch (month) {

case 1, 3, 5, 7, 8, 10, 12 -> {

return 31;

}

case 4, 6, 9, 11 -> {

return 30;

}

default -> {

if (isLeapYear(year)) {

return 29;

} else {

return 28;

}

}

}

}

/\*\*

\* 判断是否为闰年

\*

\* @param year 年

\* @return 是否为闰年

\*/

private static boolean isLeapYear(int year) {

return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);

}

}

**实 验 报 告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **课程名称** | 软件测试 | | | | | |
| **实验项目名称** | **软件测试试验报告2** | | | | | |
| **实验时间**  **（日期及节次）** | **2024.06.12 7-8节** | | | | | |
| **专业** | **软件工程** | | **学生所在学院** | | **软件学院** | |
| **年级** | **2022** | | **学号** | | **20225958** | |
| **姓名** | **李嘉富** | | **指导教师** | | **王楠** | |
| **实验室名称** | **4-505** | | | | | |
| **实验成绩** | **预习情况** | **操作技术** | **实验报告** | **附加：综合创新能力** | | **实验**  **综合成绩** |
|  |  |  |  | |  |
| **教师签字** |  | | | | | |

**黑龙江大学教务处**

## 等价类划分

等价类划分表如表1

表1 等价类划分表



## 测试用例

测试用例表如表2

表2 测试用例表



## 测试结果和结论

测试结果如表3

表3 测试结果



测试结论

通过测试可以看出，程序在处理各种合法、不合法电话号码及无效输入时表现良好。它能够有效地检查电话号码的长度、格式以及是否包含非数字字符，并根据不同的错误情况输出相应的提示信息，帮助用户了解输入电话号码的问题所在。

附录1源代码（李嘉富20225958）：

import java.util.Scanner;

/\*\*

\* 程序二

\*/

public class PhoneNumberValidator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.print("请输入电话号码:");

String phoneNumber = scanner.nextLine();

phoneNumber = phoneNumber.replaceAll("\\s", "");

if ("exit".equals(phoneNumber)) {

break;

}

boolean isValid = validatePhoneNumber(phoneNumber);

System.out.println(isValid ? "电话号码格式正确":"电话号码格式错误");

}

scanner.close();

}

/\*\*

\* 验证电话号码格式是否正确

\*

\* @param phoneNumber 电话号码

\* @return 是否正确

\*/

public static boolean validatePhoneNumber(String phoneNumber) {、

// 检查电话号码是否为空白

if (phoneNumber.isBlank()) {

System.out.println("电话号码不能为空。");

return false;

}

// 检查输入是否都是数字

if (!phoneNumber.matches("\\d+")) {

System.out.println("电话号码必须为数字。");

return false;

}

// 检查电话号码是否为空白或不是7位或10位

if ((phoneNumber.length() != 7 && phoneNumber.length() != 10)) {

System.out.println("长度错误，请输入7位或10位电话号码。");

return false;

}

int skew = (phoneNumber.length() == 7) ? 0 : 3;

// 检查地区码是否为空白或三位数字

if (phoneNumber.length() == 10 && !phoneNumber.substring(0, 3).matches("\\d{3}")) {

System.out.println("地区码错误，请输入三位数字。");

return false;

}

// 检查前缀码是否非"0"或”1”开头的三位数字

if (!phoneNumber.substring(skew, 3 + skew).matches("^(?![01])\\d{3}$")) {

System.out.println("前缀码错误，请输入非0或1开头的三位数字。");

return false;

}

// 检查后缀码是否为四位数字

if (!phoneNumber.substring(3 + skew, 7 + skew).matches("\\d{4}")) {

System.out.println("后缀码错误，请输入四位数字。");

return false;

}

return true;

}

}

**实 验 报 告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **课程名称** | 软件测试 | | | | | |
| **实验项目名称** | **软件测试试验报告3** | | | | | |
| **实验时间**  **（日期及节次）** | **2024.06.19 7-8节** | | | | | |
| **专业** | **软件工程** | | **学生所在学院** | | **软件学院** | |
| **年级** | **2022** | | **学号** | | **20225958** | |
| **姓名** | **李嘉富** | | **指导教师** | | **王楠** | |
| **实验室名称** | **4-505** | | | | | |
| **实验成绩** | **预习情况** | **操作技术** | **实验报告** | **附加：综合创新能力** | | **实验**  **综合成绩** |
|  |  |  |  | |  |
| **教师签字** |  | | | | | |

**黑龙江大学教务处**

## 等价类划分

等价类划分表如表1

表1 等价类划分表



## 测试用例

测试用例表如表2

表2 测试用例表



## 测试结果和结论

测试结果如表3

表3 测试结果



#### 测试结论

通过边界值分析法进行测试，所有测试用例均通过，程序能够正确处理合法输入并给出正确的中文大写金额，同时对非法输入给出适当的错误提示或处理措施。

## 输入域的制约条件及因果图法分析

#### 制约条件

1. 整数部分长度限制：整数部分的长度不能超过12位。

2. 小数部分长度限制：小数部分的长度不能超过2位。

3. 负数限制：输入不能为负数。

4. 非数字字符限制：输入必须为数字字符，不能包含字母或特殊符号。

#### 因果图法分析

● 当整数部分长度超过12位时，程序应输出错误信息并提示重新输入。

● 当小数部分长度超过2位时，程序应输出错误信息并提示重新输入。

● 当输入为负数时，程序应输出负数提示并退出。

● 当输入包含非数字字符时，程序应输出无效输入提示并重新输入。

附录1源代码（李嘉富20225958）：

import java.math.BigDecimal;

import java.util.Scanner;

public class RMBAmountConverter {

private static final String[] CN\_NUMBERS = {"零", "壹", "贰", "叁", "肆", "伍", "陆", "柒", "捌", "玖"};

private static final String[] CN\_UNITS = {"", "拾", "佰", "仟", "万", "亿"};

private static final String CN\_YUAN = "元";

private static final String CN\_JIAO = "角";

private static final String CN\_FEN = "分";

private static final String CN\_ZHENG = "整";

/\*\*

\* 将金额转换为中文大写

\*

\* @param amount 金额

\* @return 中文大写金额

\*/

public static String convertToCN(BigDecimal amount) {

StringBuilder result = new StringBuilder("人民币");

long intValue = amount.longValue();

int decimalPart = amount.movePointRight(2).intValue() % 100;

if (intValue == 0) {

result.append(CN\_NUMBERS[0]).append(CN\_YUAN);

} else {

int unitIndex = 0;

while (intValue > 0) {

long digit = intValue % 10;

if (digit != 0) {

result.insert(2, CN\_NUMBERS[(int) digit] + CN\_UNITS[unitIndex]);

} else {

if (result.charAt(2) != '零') {

result.insert(2, CN\_NUMBERS[0]);

}

}

intValue /= 10;

unitIndex++;

}

result.append(CN\_YUAN);

}

if (decimalPart > 0) {

result.append(CN\_NUMBERS[decimalPart / 10]).append(CN\_JIAO);

if (decimalPart % 10 > 0) {

result.append(CN\_NUMBERS[decimalPart % 10]).append(CN\_FEN);

}

} else {

result.append(CN\_ZHENG);

}

return result.toString();

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.print("请输入金额：");

BigDecimal amount;

try {

amount = scanner.nextBigDecimal();

if (amount.compareTo(BigDecimal.ZERO) < 0) {

System.out.println("金额为负数，程序退出。");

break;

}

} catch (Exception e) {

System.out.println("输入无效，请重新输入。");

scanner.nextLine(); // 清除输入缓冲区

continue;

}

System.out.println("中文大写金额：\t" + convertToCN(amount));

}

}

}

**实 验 报 告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **课程名称** | **软件质量保证与测试** | | | | | |
| **实验项目名称** | **软件测试试验报告** | | | | | |
| **实验时间**  **（日期及节次）** | **2024.06.26 星期三5-6节** | | | | | |
| **专业** | **软件工程** | | **学生所在学院** | | **软件学院** | |
| **年级** | **2022级** | | **学号** | | **20225958** | |
| **姓名** | **李嘉富** | | **指导教师** | | **王楠** | |
| **实验室名称** | **4号楼-[520]软件3机房** | | | | | |
| **实验成绩** | **预习情况** | **操作技术** | **实验报告** | **附加：综合创新能力** | | **实验**  **综合成绩** |
|  |  |  |  | |  |
| **教师签字** |  | | | | | |

**黑龙江大学教务处**

1. 引言

1.1编写目的

此测试分析报告是为了对人事管理系统的设计、编程和使用进行测试考核，以确定其软件本身是否有缺陷和不足以及可能给软件运行带来的影响，改正系统BUG，并对其测试结果做出结论。

1.2项目背景

随着企业，机构的日渐庞大，需要一套能够管理人事的系统

1. 任务概述

2.1目标

详细描述对该人事观念系统进行测试的测试过程。

2.2运行环境

操作系统: win10

软件：idea vscode

支持环境 jdk1.8

2.3需求概述

本次测试针对开发的人事管理管理系统进行，包括功能测试，界面测试，负载测试，文档测试。按照规格需求说明书中的功能进行测试，在测试过程中发现软件的漏洞不足并予以改正。

2.4条件与限制

本次测试采用的主要是黑盒测试，对系统各个功能模块进行测试，设计合理的测试用例即可，没有其他特殊的要求。

1. 计划

3.1测试方案

采用黑盒测试方法。对功能进行逐一测试，在输入合理及不合理的数据后测试系统的正常运作情况。

3.2测试项目

测试1:登录测试

测试2:新增员工信息

测试3:删除员工信息

测试4:新增员工工资信息

测试5:新增管理员信息

3.3测试准备

1 系统登录测试

等价类划分

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 账号 | 管理员账号  普通用户账号 | 1  2 | 账号不存在  输入为空 | 5  6 |
| 密码 | 密码正确 | 3 | 错误的密码  空白账号 | 7  8 |
| 验证码 | 验证码正确 | 4 | 错误的验证码  空白的验证码 | 9  10 |

测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 账号 | 密码 | 验证码 | 覆盖等价类 | 预期输出 |
| 1 | boss | boss | 与图片一致 | 1,3,4 | 跳到首页 |
| 2 | sunqjao | sunqjao | 与图片一致 | 2,3,4 | 跳到首页 |
| 3 | sunjao | sunqjao | 与图片一致 | 5,3,4 | 账号或密码错误 |
| 4 |  | sunqjao | 与图片一致 | 6,3,4 | 账号能为空 |
| 5 | sunqjao | sunao | 与图片一致 | 2,7,9 | 账号或密码错误 |
| 6 | sunqjao |  | 与图片一致 | 2,8,9 | 密码不能为空 |
| 7 | sunqjao | sunqjao | 与图片不一致 | 2,3,9 | 验证码错误 |
| 8 | sunqjao | sunqjao |  | 2.3,10 | 验证码不嫩为空 |

2 新增员工

等价类划分

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 名字 | 长度大于0小于等于20 | 1 | 长度大于20  输入为空 | 8  9 |
| 电话 | 输入纯数字  长度为11  第一位为1  第二位为[3~8] | 3  4  5  6 | 输入特殊字符  长度不为11  长度为空  第一位不为1  第二位不在3~8之间 | 10  11  12  13  14 |
| 部门 | 部门存在 | 7 | 部门不存在  部门为空 | 15  16 |

测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 名字 | 电话 | 部门 | 覆盖等价类 | 预期输出 |
| 1 | 张三 | 18745677889 | 研发 | 1,3,4,5,6,7 | 添加成功 |
| 2 | Aaaaaaaaaaaaaaaaaaaaa  aaaaaaaaaaaaaaaaaaaaa | 18745677889 | 研发 | 8,3,4,5,6,7 | 名字过长 |
| 3 |  | 18745677889 | 研发 | 9,3,4,5,6,7 | 名字为空 |
| 4 | 张三 | 1874567788a | 研发 | 1,10,4,5,6,7 | 电话格式非法 |
| 5 | 张三 | 187456778899 | 研发 | 1,3,11,5,6,7 | 电话格式非法 |
| 6 | 张三 |  | 研发 | 1,12,7 | 电话不能为空 |
| 7 | 张三 | 28745677889 | 研发 | 1,3,4,13,6,7 | 电话格式错误 |
| 8 | 张三 | 10745677889 | 研发 | 1,3,4,5,14,7 | 电话格式错误 |
| 9 | 张三 | 18745677889 |  | 1,3,4,5,6,15 | 部门不能为空 |
| 10 | 张三 | 18745677889 | 研发 | 1,3,4,5,6,16 | 部门不存在 |

3 删除员工

等价类划分

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 操作员 | 系统管理员 | 1 | 普通用户 | 3 |
| 被删除者 | 存在的用户 | 2 | 不存在用户 | 4 |

测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 操作员 | 删除的用户 | 覆盖等价类 | 预期输出 |
| 1 | boss | 张三 | 1,2 | 添加成功 |
| 2 | sunqjao | 张三 | 3，2 | 权限不足 |
| 3 | boss | 张四 | 1，4 | 用户不存在 |

4 新增员工工资信息

等价类划分

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 薪资 | 纯数字组成  非空  大于0小于20000 | 1  2  3 | 含有特殊字符  输入为空  小于0  大于20000 | 7  8  9  10 |
| 员工id | 输入纯数字  员工id不为空  员工存在 | 4  5  6 | 包含特殊字符  输入为空  员工不存在 | 11  12  13 |

测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 薪资 | 员工 | 覆盖等价类 | 预期输出 |
| 1 | 1000 | 1 | 1,2,3,4,5,6 | 添加成功 |
| 2 | 1000e | 1 | 7,2,3,4,5,6 | 薪资非法 |
| 3 |  | 1 | 1,8,3,4,5,6 | 薪资不能为空 |
| 4 | -1 | 1 | 1,2,9,4,5,6 | 薪资非法 |
| 5 | 200000 | 1 | 1,2,10,4,5,6 | 工资非法 |
| 6 | 1000 | a | 1,2,3,11,5,6 | 员工不存在 |
| 7 | 1000 |  | 1,2,3,4,12,6 | 员工不能为空 |
| 8 | 1000 | 11111 | 1,2,3,4,5,13 | 员工不存在 |

5 新增账号信息

等价类划分

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 用户名 | 非空  用户不存在 | 1  2 | 为空  用户存在 | 15  16 |
| 密码 | 非空 | 3 | 为空 | 17 |
| 确认密码 | 与密码一致  非空 | 4  5 | 与密码不一致  为空 | 18  19 |
| 邮箱 | 非空  包含@符号  @后满足url格式 | 6  7  8 | 为空  不包含@  @后不满足url格式 | 20  21  22 |
| 电话 | 输入纯数字  长度为11  第一位为1  第二位为[3~8] | 9  10  11  12 | 输入特殊字符  长度不为11  长度为空  第一位不为1  第二位不在3~8之间 | 23  24  25  26  27 |
| 角色 | 管理  员工 | 13  14 | 未选择 | 28 |

测试用例

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 编号 | 用户名 | 密码 | 确认密码 | 邮箱 | 电话 | 角色 | 等价类覆盖 | 预期输出 |
| 1 | Sunqjao2 | Sunqjao2 | Sunqjao2 | a@qq.com | 18745685296 | 员工 | 1,2,3,4,5,6,7,  8,9,10,11,12,13 | 添加成功 |
| 2 | Boss2 | Boss2 | Boss2 | b@163.com | 14278945678 | 管理 | 1,2,3,4,5,6,7,  8,9,10,11,12,13 | 添加成功 |
| 3 |  | Sunqjao2 | Sunqjao2 | a@qq.com | 18745685296 | 员工 | 15,2,3,4,5,6,7,  8,9,10,11,12,13 | 用户明不为空 |
| 4 | sunqjao | Sunqjao2 | Sunqjao2 | a@qq.com | 18745685296 | 员工 | 1,16,3,4,5,6,7,  8,9,10,11,12,13 | 用户存在 |
| 5 | Sunqjao2 |  |  | a@qq.com | 18745685296 | 员工 | 1,2,17,4,19,6,7,  8,9,10,11,12,13 | 密码为空 |
| 6 | Sunqjao2 | Sunqjao2 | Su2 | a@qq.com | 18745685296 | 员工 | 1,2,3,18,5,6,7,  8,9,10,11,12,13 | 密码不一致 |
| 7 | Sunqjao2 | Sunqjao2 |  | a@qq.com | 18745685296 | 员工 | 1,2,3,18,19,6,7,  8,9,10,11,12,13 | 确认密码为空 |
| 8 | Sunqjao2 | Sunqjao2 | Sunqjao2 |  | 18745685296 | 员工 | 1,2,3,4,5,20,7,  8,9,10,11,12,13 | 邮箱为空 |
| 9 | Sunqjao2 | Sunqjao2 | Sunqjao2 | aqq.com | 18745685296 | 员工 | 1,2,3,4,5,6,21,  8,9,10,11,12,13 | 邮箱格式错误 |
| 10 | Sunqjao2 | Sunqjao2 | Sunqjao2 | a@qqcom | 18745685296 | 员工 | 1,2,3,4,5,6,7,  22,9,10,11,12,13 | 邮箱格式错误 |
| 11 | Sunqjao2 | Sunqjao2 | Sunqjao2 | a@qq.com |  | 员工 | 1,2,3,4,5,6,7,  8,23,10,11,12,13 | 手机号为空 |
| 12 | Sunqjao2 | Sunqjao2 | Sunqjao2 | a@qq.com | 1874685296 | 员工 | 1,2,3,4,5,6,7,  8,9,24,11,12,13 | 格式错误 |
| 13 | Sunqjao2 | Sunqjao2 | Sunqjao2 | a@qq.com | 18745685296  4 | 员工 | 1,2,3,4,5,6,7,  8,9,25,11,12,13 | 格式错误 |
| 14 | Sunqjao2 | Sunqjao2 | Sunqjao2 | a@qq.com | 38745685296 | 员工 | 1,2,3,4,5,6,7,  8,9,10,26,12,13 | 格式错误 |
| 15 | Sunqjao2 | Sunqjao2 | Sunqjao2 | a@qq.com | 10745685296 | 员工 | 1,2,3,4,5,6,7,  8,9,10,11,27,13 | 格式错误 |
| 16 | Sunqjao2 | Sunqjao2 | Sunqjao2 | a@qq.com | 18745685296 |  | 1,2,3,4,5,6,7,  8,9,10,11,12,28 | 请选择角色 |

1. 项目测试说明

4.1测试项目名称及测试内容

测试1:

名称:系统登陆测试

内容:账号输入，密码输入，验证码输入，信息检验，人员类型鉴别

测试2：

名称: 新增员工信息

内容:员工名输入，电话号输入，部门输入，信息检验，

测试3：

名称: 修改员工信息

内容: 员工名输入，电话号输入，部门输入，信息检验，

测试4：

名称: 新增员工工资信息

内容:员工输入，工资输入，信息检验

测试5：

名称: 新增管理员信息

内容:

4.2测试用例

测试1登录测试

测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 账号 | 密码 | 验证码 | 预期输出 | 测试结果 |
| 1 | boss | boss | 与图片一致 | 跳到首页 |  |
| 2 | sunqjao | sunqjao | 与图片一致 | 跳到首页 |  |
| 3 | sunjao | sunqjao | 与图片一致 | 账号或密码错误 |  |
| 4 |  | sunqjao | 与图片一致 | 账号能为空 |  |
| 5 | sunqjao | sunao | 与图片一致 | 账号或密码错误 |  |
| 6 | sunqjao |  | 与图片一致 | 密码不能为空 |  |
| 7 | sunqjao | sunqjao | 与图片不一致 | 验证码错误 |  |
| 8 | sunqjao | sunqjao |  | 验证码不嫩为空 |  |

测试截图

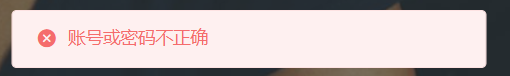
1



2



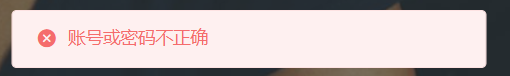
3



4



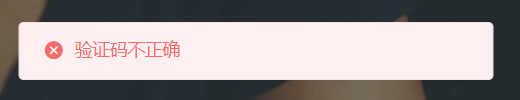
5



6



7



8



测试2 添加员工

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 名字 | 电话 | 部门 | 预期输出 | 测试结果 |
| 1 | 张三 | 18745677889 | 研发 | 添加成功 |  |
| 2 | Aaaaaaaaaaaaaaaaaaaaa  aaaaaaaaaaaaaaaaaaaaa | 18745677889 | 研发 | 名字过长 |  |
| 3 |  | 18745677889 | 研发 | 名字为空 |  |
| 4 | 张三 | 1874567788a | 研发 | 电话格式非法 |  |
| 5 | 张三 | 187456778899 | 研发 | 电话格式非法 |  |
| 6 | 张三 |  | 研发 | 电话不能为空 |  |
| 7 | 张三 | 28745677889 | 研发 | 电话格式错误 |  |
| 8 | 张三 | 10745677889 | 研发 | 电话格式错误 |  |
| 9 | 张三 | 18745677889 |  | 部门不能为空 |  |
| 10 | 张三 | 18745677889 | 研发 | 部门不存在 |  |

测试3 删除用户

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 操作员 | 删除的用户 | 预期输出 | 测试结果 |
| 1 | boss | 张三 | 添加成功 |  |
| 2 | sunqjao | 张三 | 权限不足 |  |
| 3 | boss | 张四 | 用户不存在 |  |

测试4 新增薪资

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 薪资 | 员工 | 预期输出 | 测试结果 |
| 1 | 1000 | 1 | 添加成功 |  |
| 2 | 1000e | 1 | 薪资非法 |  |
| 3 |  | 1 | 薪资不能为空 |  |
| 4 | -1 | 1 | 薪资非法 |  |
| 5 | 200000 | 1 | 工资非法 |  |
| 6 | 1000 | a | 员工不存在 |  |
| 7 | 1000 |  | 员工不能为空 |  |
| 8 | 1000 | 11111 | 员工不存在 |  |

测试5 管理员录入

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 编号 | 用户名 | 密码 | 确认密码 | 邮箱 | 电话 | 角色 | 预期输出 | 测试结果 |
| 1 | Sunqjao2 | Sunqjao2 | Sunqjao2 | a@qq.com | 18745685296 | 员工 | 添加成功 |  |
| 2 | Boss2 | Boss2 | Boss2 | b@163.com | 14278945678 | 管理 | 添加成功 |  |
| 3 |  | Sunqjao2 | Sunqjao2 | a@qq.com | 18745685296 | 员工 | 用户明不为空 |  |
| 4 | sunqjao | Sunqjao2 | Sunqjao2 | a@qq.com | 18745685296 | 员工 | 用户存在 |  |
| 5 | Sunqjao2 |  |  | a@qq.com | 18745685296 | 员工 | 密码为空 |  |
| 6 | Sunqjao2 | Sunqjao2 | Su2 | a@qq.com | 18745685296 | 员工 | 密码不一致 |  |
| 7 | Sunqjao2 | Sunqjao2 |  | a@qq.com | 18745685296 | 员工 | 确认密码为空 |  |
| 8 | Sunqjao2 | Sunqjao2 | Sunqjao2 |  | 18745685296 | 员工 | 邮箱为空 |  |
| 9 | Sunqjao2 | Sunqjao2 | Sunqjao2 | aqq.com | 18745685296 | 员工 | 邮箱格式错误 |  |
| 10 | Sunqjao2 | Sunqjao2 | Sunqjao2 | a@qqcom | 18745685296 | 员工 | 邮箱格式错误 |  |
| 11 | Sunqjao2 | Sunqjao2 | Sunqjao2 | a@qq.com |  | 员工 | 手机号为空 |  |
| 12 | Sunqjao2 | Sunqjao2 | Sunqjao2 | a@qq.com | 1874685296 | 员工 | 格式错误 |  |
| 13 | Sunqjao2 | Sunqjao2 | Sunqjao2 | a@qq.com | 18745685296  4 | 员工 | 格式错误 |  |
| 14 | Sunqjao2 | Sunqjao2 | Sunqjao2 | a@qq.com | 38745685296 | 员工 | 格式错误 |  |
| 15 | Sunqjao2 | Sunqjao2 | Sunqjao2 | a@qq.com | 10745685296 | 员工 | 格式错误 |  |
| 16 | Sunqjao2 | Sunqjao2 | Sunqjao2 | a@qq.com | 18745685296 |  | 请选择角色 |  |

五、评价

5.1范围

此测试可将系统各功能模块进行逐一的检测,对说明书中列举的功能进行排查，对系统实现各功能的正常运行做充分的测试，输入合理及不合理的测试数据检验功能的运行及出错处理情况。但由于功能模块比较多，采用功能测试设计的测试用例和对比较多，测试需花费一定的时问。

5.2 结论及BUG列表

根据我了解的情况，针对测试人事管理系统的软件测试结论如下：

1. 功能测试：在功能测试方面，测试人员应该对系统的各项功能进行全面的测试，包括但不限于创建和管理用户账户、发布任务、修改任务状态、生成报告等等。如果在测试过程中发现任何问题，应该及时反馈给开发人员进行修复。

2. 性能测试：测试人员应该对系统的响应时间、负载能力、并发能力等方面进行测试。如果系统在高负载或高并发情况下出现了性能问题，开发人员应该及时调整系统的配置和优化代码，以保证系统的正常运行。

3. 安全测试：测试人员应该对系统的安全性进行全面的测试，例如用户认证、数据加密、防止SQL注入和跨站脚本等常见安全漏洞。如果测试人员发现了任何安全问题，应该及时通知开发人员进行修复。

4. 兼容性测试：测试人员应该确认系统能够在各种不同的操作系统、浏览器和设备上正常运行。如果测试人员发现了任何兼容性问题，开发人员应该尽快深入研究原因并进行修复。

综上所述，软件测试人员应该从多个角度对测试人事管理系统进行测试，并及时发现并修复任何问题，以确保系统能够正常运行，并满足用户需求。

无BUG

5.3 修改建议

1. 在用户界面中增加一些提示，帮助用户更好地使用系统，例如增加一个帮助中心的页面，或者在每个页面的右上角增加问号图标链接到相关帮助文档。

2. 对于数据闪退、修改失败等问题，需要更好地控制用户的输入数据。因此，可以增加一些校验并提供相应的提示信息，比如让用户输入的密码符合长度要求、不允许重复的任务名称等。

3. 对于系统的安全性问题，建议对一些重要操作增加二次验证或审核，例如增加密码验证和短信验证，这将有助于减少恶意攻击的风险。

4. 对于系统的负载能力问题，可以调整数据库连接池大小、增大服务器的内存等，提高系统的性能。

5. 在进行修改时，应遵循开发规范和架构，以便于维护和扩展，同时应进行一些单元测试和集成测试，确保修改不会影响系统的其他模块功能。

总之，经过测试，对软件测试中发现的问题和潜在风险进行修改和优化，以便于提高软件质量，保证系统稳定性和用户能够快捷方便地使用。