**黑龙江大学**

**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年6月12日 7-8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** | 四号楼-520 | | | | | |
| **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**

## 程序一：前一日函数

### 1.1编写目的

此测试分析报告是为了对自编写的前一日函数PreDate程序的测试报告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 程序内容

自选语言编写程序：前一日函数PreDate。实现功能为：输入1000 年到2020 年之间的某个日期（20180412），函数返回这一天的前一天的日期。（考虑无效输入:首先判断输入日期是否合法.）

闰年满足条件(year%4==0 && year%100!=0)||(year%400==0)

### 1.3 运行环境

操作系统:Windows10

软件：IDEA

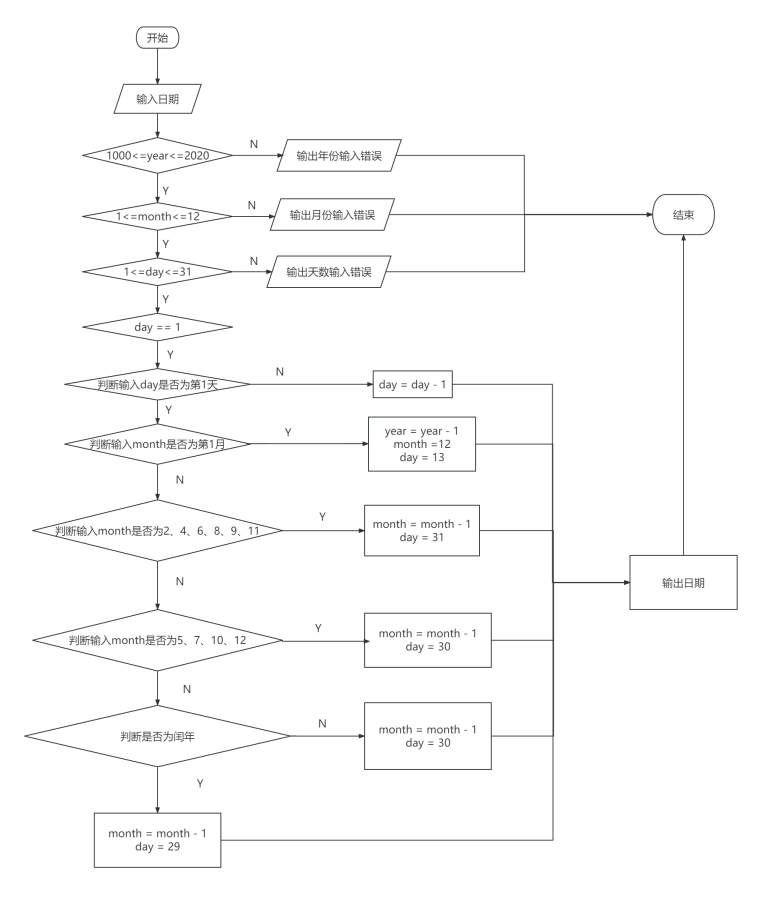
### 1.4 条件与限制

本次测试采用的主要是白盒测试，对系统各个功能模块进行测试。

### 1.5 基本路径测试方法

#### 1.5.1 程序流程图

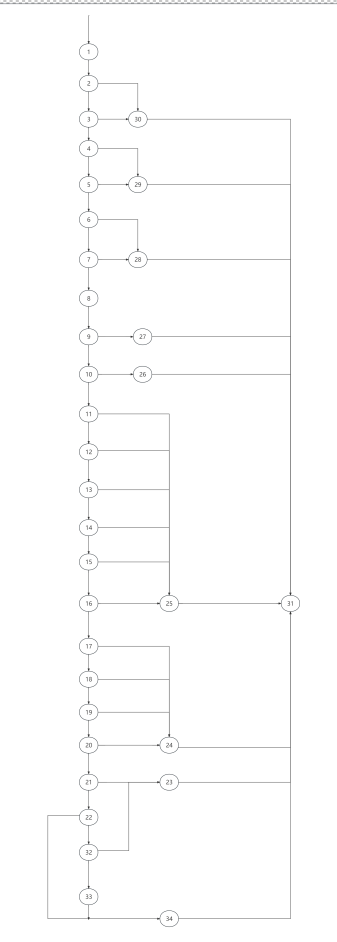
### 函数PreDate程序流程图如图1.1所示:



**图1.1 函数PreDate流程图**

#### 1.5.2 控制流图

函数PreDate控制流图如图1.2所示:



**图1.2 函数PreDate控制流图图**

#### 1.5.3 确定环形复杂性度量

因为V(G)=E-N+2，E是流图中边的数量，N是流图中结点的数量，该流图中边的数量为51，结点的数量为31，所以圈复杂度为22

#### 1.5.4 确定基本路径集合

因为圈复杂度为22，所以有22条独立路径

路径1：1-2-30-31

路径2：1-2-3-30-31

路径3：1-2-3-4-29-31

路径4：1-2-3-4-5-29-31

路径5：1-2-3-4-5-6-28-31

路径6：1-2-3-4-5-6-7-28-31

路径7：1-2-3-4-5-6-7-8-9-27-31

路径8：1-2-3-4-5-6-7-8-9-10-26-31

路径9：1-2-3-4-5-6-7-8-9-10-11-25-31

路径10：1-2-3-4-5-6-7-8-9-10-11-12-25-31

路径11：1-2-3-4-5-6-7-8-9-10-11-12-13-25-31

路径12：1-2-3-4-5-6-7-8-9-10-11-12-13-14-25-31

路径13：1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-25-31

路径14：1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-25-31

路径15：1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-24-31

路径16：1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-24-31

路径17：1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-24-31

路径18：1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-24-31

路径19：1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-23-31

路径20：

1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-34-31

路径21：

1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-32-23-31

路径22：

1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-32-33-34-31

#### 1.5.5测试案例

测试结果和预期结果如图表1-3所示

表1-3 测试案例

|  |  |  |  |
| --- | --- | --- | --- |
| 路径 | 测试用例 | 预期结果 | 实际结果 |
| 1 | 09990101 | "超出1000年到2020年范围" | "超出1000年到2020年范围" |
| 2 | 20210101 | "超出1000年到2020年范围" | "超出1000年到2020年范围" |
| 3 | 20000001 | "月份不合法" | "月份不合法" |
| 4 | 20001301 | "月份不合法" | "月份不合法" |
| 5 | 20001000 | "天数不合法" | "天数不合法" |
| 6 | 20001032 | "天数不合法" | "天数不合法" |
| 7 | 20000222 | 20000221 | 20000221 |
| 8 | 20000101 | 19991231 | 19991231 |
| 9 | 20000201 | 20000131 | 20000131 |
| 10 | 20000401 | 20000331 | 20000331 |
| 11 | 20000601 | 20000531 | 20000531 |
| 12 | 20000801 | 20000731 | 20000731 |
| 13 | 20000901 | 20200831 | 20200831 |
| 14 | 20001101 | 20001031 | 20001031 |
| 15 | 20000501 | 20000430 | 20000430 |
| 16 | 20000701 | 20000630 | 20000630 |
| 17 | 20001001 | 20000930 | 20000930 |
| 18 | 20001201 | 20001130 | 20001130 |
| 19 | 10010301 | 10010228 | 10010228 |
| 20 | 10040301 | 10040229 | 10040229 |
| 21 | 11000301 | 11000228 | 11000228 |
| 22 | 12000301 | 12000229 | 12000229 |

### 1.6 测试结论

在对实验一中的PreDate类进行测试时，我们设计了多个测试用例来覆盖不同的情况。经过测试，所有的测试用例都通过了，实现了预期的功能。

PreDate类实现了处理日期的功能。我们测试了处理不同日期的情况，包括正常日期、月初日期和闰年的情况。在所有的测试用例中，PreDate类都正确地处理了日期，并给出了预期的结果。

综上所述，实验一中的PreDate类经过了充分的测试，符合预期的功能要求。测试的结果表明这个类的实现是正确且可靠的。然而，为了更全面地测试和验证代码的正确性，可以进一步设计更多的边界测试用例和异常情况的测试用例。

**1.7 代码**

package 实验一;  
  
import java.util.Scanner;  
  
public class Test {  
 public static void main(String[] args) {  
 while(true) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("请输入日期(格式为：yyyyMMdd)：");  
 String inputDate = scanner.nextLine();  
 String preDate = *getPreDate*(inputDate);  
 System.*out*.println("前一天日期为：" + preDate);  
 }  
 }  
  
  
 // 判断输入日期是否合法  
 private static boolean isValidDate(int year, int month, int day) {  
 if (year >= 1000 && year <= 2020) {  
 if(month >= 1 && month <= 12) {  
 if(day >= 1 && day <= 31) {  
 return true;  
 }  
 }  
 }  
 return false;  
 }  
  
 // 判断是否为闰年  
 public static boolean leapYear(int year) {  
 if((year % 4 == 0 && year%100 != 0) || (year%400 == 0)) {  
 return true;  
 }  
 return false;  
 }  
  
 // 获取前一天  
 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, 8));  
 if(*isValidDate*(year, month, day)) {  
 boolean isLeapYear = *leapYear*(year);  
 if(day == 1) {  
 if(month == 1) {  
 // 2020/01/01  
 year = year - 1;  
 month = 12;  
 day = 31;  
 }else if(month == 2 || month == 4 || month == 6 || month == 8 || month == 9 || month == 11) {  
 // 2020/08/01  
 month = month - 1;  
 day = 31;  
 }else if(month == 5 || month == 7 || month == 10 || month == 12) {  
 // 2020/07/01  
 month = month - 1;  
 day = 30;  
 }else {  
 // 如果是闰年  
 if(isLeapYear) {  
 month = month - 1;  
 day = 29;  
 }else {  
 // 不是闰年  
 month = month - 1;  
 day = 28;  
 }  
 }  
 }else {  
 day = day - 1;  
 }  
 return String.*format*("%04d%02d%02d", year, month, day);  
 }else {  
 System.*out*.println("日期不合法");  
 return "";  
 }  
 }  
}

**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 | | | | | |
| **实验时间**  **（日期及节次）** | 2022.06.06 (第15周3、4节) | | | | | |
| **专业** | 软件工程 | | **学生所在学院** | | 软件学院 | |
| **年级** | 2021级 | | **学号** | | 20194387 | |
| **姓名** | 刘鑫 | | **指导教师** | | 于海洋 | |
| **实验室名称** | 4号楼134 | | | | | |
| **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**

## 程序二：合法电话号码

### 1.1编写目的

此测试分析报告是为了对合法电话号码程序的测试报告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 程序内容

自选语言编写程序：输入电话号码，对电话号码的合法性进行检查,输出是否合法,不合法在屏幕上输出错误原因。

电话号码合法要求:

电话号码由三部分构成

地区码:空白或三位数字

前缀码:非"0"或”1”开头的三位数字

后缀码:四位数字

符合以上要求的为合法电话号码

### 1.3 运行环境

操作系统:Windows10

软件：IDEA

### 1.4 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。

### 1.5 划分等价类

等价表如表1-1所示:

**表1-1 等价表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输入等价类** | **有效等价类** |  | **无效等价类** |  |
| 输入的字符 | 全是数字 | 1 | 7位前缀码含非法字符  7位后缀码含非法字符  10位地区码含非法字符  10位前缀码含非法字符  10位后缀码含非法字符 | 6  7  8  9  10 |
| 输入字符位数 | 输入7位  输入10位 | 2  3 | 非7位数字或非10位数字 | 11 |
| 前缀码开头数字 | 非1开头  非0开头 | 4  5 | 7位数字时开头为1  7位数字时开头为0  10位数字时开头为1  10位数字时开头为0 | 12  13  14  15 |

### 1.6 测试用例

测试用例如表1-2所示:

**表1-2 测试用例**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **测试用例**  **编号** |  | **输入数据** |  | **预期**  **输出** | **覆盖等价类** |
| **电话号码** | | |
| 1 | 2345678 | | | 正确 | 1 2 4 5 |
| 2 | 0123456789 | | | 正确 | 1 3 4 5 |
| 3 | Ab12345 | | | 前缀码含非法字符 | 6 |
| 4 | 789abcd7 | | | 后缀码含非法字符 | 7 |
| 5 | Ab47894568 | | | 地区码含非法字符 | 8 |
| 6 | 123a567890 | | | 前缀码含非法字符 | 9 |
| 7 | 123456789b | | | 后缀码含非法字符 | 10 |
| 8 | 555 | | | 位数不合法 | 11 |
| 9 | 1234567 | | | 前缀码不合法 | 12 |
| 10 | 0234567 | | | 前缀码不合法 | 13 |
| 11 | 1231234567 | | | 前缀码不合法 | 14 |
| 12 | 1230234567 | | | 前缀码不合法 | 15 |

### 1.7 实施测试

测试用例如表1-3所示:

**表1-3 测试用例**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例**  **编号** |  | **输入数据** |  | **实际**  **输出** |
| **电话号码** | | |
| 1 | 2345678 | | | 正确 |
| 2 | 0123456789 | | | 正确 |
| 3 | Ab12345 | | | 前缀码含非法字符 |
| 4 | 789abcd7 | | | 后缀码含非法字符 |
| 5 | Ab47894568 | | | 地区码含非法字符 |
| 6 | 123a567890 | | | 前缀码含非法字符 |
| 7 | 123456789b | | | 后缀码含非法字符 |
| 9 | 555 | | | 位数不合法 |
| 10 | 1234567 | | | 前缀码不合法 |
| 11 | 0234567 | | | 前缀码不合法 |
| 12 | 1231234567 | | | 前缀码不合法 |

### 1.8 测试结论

通过本次对电话号码程序二的黑盒测试,对黑盒测试的测试方法、流程有了更加深入的认识。预期结果与程序执行结果一致，说明测试成功，程序无异常。

### 1.9 代码

package 实验二;  
  
import java.util.Scanner;  
  
public class PhoneValidator {  
  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("请输入电话号码：");  
 String phoneNumber = scanner.nextLine();  
 String errorMessage = *validatePhoneNumber*(phoneNumber);  
 if (errorMessage != null) {  
 System.*out*.println("电话号码不合法：" + errorMessage);  
 } else {  
 System.*out*.println("电话号码合法");  
 }  
 }  
  
 public static String validatePhoneNumber(String phoneNumber) {  
 if (phoneNumber == null || phoneNumber.isEmpty()) {  
 return "电话号码不能为空";  
 }  
  
 String[] parts = phoneNumber.split("\\s+");  
  
 if (parts.length != 3) {  
 return "电话号码必须由三部分构成";  
 }  
  
 String areaCode = parts[0];  
 String prefix = parts[1];  
 String suffix = parts[2];  
  
 if (areaCode.isEmpty() || areaCode.length() != 3) {  
 return "地区码必须为空白或三位数字";  
 }  
  
 if (prefix.isEmpty() || prefix.length() != 3 || prefix.startsWith("0") || prefix.startsWith("1")) {  
 return "前缀码必须为非\"0\"或\"1\"开头的三位数字";  
 }  
  
 if (suffix.isEmpty() || suffix.length() != 4 || !suffix.matches("\\d+")) {  
 return "后缀码必须为四位数字";  
 }  
  
 return null;  
 }  
  
}

**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 | | | | | |
| **实验时间**  **（日期及节次）** | 2023.06.13 (第16周3、4节) | | | | | |
| **专业** | 软件工程 | | **学生所在学院** | | 软件学院 | |
| **年级** | 2021级 | | **学号** | | 20215782 | |
| **姓名** | 欧乐 | | **指导教师** | | 于海洋 | |
| **实验室名称** | 4号楼138 | | | | | |
| **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**

### 1.1编写目的

此测试分析报告是为了对人民币数字到大写转换系统的设计、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 程序内容

编写“人民币数字到大写转换”程序。

计算范围：整数部分长度在12位（千亿单位）及以内， 小数部分长度在2位（分单位）

具体需求如下：

1）中文大写金额数字应用壹、贰、叁、肆、伍、陆、柒、捌、玖、拾、佰、仟、万、亿、元、角、分、零、整(正)等字样。

2）中文大写金额数字到"元"为止的，在"元"之后，应写"整"(或"正")字，在"角"之后，可以不写"整"(或"正")字。

3）中文大写金额数字前应标明"人民币"字样，大写金额数字有"分"的，“分"后面不写"整”(或"正")字。

4）大写金额数字应紧接"人民币"字样填写，不得留有空白。

5）阿拉伯数字小写金额数字中有"0"时，中文大写应按照汉语语言规律、金额数字构成和防止涂改的要求进行书写。

举例如下：

1、阿拉伯数字中间有"0"时，中文大写要写"零"字，如￥1409.50，应写成：人民币壹仟肆佰零玖元伍角。

2、阿拉伯数字中间连续有几个"0"时，中文大写金额中间只写一个"零"字，如￥6007.14，应写成：人民币陆仟零柒元壹角肆分。

3、阿拉伯金额数字万位和元位是"0"，或者数字中间连续有几个"0"，万位、元位也是"0"，但千位、角位不是"0"时，中文大写金额中只写一个零字，也可以不写"零"字。如￥1680.32，应写成：人民币壹仟陆佰捌拾元叁角贰分，又如￥107000.53，应写成：人民币壹拾万零柒仟元伍角叁分。

4、阿拉伯金额数字角位是"0"，而分位不是"0"时，中文大写金额"元"后面应写"零"字。如￥16409.02，应写成人民币：壹万陆仟肆佰零玖元零贰分；又如￥325.04，应写成人民币叁佰贰拾伍元零肆分。

### 1.3 运行环境

操作系统:Windows10

软件：CodeBlocks

### 1.4 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。

### 1.5 划分等价类

等价表如表1-1所示:

**表1-1:等价表**

|  |  |  |
| --- | --- | --- |
| **输入等价类** | **有效等价类** | **无效等价类** |
| **长度** | 1. 12>=整数长度>0 2. 2>=小数长度>=0 | 1. 整数长度>12 2. 小数长度>2 3. 什么都不输入 4. 整数长度=0 |
| **输入的字符** | 1. 输入有效数字 2. 小数点个数<=1 3. 1<=输入中间连续的0的个数<=10 4. 输入0的位置在整数部分 5. 输入0的位置在小数部分 6. 整数部分为0 7. 小数部分为0 8. 整数部分和小数部分都为0 | 1. 含有非法字符 2. 输入负值 3. 小数点个数>1 |
| **输出的字符** | 1. 壹 2. 贰 3. 叁 4. 肆 5. 伍 6. 陆 7. 柒 8. 捌 9. 玖 10. 零 11. 整 12. 角 13. 元 14. 分 15. 拾 16. 佰 17. 仟 18. 万 19. 亿 20. 元后写整 21. 分后不写整 22. 角后不写整 23. 输出紧接人们币字样 |  |

### 1.6 测试用例

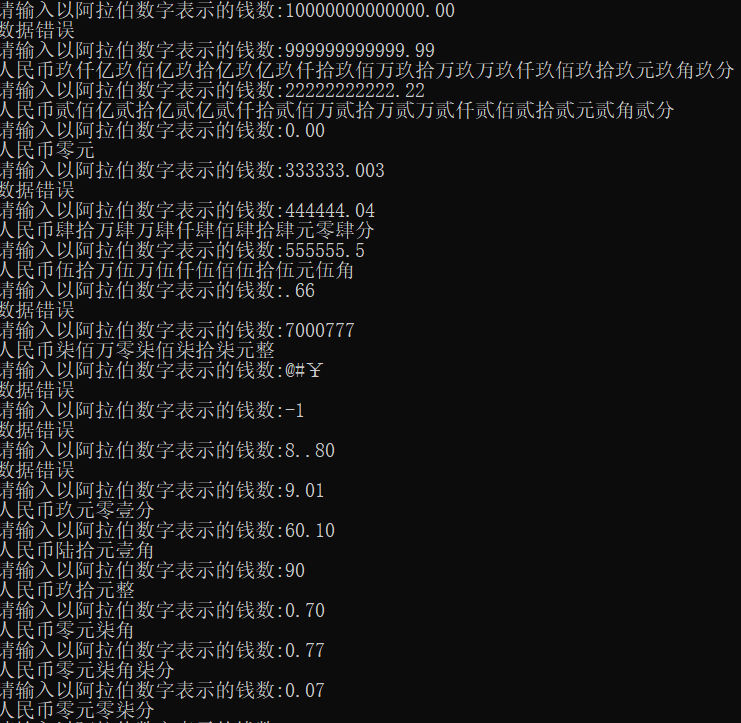
测试用例如表1-2所示:

**表1-2:测试用例**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | **输入数据** | **期望结果** | **运行结果** | **覆盖等价类** |
| 1 | 10000000000000.00 | 提示错误 | 数据错误 | 3 |
| 2 | 12345678909.99 | 人民币壹佰贰拾叁亿肆仟伍佰陆拾柒万捌仟玖佰零玖元玖角玖分 | 人民币壹佰亿贰拾亿叁亿肆仟拾伍佰万陆拾万柒万捌仟玖佰零玖元玖角玖分 | 1、2、7、8、38、40 |
| 3 | 22222222222.22 | 人民币壹佰贰拾叁亿肆仟伍佰陆拾玖万陆仟叁佰贰拾壹元伍角肆分 | 人民币贰佰亿贰拾亿贰亿贰仟拾贰佰万贰拾万贰万贰仟贰佰贰拾贰元贰角贰分 | 1、2、7、8、38 |
| 4 | 0.00 | 人民币零元整 | 人民币零元 | 1、2、7、8、14、37 |
| 5 | 333333.003 | 提示错误 | 数据错误 | 4 |
| 6 | 444444.04 | 人民币肆拾万肆万肆仟肆佰肆拾肆元零肆分 | 人民币肆拾万肆万肆仟肆佰肆拾肆元零肆分 | 1、2、7、8、11、38 |
| 7 | 555555.5 | 人民币伍拾万伍万伍仟伍佰伍拾伍元伍角 | 人民币伍拾万伍万伍仟伍佰伍拾伍元伍角 | 1、2、7、8、39 |
| 9 | .66 | 提示错误 | 数据错误 | 6 |
| 10 | 101 | 人民币壹佰零壹元整 | 人民币壹佰零壹元整 | 1、7、9、37 |
| 11 | 1001 | 人民币壹仟零壹元整 | 人民币壹仟零壹元整 | 1、7、9、37 |
| 12 | 1000001 | 人民币壹佰万零壹元整 | 人民币壹佰万零壹元整 | 1、7、9、37 |
| 13 | 100000000001 | 人民币壹仟亿零壹元整 | 人民币壹仟亿零壹元整 | 1、7、9、37 |
| 15 | 10000000001 | 人民币壹佰亿零壹元整 | 人民币壹佰亿零壹元整 | 1、7、9、37 |
| 16 | @#￥ | 提示错误 | 数据错误 | 15 |
| 17 | -1 | 提示错误 | 数据错误 | 16 |
| 18 | 8..80 | 提示错误 | 数据错误 | 17 |
| 19 | 90 | 人民币贰拾元整 | 人民币玖拾元整 | 1、7、10 |
| 20 | 0.70 | 人民币叁角 | 人民币零元柒角 | 1、2、7、8、10、11、39 |
| 21 | 0.77 | 人民币叁角叁分 | 人民币零元柒角柒分 | 1、2、7、8、10、12 |
| 22 | 0.07 | 人民币叁分 | 人民币零元零柒分 | 1、2、7、8、10、12 |
| 23 | 0 | 人民币零元整 | 数据错误 | 16 |
| 24 | “空” | 数据错误 | “空” | 5 |

### 1.7 测试结果截图

测试结果如图1-1所示



**图1-1测试结果**

### 1.8 制约条件

根据程序的具体要求,将程序的输入(原因)列如下几条:

1.输入的整数长度最大为12

2.没有小数部分

3.输入至小数点后一位

4.输入至小数点后两位

5.输入的数字中间有零

6.非法输入

7.输入的数是最大数

E关系:①2,3,7

②2,3,4

③6与其他

R关系:4.7

### 1.9 测试结论

通过对程序三进行的黑盒边界值法进行测试,发现代码中间的转换有一定的问题，测试用例并没有全部通过。

### 1.10 代码

#include<stdio.h>

#include<stdlib.h>

int change()

{

char m[20] = { '0' };

printf("请输入以阿拉伯数字表示的钱数:");

scanf("%s", &m);

int d = 0, sum = 0, length = 0, length2 = 0;

int num = 0;

for (int i = 0; m[i] != '\0'; i++)

{

if (m[i] == '.')

{

d = i;

num++;

}

if (m[i] >= '0'&&m[i] <= '9' || m[i] == '.')

length2++;

length = i;

}

if (m[0] == '.' || num > 1|| length + 1 != length2)

{

printf("数据错误\n");

return 0;

}

if (d != 0 && length == d + 1)

{

m[d + 2] = '0';

}

if (d == 0)

d = length + 1;

if (length > d + 2 || d > 12)

{

printf("数据错误\n");

return 0;

}

if (m[0] == '0'&&m[1] != '.')

{

printf("数据错误\n");

return 0;

}

printf("人民币");

for (int i = 0; m[i] != '\0'; i++)

{

switch (m[i])

{

case '0':

{

sum = 0;

for (int j = i; j < d; j++)

{

if (m[j] == '0')

{

sum++;

}

}

if (sum != d - i && i != d + 2 && m[i + 1] != '0' || i == 0 )

{

printf("零");

}

break;

}

case '1':

printf("壹");

break;

case '2':

printf("贰");

break;

case '3':

printf("叁");

break;

case '4':

printf("肆");

break;

case '5':

printf("伍");

break;

case '6':

printf("陆");

break;

case '7':

printf("柒");

break;

case '8':

printf("捌");

break;

case '9':

printf("玖");

break;

}

if (i == d - 1 && d == length + 1)

printf("元整");

if (i == d - 1 && d != length + 1)

printf("元");

if (m[i] != '0')

{

switch (d - i)

{

case -1:

printf("角");

break;

case -2:

printf("分");

break;

case 2:

printf("拾");

break;

case 3:

printf("佰");

break;

case 4:

printf("仟");

break;

case 5:

printf("万");

break;

case 6:

printf("拾万");

break;

case 7:

printf("佰万");

break;

case 8:

printf("仟拾");

break;

case 9:

printf("亿");

break;

case 10:

printf("拾亿");

break;

case 11:

printf("佰亿");

break;

case 12:

printf("仟亿");

break;

}

}

}

printf("\n");

return 0;

}

int main()

{

while(1)

{

change();

}

}

**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）** | 2023.06.13 星期二3-4节 | | | | | |
| **专业** | 软件工程 | | **学生所在学院** | | 软件学院 | |
| **年级** | 2021级 | | **学号** | | 20194387 | |
| **姓名** | 刘鑫 | | **指导教师** | | 于海洋 | |
| **实验室名称** | 4号楼-[134]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 | 盐发 | 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

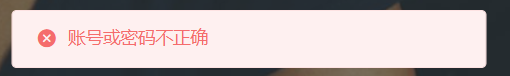
1



2



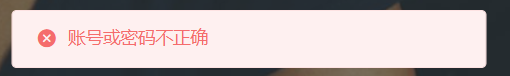
3



4



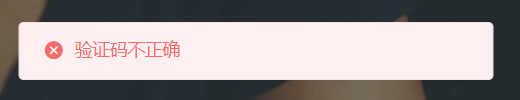
5



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 | 盐发 | 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。