网页版四则运算

2016-04-10 21:39

一程序设计思想：

1.用jsp加java写一个简单地开始界面，在此界面中，利用form表单，用户输入对算式的要求，点击确认按钮后跳转到出题页面。

2.在出题界面要用request来接收form表单中的值，借鉴之前java代码（稍微修改一下，直接复制粘贴就行），编写对用户要求的实现代码和计算结果的代码，每次只出一道题，然后通过form表单把结果和用户输入的值传到判断界面。

3.在判断界面获取表单的值，调用equals()进行判断。判断结束后，点击返回按钮返回出题页面。因为form表单提交后会刷新页面，导致出题页面重新刷新，使一开始用户对题目的要求丢失，所以在从出题页面转到判断页面的时候，利用input中的hidden属性，隐藏的把用户的要求传入判断页面，同样的在把用户要求从判断页面传回出题页面。

4.添加出题上限的功能，这里我用的是全局变量来控制题目的数量，在出题页面定义一个全局变量，然后每出一道题，全局变量加1，达到上限后则不再出题。

5.添加对题目不重复的功能，要想判断题目是否重复，就需要记录之前出的所有题，这里我用文本记录出的题目，用file writer和filereader来读取文本中的数据，BufferedReader中的readLine()来取出文本中每一行的数据，注意只要没有关闭BufferedReader，则每调用一次readLine()，读取一行文本中的数据。

6.解决文本清空问题，在题目输出到文本中后，程序结束后，文本中的内容还存在，所以要在程序再次执行时清空文本内容，只要在用户输入题目要求的页面中一开始就创建一个路径和名字一样的文本就可以了。

7.添加四则混合运算功能，这里括号不能嵌套，算式只有4个数参与运算，利用java栈进行求和运算，需要考虑符号的优先级。

二.源程序代码：

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<%@page import= "java.io.\*" %>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>四则运算</title>

</head>

<body>

<%

File f = new File("read.txt");

FileWriter fw = new FileWriter(f);

fw.write("");

fw.close();

%>

<form id="form1" method="post" action="request.jsp">

运算题目数量: <input type="text" name="num"><br/>

取值范围上限: <input type="text" name="random"><br/>

是否有乘除法:<br/>

是：<input type="radio" name="select1" value="1">

否：<input type="radio" name="select1" value="0"><hr/>

除法有无余数:<br/>

是：<input type="radio" name="select2" value="1">

否：<input type="radio" name="select2" value="0"><hr/>

是否有混合运算:<br/>

是：<input type="radio" name="select3" value="1">

否：<input type="radio" name="select3" value="0"><hr/>

<input type="submit" value="提交">

<input type="reset" value="重置">

</form>

</body>

</html>

[复制代码](javascript:void(0);)

[复制代码](javascript:void(0);)

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<%@page import="java.util.\*" %>

<%@page import="java.io.\*" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>题目</title>

</head>

<body>

<%

//设置解码方式，对于简体中文，使用gb2312解码

request.setCharacterEncoding("gb2312");

//下面依次获取表单域的值

int num=Integer.parseInt(request.getParameter("num"));

int random=Integer.parseInt(request.getParameter("random"));

int select1=Integer.parseInt(request.getParameter("select1"));

int select2=Integer.parseInt(request.getParameter("select2"));

int select3=Integer.parseInt(request.getParameter("select3"));

%>

<%!

int count=0;

boolean index1=true;

%>

<%!

class MathsCalculate {

private Stack<Character> priStack = new Stack<Character>();// 操作符栈

private Stack<Integer> numStack = new Stack<Integer>();;// 操作数栈

/\*\*

\* 传入需要解析的字符串，返回计算结果(此处因为时间问题，省略合法性验证)

\* @param str 需要进行技术的表达式

\* @return 计算结果

\*/

public int caculate(String str) {

// 1.判断string当中有没有非法字符

String temp;// 用来临时存放读取的字符

// 2.循环开始解析字符串，当字符串解析完，且符号栈为空时，则计算完成

StringBuffer tempNum = new StringBuffer();// 用来临时存放数字字符串(当为多位数时)

StringBuffer string = new StringBuffer().append(str);// 用来保存，提高效率

while (string.length() != 0) {

temp = string.substring(0, 1);

string.delete(0, 1);

// 判断temp，当temp为操作符时

if (!isNum(temp))

{

// 1.此时的tempNum内即为需要操作的数，取出数，压栈，并且清空tempNum

if (!"".equals(tempNum.toString())) {

// 当表达式的第一个符号为括号

int num = Integer.parseInt(tempNum.toString());

numStack.push(num);

tempNum.delete(0, tempNum.length());

}

// 用当前取得的运算符与栈顶运算符比较优先级：若高于，则因为会先运算，放入栈顶；若等于，因为出现在后面，所以会后计算，所以栈顶元素出栈，取出操作数运算；

// 若小于，则同理，取出栈顶元素运算，将结果入操作数栈。

// 判断当前运算符与栈顶元素优先级，取出元素，进行计算(因为优先级可能小于栈顶元素，还小于第二个元素等等，需要用循环判断)

while ((!compare(temp.charAt(0))) && (!priStack.empty())) {

int a = (int) numStack.pop();// 第二个运算数

int b = (int) numStack.pop();// 第一个运算数

char ope = priStack.pop();

int result = 0;// 运算结果

switch (ope) {

// 如果是加号或者减号，则

case '+':

result = b + a;

// 将操作结果放入操作数栈

numStack.push(result);

break;

case '-':

result = b - a;

// 将操作结果放入操作数栈

numStack.push(result);

break;

case '\*':

result = b \* a;

// 将操作结果放入操作数栈

numStack.push(result);

break;

case '/':

result = b / a;// 将操作结果放入操作数栈

numStack.push(result);

break;

}

}

// 判断当前运算符与栈顶元素优先级， 如果高，或者低于平，计算完后，将当前操作符号，放入操作符栈

if (temp.charAt(0) != '#') {

priStack.push(new Character(temp.charAt(0)));

if (temp.charAt(0) == ')') {// 当栈顶为'('，而当前元素为')'时，则是括号内以算完，去掉括号

priStack.pop();

priStack.pop();

}

}

}

else

// 当为非操作符时（数字）

tempNum = tempNum.append(temp);// 将读到的这一位数接到以读出的数后(当不是个位数的时候)

}

return numStack.pop();

}

/\*\*

\* 判断传入的字符是不是0-9的数字

\*

\* @param str

\* 传入的字符串

\* @return

\*/

private boolean isNum(String temp) {

return temp.matches("[0-9]");

}

/\*\*

\* 比较当前操作符与栈顶元素操作符优先级，如果比栈顶元素优先级高，则返回true，否则返回false

\*

\* @param str 需要进行比较的字符

\* @return 比较结果 true代表比栈顶元素优先级高，false代表比栈顶元素优先级低

\*/

private boolean compare(char str) {

if (priStack.empty()) {

// 当为空时，显然 当前优先级最低，返回高

return true;

}

char last = (char) priStack.lastElement();

// 如果栈顶为'('显然，优先级最高，')'不可能为栈顶。

if (last == '(') {

return true;

}

switch (str) {

case '#':

return false;// 结束符

case '(':

// '('优先级最高,显然返回true

return true;

case ')':

// ')'优先级最低，

return false;

case '\*': {

// '\*/'优先级只比'+-'高

if (last == '+' || last == '-')

return true;

else

return false;

}

case '/': {

if (last == '+' || last == '-')

return true;

else

return false;

}

// '+-'为最低，一直返回false

case '+':

return false;

case '-':

return false;

}

return true;

}

}

%>

<%!

class SiZe

{

public String ZS\_Sum(String dividend,String sign,String divider){ //求整数算式的结果并输出运算结果

String result=null;

int dd=Integer.parseInt(dividend);

int dr=Integer.parseInt(divider);

if(sign=="+")

result=String.valueOf(dd+dr);

if(sign=="-")

result=String.valueOf(dd-dr);

if(sign=="\*")

result=String.valueOf(dd\*dr);

if(sign=="/"){

if(dd%dr==0)

result=String.valueOf(dd/dr);

else

result=String.valueOf(dd)+"/"+String.valueOf(dr);

}

return result;

}

public String FS\_Sum(int dividend1,int divider1,String sign,int dividend2,int divider2)

{//求分数算式的结果并输出运算结果

String result;

String sum=null;

int dividend=0,divider=0;

int gcd;

if(sign=="+")

{

dividend=dividend1\*divider2+dividend2\*divider1;

divider=divider1\*divider2;

int r1;

int x1=dividend;

int y1=divider;

while(y1>0)

{

r1=x1%y1;

x1=y1;

y1=r1;

}

dividend=dividend/x1;

divider=divider/x1;

sum=String.valueOf(dividend)+"/"+String.valueOf(divider);

}

if(sign=="-")

{

dividend=dividend1\*divider2-dividend2\*divider1;

divider=divider1\*divider2;

int r1;

int x1=dividend;

int y1=divider;

while(y1>0)

{

r1=x1%y1;

x1=y1;

y1=r1;

}

dividend=dividend/x1;

divider=divider/x1;

sum=String.valueOf(dividend)+"/"+String.valueOf(divider);

}

if(sign=="\*")

{

dividend=dividend1\*dividend2;

divider=divider1\*divider2;

int r1;

int x1=dividend;

int y1=divider;

while(y1>0)

{

r1=x1%y1;

x1=y1;

y1=r1;

}

dividend=dividend/x1;

divider=divider/x1;

sum=String.valueOf(dividend)+"/"+String.valueOf(divider);

}

if(sign=="/")

{

dividend=dividend1\*divider2;

divider=divider1\*dividend2;

int r1;

int x1=dividend;

int y1=divider;

while(y1>0)

{

r1=x1%y1;

x1=y1;

y1=r1;

}

dividend=dividend/x1;

divider=divider/x1;

}

if(dividend%divider==0)

sum=String.valueOf(dividend);

else

sum=String.valueOf(dividend)+"/"+String.valueOf(divider);

return sum;

}

public String FenShu(int dividend1,int divider1,String sign,int dividend2,int divider2)

{

String dividend=null;

String divider=null;

String dd1=null,dd2=null,dr1=null,dr2=null;

String suanshi=null;

dd1=String.valueOf(dividend1);

dd2=String.valueOf(dividend2);

dr1=String.valueOf(divider1);

dr2=String.valueOf(divider2);

dividend=dd1+"/"+dr1;

divider=dd2+"/"+dr2;

if(sign=="+")

suanshi=dividend + " + " + divider + "=?";

if(sign=="-")

suanshi=dividend + " - " + divider + "=?";

if(sign=="\*")

suanshi=dividend + " \* " + divider + "=?";

if(sign=="/")

suanshi=dividend + " / " + divider + "=?";

return suanshi;

}

public String ZhengShu(int dividend1,String sign,int divider1)

{

String divider=null;

String dividend=null;

String suanshi=null;

dividend=String.valueOf(dividend1);

divider=String.valueOf(divider1);

if(sign=="+")

suanshi=dividend + " + " + divider + "=?";

if(sign=="-")

suanshi=dividend + " - " + divider + "=?";

if(sign=="\*")

suanshi=dividend + " \* " + divider + "=?";

if(sign=="/")

suanshi=dividend + " / " + divider + "=?";

return suanshi;

}

}

%>

<%SiZe fo=new SiZe();

MathsCalculate operate=new MathsCalculate();

%>

<%

Random rand=new Random();

String divider;

String dividend;

String sign=null;

int dividend1=0,divider1=0,dividend2=0,divider2=0;

%>

<%

if(count>=num)

{

index1=false;

}

while(index1)

{

int mark=rand.nextInt();

int key=rand.nextInt();

boolean index=false;

if(select1==0)

{

switch(Math.abs(mark%2))

{

case 0:sign = "+";break;

case 1:sign = "-";break;

}

}

else

{

switch(Math.abs(mark%4))

{

case 0:sign = "+";break;

case 1:sign = "-";break;

case 2:sign = "\*";break;

case 3:sign = "/";break;

}

}

if(Math.abs(key%3)==0)

{

dividend1=rand.nextInt(random);

divider1=rand.nextInt(random);

dividend2=rand.nextInt(random);

divider2=rand.nextInt(random);

if((divider1==0)||(divider2==0)||(dividend1>divider1)||(dividend2>divider2))

{

index=true;

}

else

{

int r1;

int x1=dividend1;

int y1=divider1;

while(y1>0)

{

r1=x1%y1;

x1=y1;

y1=r1;

}

dividend1=dividend1/x1;

divider1=divider1/x1;

int r2;

int x2=dividend2;

int y2=divider2;

while(y2>0)

{

r2=x2%y2;

x2=y2;

y2=r2;

}

dividend2=dividend2/x2;

divider2=divider2/x2;

if(sign=="+")

{

if((dividend1\*divider2+dividend2\*divider1)>(divider1\*divider2))

{

index=true;

}

}

if(sign=="-")

{

if(dividend1\*divider2<dividend2\*divider1)

{

index=true;

}

}

if(sign=="/")

{

if((dividend2==0)||(dividend1\*divider2>divider1\*dividend2))

{

index=true;

}

if(select2==0)

{

if((divider1\*dividend2)%(dividend1\*divider2)!=0)

{

index=true;

}

}

}

}

String path = "F:\\eclipse-java-mars-R-win32-x86\_64\\eclipse\\read.txt" ;// 定义文件路径

FileReader fr = new FileReader(path); //这里定义一个字符流的输入流的节点流，用于读取文件（一个字符一个字符的读取）

FileReader fr1 = new FileReader(path);

BufferedReader br = new BufferedReader(fr); // 在定义好的流基础上套接一个处理流，用于更加效率的读取文件（一行一行的读取）

BufferedReader br1 = new BufferedReader(fr1);

while(true)

{

if(br1.readLine()!=null)

{

if(br.readLine().equals(fo.FenShu(dividend1,divider1,sign,dividend2,divider2)))

index=true;

}

else

break;

}

br.close();

br1.close();

if(index==false)

{

%>

<%=fo.FenShu(dividend1,divider1,sign,dividend2,divider2) %>

<%

java.io.FileWriter fw=new java.io.FileWriter("read.txt",true);

java.io.PrintWriter pw=new java.io.PrintWriter(fw);

pw.println(fo.FenShu(dividend1,divider1,sign,dividend2,divider2));

pw.close();

fw.close();

%>

<form id="form2" method="post" action="Result.jsp">

<input type="text" name="sum">

<input type="hidden" name="result" value=<%=fo.FS\_Sum(dividend1, divider1, sign, dividend2, divider2) %>>

<input type="hidden" name="num" value=<%=num%>>

<input type="hidden" name="random" value=<%=random %>>

<input type="hidden" name="select1" value=<%=select1 %>>

<input type="hidden" name="select2" value=<%=select2 %>>

<input type="submit" value="提交">

</form>

<br/>

<%

count++;

break;

}

}

if(Math.abs(key%3)==1)

{

dividend1=rand.nextInt(random);

divider1=rand.nextInt(random);

if(sign=="-")

{

if(dividend1-divider1<0)

{

index=true;

}

}

if(sign=="/")

{

if((divider1==0)||(dividend1>divider1))

{

index=true;

}

if(select2==0)

{

if((dividend1%divider1)!=0)

{

index=true;

}

}

}

String path = "F:\\eclipse-java-mars-R-win32-x86\_64\\eclipse\\read.txt" ;// 定义文件路径

FileReader fr = new FileReader(path); //这里定义一个字符流的输入流的节点流，用于读取文件（一个字符一个字符的读取）

FileReader fr1 = new FileReader(path);

BufferedReader br = new BufferedReader(fr); // 在定义好的流基础上套接一个处理流，用于更加效率的读取文件（一行一行的读取）

BufferedReader br1 = new BufferedReader(fr1);

while(true)

{

if(br1.readLine()!=null)

{

if(br.readLine().equals(fo.ZhengShu(dividend1,sign,divider1)))

index=true;

}

else

break;

}

br.close();

br1.close();

if(index==false)

{

dividend=String.valueOf(dividend1);

divider=String.valueOf(divider1);

%>

<%=fo.ZhengShu(dividend1,sign,divider1) %>

<%

java.io.FileWriter fw=new java.io.FileWriter("read.txt",true);

java.io.PrintWriter pw=new java.io.PrintWriter(fw);

pw.println(fo.ZhengShu(dividend1,sign,divider1));

pw.close();

fw.close();

%>

<form id="form2" method="post" action="Result.jsp">

<input type="text" name="sum">

<input type="hidden" name="result" value=<%=fo.ZS\_Sum(dividend, sign, divider) %>>

<input type="hidden" name="num" value=<%=num%>>

<input type="hidden" name="random" value=<%=random %>>

<input type="hidden" name="select1" value=<%=select1 %>>

<input type="hidden" name="select2" value=<%=select2 %>>

<input type="submit" value="提交">

</form>

<br/>

<%

count++;

break;

}

}

if(Math.abs(key%3)==2)

{

int kuohao=rand.nextInt();

}

}

%>

<%

if(!index1)

{

out.println("题目已出完！");

%>

<%

}

%>

</body>

</html>

[复制代码](javascript:void(0);)

[复制代码](javascript:void(0);)

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>结果</title>

</head>

<body>

<%

String sum=request.getParameter("sum");

String result=request.getParameter("result");

%>

<%

if(result.equals(sum))

out.println("正确！");

else

{

out.println("错误！");

out.println("正确结果为："+result);

}

%>

<%

String num=request.getParameter("num");

String random=request.getParameter("random");

String select1=request.getParameter("select1");

String select2=request.getParameter("select2");

%>

<br>

<form id=form3 action="request.jsp" method="post">

<input type="hidden" name="num" value=<%=num%>>

<input type="hidden" name="random" value=<%=random %>>

<input type="hidden" name="select1" value=<%=select1 %>>

<input type="hidden" name="select2" value=<%=select2 %>>

<input type="submit" value="返回">

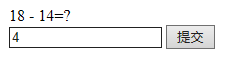
</form>

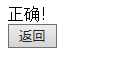
</body>

</html>

三.程序结果截图：







http://images2015.cnblogs.com/blog/811552/201604/811552-20160410213819718-2041076289.png

http://images2015.cnblogs.com/blog/811552/201604/811552-20160410213845140-598557854.png



四.总结

1项目计划总结：

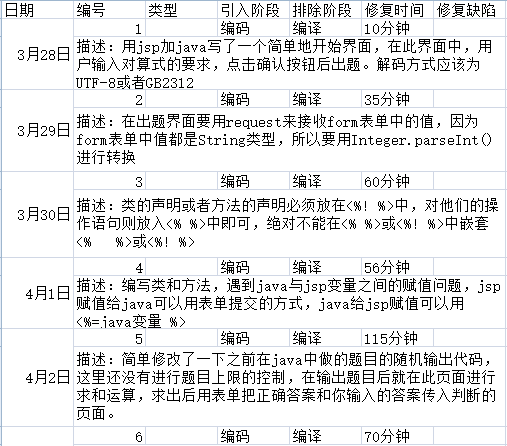


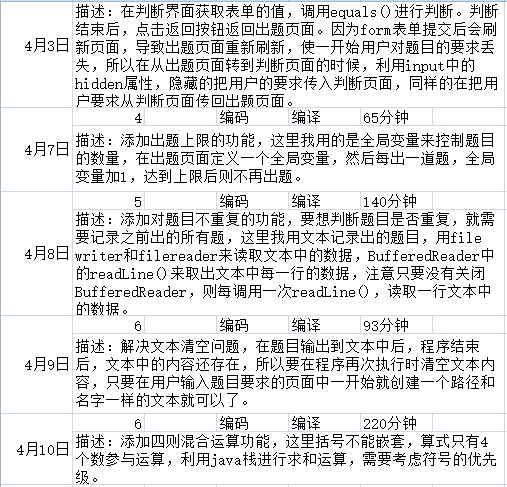


2时间记录日志：



3.缺陷记录日志：





博客地址：http://www.cnblogs.com/java-meng/p/5375621.html