武汉纺织大学

Java应用开发课程设计

**小学生算术运算测验系统**

**学 院： 数学与计算机学院**

**班 级： 软件11801**

**姓 名： 黄璇**

**学 号： 1704240618**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2021年6月26日**

目 录

[1 需求分析 1](#_Toc2965)

[1.1用户登陆 1](#_Toc6465)

[1.2主菜单 1](#_Toc27274)

[1.3各项功能 1](#_Toc17655)

[1.3.1 出题 1](#_Toc21322)

[1.3.2 答题 1](#_Toc30258)

[1.3.3 查询成绩 2](#_Toc28610)

[1.3.4 排行榜 2](#_Toc14439)

[1.3.5 退出 3](#_Toc28851)

[1.3.6 \*\*\*\* 3](#_Toc4429)

[2 系统设计 3](#_Toc15323)

[2.1用户用例图 3](#_Toc21899)

[2.2 UML活动图（Activity Diagram） 6](#_Toc5309)

[2.2.1 出题 6](#_Toc12858)

[2.2.2 答题 7](#_Toc2967)

[2.2.3 查询成绩 7](#_Toc2967)

[2.2.4 排行榜 7](#_Toc2967)

[2.2.5 退出 8](#_Toc16882)

[2.2.6 \*\*\*\*模块 8](#_Toc4032)

[3 系统实现 9](#_Toc27433)

[3.1 项目结构 9](#_Toc1061)

[3.2 用户类User.java 9](#_Toc2679)

[3.3 运算表达式类 Expression.java 10](#_Toc26179)

[3.4 随机生成类Genarate.java 11](#_Toc32366)

[3.5 主界面类Driver.java 14](#_Toc26330)

[3.6 \*\*\*\*\*\*.java 14](#_Toc14997)

[4 系统测试 14](#_Toc19341)

[5 系统总结 14](#_Toc30615)

# 1 需求分析

开发一个简单的算术运算测验程序，完成两位数以内两个随机整数的加减乘除运算作答（注：两个数相除时取整）。实现“出题”、“答题”、“查看成绩”、“排行榜”、“退出”等功能，具体要求如下：

## 1.1用户记录

当程序运行时，首先提示“请输入测验人员的姓名：”，并记录下来。

## 1.2主菜单

输入完毕后，显示以下主菜单：

简单算术运算测验系统

1.出题

2.答题

3.查看成绩

4.排行榜

5.退出

请选择（1-5）：

如果输入1-5以外的选择，则显示提示信息“输入错误！只能输入1-5，请重新输入：”后，返回主界面。

## 1.3各项功能

### 1.3.1 出题

当选择1时，提示“请输入题目数量(输入0退出): ”，用户输入题目数量后，如果输入为0，则提示“退出成功！” 并返回主菜单，否则，提示“题数数量为用户输入的数量” (假定为m)，并返回主菜单。

### 1.3.2 答题

当选择2时，如果尚未出题则，提示“请先出题！”并返回主菜单，否则，如果测验人员人数不为0（初始为0），才提示“请输入测验人员的姓名：”（假定为n）并记录（由于题目给定的是当程序运行时，首先输入测验人员的姓名，再显示主菜单，故对第一个用户不用重复提示输入姓名，但之后的用户姓名也要记录下来以供排名时所用，每次答题会进行测验人员人数加1），

然后显示随机生成的两位数整数运算的第一个表达式，并提示“请输入运算结果：”，如果结果正确则提示“答案正确！”，否则提示“答题错误！正确答案为XX”，然后显示第二个表示式，再次提示“请输入运算结果：”，直到用户答完第m道题提示对应信息后，输出“答题完毕，恭喜你获得了xx.xx分！成绩以保存至xxx”，并返回主菜单。

### 1.3.3 查询成绩

当选择3时，如果测试人员人数为0，则提示“请先答题！”并返回主菜单，否则，输出如下信息：

“本次测验情况如下：

姓名： xx

题目：xx

自己的答案：xx 正确答案：xx

题目：xx

自己的答案：xx 正确答案：xx

…

成绩：xx.xx

正确率：xx.xx%”

并返回主菜单

### 1.3.4 排行榜

当选择4时，如果测试人员人数为0，则提示“请先答题！”并返回主菜单，否则，输出如下信息：

“升序显示所有测验人员排名：

排名：1 姓名：xx 成绩：xx.xx 测验时间：xxxx-xx-xx xx:xx:xx”

排名：2 姓名：xx 成绩：xx.xx 测验时间：xxxx-xx-xx xx:xx:xx”

…

排名：n 姓名：xx 成绩：xx.xx 测验时间：xxxx-xx-xx xx:xx:xx”

并返回主菜单

### 1.3.5 退出

当选择5时，显示“您确认要退出吗（y/n）”,输入y，显示“退出成功！欢迎下次继续使用”，退出程序，输入“n”，返回主菜单。

### 1.3.6 \*\*\*\*

每个功能按编号往后递增描述

# 2 系统设计

## 2.1用户用例图

系统分为收银员，管理员两个角色，各角色的用户用例如下图2-1所示：

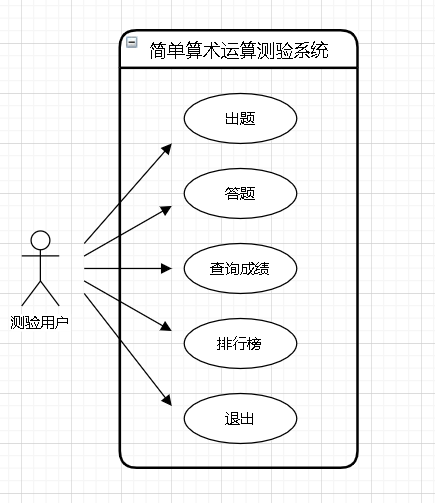


图2-1 用户用例图

## 2.2 UML活动图（Activity Diagram）

### 2.2.1 出题

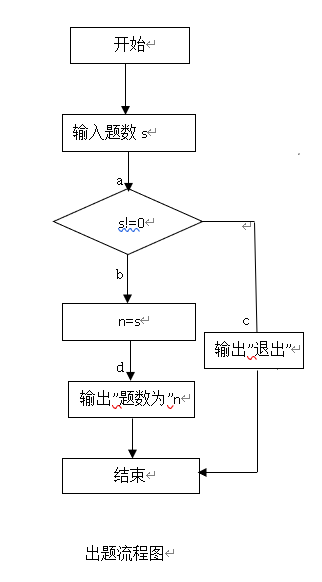


图2-2 出题流程图

### 2.2.2 答题

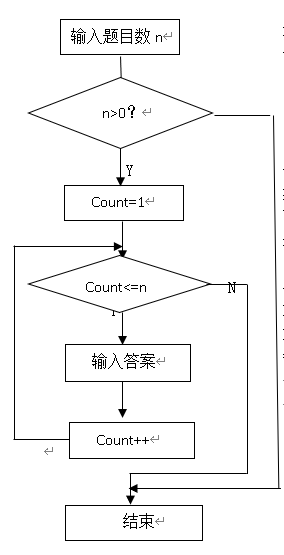
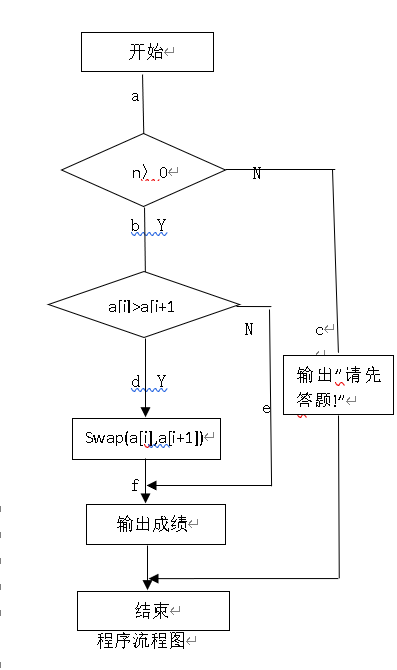


图2-5 答题功能活动图

### 2.2.3 查询成绩

只要答题了直接输出

### 2.2.4 排行榜



### 2.2.5 退出

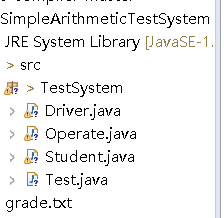
选y（yes）直接退出

### 2.2.6 \*\*\*\*模块

每个功能按编号往后递增绘图

# 3 系统实现

## 3.1 项目结构



## 3.2 用户类User.java

描述该文件作用，并对里面的关键代码进行解释

**作用：用户类用来存放用户的姓名、成绩、测验时间等信息**

**package** TestSystem;

**import** java.util.Scanner;

**public** **class** User {

**private** String name;

**private** **double** grade;

**private** String time;

// 有参构造函数

**public** User(String name, **double** grade, String time) {

**this**.name = name;

**this**.grade = grade;

**this**.time = time;

}

// getter,setter方法

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **double** getGrade() {

**return** grade;

}

**public** **void** setGrade(**double** grade) {

**this**.grade = grade;

}

**public** String getTime() {

**return** time;

}

**public** **void** setTime(String time) {

**this**.time = time;

}

// 输出用户的姓名、成绩和测验时间

**public** String toString() {

String msg;

msg = "姓名：" + **this**.name;

msg += " 成绩：" + **this**.grade;

msg += " 测验时间：" + **this**.time;

**return** msg;

}

}

## 3.3 运算表达式类Expression.java

描述该文件作用，并对里面的关键代码进行解释

**作用：运算表达式类用来实例化运算表达式对象，可通过getResult方法得到表达式的正确答案**

**package** TestSystem;

**public** **class** Expression{

**private** **int** number1; // 随机整数1

**private** **int** number2; // 随机整数2

**private** String operator; // 运算符

**private** **static** **int** *grade* = 0; // 成绩

// 有参构造函数

**public** Expression(**int** number1, **int** number2, String operator) {

**this**.number1 = number1;

**this**.number2 = number2;

**this**.operator = operator;

}

// getter,setter方法

**public** **float** getNumber1() {

**return** number1;

}

**public** **void** setNumber1(**int** number1) {

**this**.number1 = number1;

}

**public** **float** getNumber2() {

**return** number2;

}

**public** **void** setNumber2(**int** number2) {

**this**.number2 = number2;

}

// 输出随机生成的运算表达式

**public** String toString() {

**return** **this**.number1 + **this**.operator + **this**.number2;

}

// 根据不同的运算符得到相应的运算结果

**public** **int** getResult() {

**int** result = 0;

**switch** (**this**.operator) {

**case** "+":

result = **this**.number1 + **this**.number2;

**break**;

**case** "-":

result = **this**.number1 - **this**.number2;

**break**;

**case** "\*":

result = **this**.number1 \* **this**.number2;

**break**;

**case** "/":

result = **this**.number1 / **this**.number2;

**break**;

}

**return** result;

}

}

## 3.4 随机生成类Genarate.java

**作用：随机生成类用来随机生成两位数以内的两个整数和四则运算中的一种运算符**

**package** TestSystem;

**import** java.util.Random;

**import** java.util.Scanner;

**public** **class** Generate {

// 随机产生一个两位数以内的整数

**public** **static** **int** getRandom() {

Random ran = **new** Random();

**return** ran.nextInt(99);

}

// 随机产生一个操作符

**public** **static** String getOperator() {

String[] operateEle = { "+", "-", "\*", "/" };

Random ran = **new** Random();

**return** operateEle[ran.nextInt(4)];

}

}

## 3.5 主界面类Driver.java

描述该文件作用，并对里面的关键代码进行解释

**作用：主界面类用来显示主菜单，包括每个功能的实现方法**

**package** TestSystem;

**import** java.util.Scanner;

**import** java.io.BufferedReader;

**import** java.io.BufferedWriter;

**import** java.io.File;

**import** java.io.FileNotFoundException;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** java.io.Reader;

**import** java.math.BigDecimal;

**import** java.io.FileReader;

**import** java.text.DecimalFormat;

**import** java.text.NumberFormat;

**import** java.text.SimpleDateFormat;

**import** java.util.Date;

**public** **class** Driver {

**static** **int** *num* = 0; // 题目数量

**static** **int** *number* = 0; // 测验人员人数

**static** User[] *array* = **new** User[100]; // 成绩排序数组

**static** String *name*; // 测验人员姓名

**static** String *gradeList*; // 本次测验成绩列表

**public** **static** **void** main(String args[]) **throws** IOException {

Scanner scan = **new** Scanner(System.***in***);

System.***out***.println("请输入测验人员的姓名:");

*name* = scan.next();

**int** select;

**for** (;;) { // 主菜单

System.***out***.println("--------------------------------------");

System.***out***.println("简单算术运算测验系统");

System.***out***.println("1出题 \n2答题\n3查询\n4排行榜\n5退出");

System.***out***.println("请选择（1-5）：");

System.***out***.println("--------------------------------------");

select = scan.nextInt();

**switch** (select) {

**case** 5:

System.***out***.println("您确认要退出吗（y/n）：");

**char** s = scan.next().charAt(0);

**if** (s == 'y') {

System.***out***.println("退出成功！欢迎下次继续使用");

**return**;

}

**break**;

**case** 1:

*question*();

**break**;

**case** 2:

*answer*();

**break**;

**case** 3:

*query*();

**break**;

**case** 4:

*sort*();

**break**;

**default**:

System.***out***.println("输入错误！只能输入1-5，请重新输入：");

**break**;

}

}

}

**public** **static** **void** question() { // 出题

**int** n;

Scanner scan = **new** Scanner(System.***in***);

System.***out***.println("请输入题目数量(输入0退出):");

n = scan.nextInt();

**if** (n == 0) {

System.***out***.println("退出成功！");

**return**;

} **else** {

*num* = n;

}

System.***out***.println("题目数量为" + *num*);

}

**public** **static** **void** answer() **throws** IOException { // 答题

File file = **new** File("F:/JuniorFirst/软件测试/SimpleArithmeticTestSystem/grade.txt");

Scanner scan = **new** Scanner(System.***in***);

**int** i = 0; // 当前第i题

**int** correct = 0; // 做对的题数

String rate; // 正确率

**int** inputResult = 0;

**double** point = 100.00 / *num*; // 每道题的分值

**double** grade = 0.00;

**if** (*num* == 0) {

System.***out***.println("请先出题！");

**return**;

}

**if** (!file.exists()) {

file.createNewFile();

} **else** {

*gradeList* = "";

**if** (*number* != 0) {

System.***out***.println("请输入测验人员的姓名：");

*name*=scan.next();

}

*gradeList* = *gradeList* + "姓名：" + *name* + "\n";

*number* = *number* + 1;

**do** {

Expression expression = **new** Expression(Generate.*getRandom*(), Generate.*getRandom*(), Generate.*getOperator*());

System.***out***.println(expression); // 输出运算表达式

System.***out***.println("请输入运算结果：");

inputResult = scan.nextInt(); // 得到用户输入的结果

*gradeList* = *gradeList* + "题目：" + expression + "\n" + "自己的答案：" + inputResult + " 正确答案：" + expression.getResult()

+ "\n";

**if** (inputResult == expression.getResult()) { // 回答正确

correct++;

System.***out***.println("答案正确!");

grade = grade + point; // 加上一题的分值

} **else** { // 回答错误

System.***out***.println("答案错误！\n正确答案为" + expression.getResult());

}

i++;

} **while** (i < *num*);

DecimalFormat df = **new** DecimalFormat("#.00");

System.***out***.println("答题完毕，恭喜你获得了" + df.format(grade) + "分！");

// 创建一个数值格式化对象

NumberFormat numberFormat = NumberFormat.*getInstance*();

// 设置精确到小数点后2位

numberFormat.setMaximumFractionDigits(2);

rate = numberFormat.format((**float**) correct / (**float**) *num* \* 100);

*gradeList* = *gradeList* + "成绩：" + df.format(grade) + "\n" + "正确率：" + rate + "%" + "\n";

BigDecimal bd = **new** BigDecimal(grade);

**double** g = bd.setScale(2, BigDecimal.***ROUND\_HALF\_UP***).doubleValue();

SimpleDateFormat sdf = **new** SimpleDateFormat("yyyy-MM-dd hh:MM:ss");

Date today = **new** Date();

String todayStr = sdf.format(today);

User user = **new** User(*name*, g, todayStr); // 实例化用户对象

*array*[*number*] = user; // 保存用户信息到数组

FileReader fr = **new** FileReader(file);

BufferedReader br = **new** BufferedReader(fr);

String last = br.readLine();

FileWriter fw = **new** FileWriter(file);

BufferedWriter bw = **new** BufferedWriter(fw);

bw.write(last + user.toString()); // 以文件形式存放测验人的姓名，测验成绩，测验时间

System.***out***.println("成绩已保存至F:\\JuniorFirst\\软件测试\\SimpleArithmeticTestSystem\\grade.txt");

bw.close();

br.close();

}

}

**public** **static** **void** query() { // 查询

**if** (*number* == 0) {

System.***out***.println("请先答题!");

**return**;

}

System.***out***.println("本次测验情况如下:");

System.***out***.println(*gradeList*);

}

**public** **static** **void** sort() { // 排序

**if** (*number* == 0) {

System.***out***.println("请先答题!");

**return**;

}

System.***out***.println("升序显示所有测验人员排名：");

**for** (**int** i = 1; i <= *number*; i++) {

**for** (**int** j = i + 1; j <= *number*; j++) {

// 根据名次由小到大即成绩由大到小排序

**if** (*array*[i].getGrade() < *array*[j].getGrade()) {

User temp = *array*[i];

*array*[i] = *array*[j];

*array*[j] = temp;

}

}

}

**int** k = 1;

**for** (; k <= *number*; k++) {

System.***out***.println("排名：" + k + " " + *array*[k] + "");

}

}

}

## 3.9 \*\*\*\*\*\*.java

描述该文件作用，并对里面的关键代码进行解释

每个文件按编号往下递增

# 4 系统测试

设计测试用例，给出程序每个功能模块的运行结果截图。

## 4.1 出题



## 4.2 答题





## 4.3 查询成绩









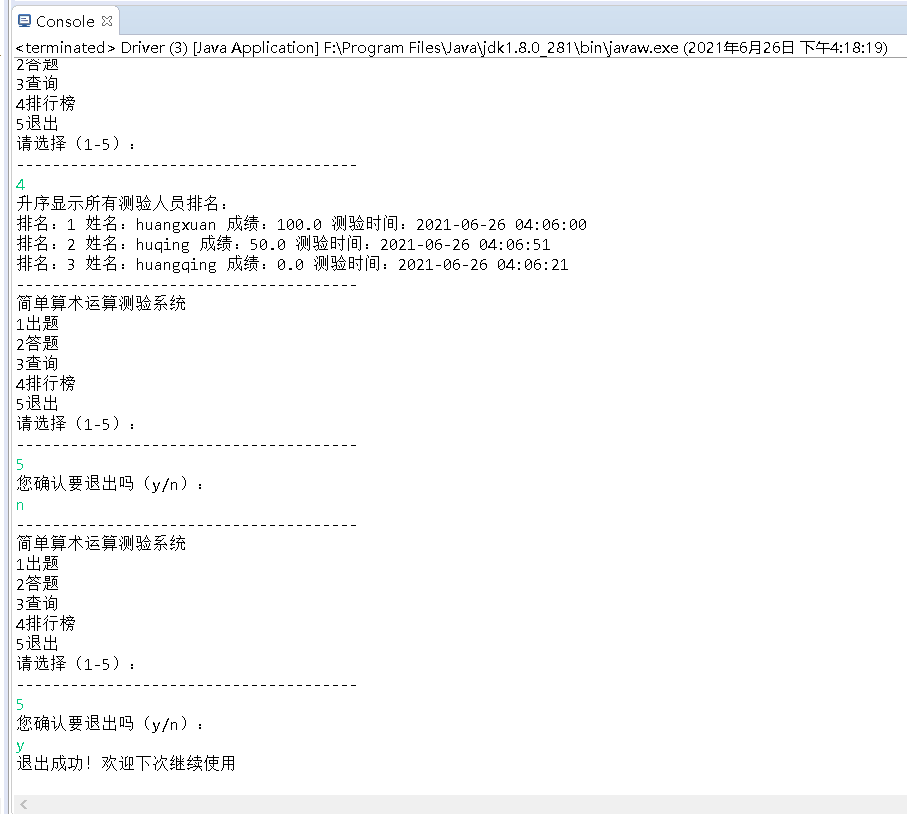
## 4.4 排行榜







## 4.5 退出



# 5 系统总结

主要问题的解决过程、系统存在的不足，本门课程的收获，课程的建议，以后的发展方向，自评成绩。

## 5.1 解决过程

按照需求分析、系统设计来进行系统实现，代码的问题就在断点输出来进行调试。

## 5.2 系统不足

在随机生成整数及运算符时，当运算符号为’/’，除数即分母为0时，会出现动态语义错误，导致程序崩溃。

## 5.3 课程收获

java的基本语法、流程控制、模块化编程、面向对象基础、类之间的组合关系、常用正则表达式、XML的四种解析方式、Dom4j主要接口（document/element等）及其相关操作，json格式数据、jdbc连接数据库的五个步骤，项目分层设计思想、面向接口编程、dao设计模式和实现步骤，MyBatis框架，网络原理：传输层tcp协议三次握手建立连接、四次挥手断开连接，无连接udp协议，各自格式和两者区别，http协议请求方法（get/post等）响应状态码 网络编程等。

## 5.4 发展方向

发展方向：目前主要是服务器开发偏后端，语言C++/java，未来想转人工智能机器学习类

## 5.5 自评成绩

60+