四则运算3

2016-03-29 16:38

一：设计思路：

1.输入

1）将条件以0和1的形式输入，通过判断0或1来执行相应的功能；

2.计算

1）建一个主类，在主类中编写方法，其中两个综合方法，分别为整数运算和分数运算，在这两个方法中再调用其他小函数，包括求运算符方法，最大公约数方法，输出方法。此次需在此基础上添加一个四则运算的方法，并且分别编写求整数运算结果判断，分数运算结果判断，四则运算结果判断的方法。需要考虑的是在有分数的情况下，计算结果时，需要进行通分化简，然后进行判断。

2）是否有乘除法：如果有则将各种符号通过switch语句控制，产生0-4的随机数来控制输出加减法还是输出乘除法；如果没有有乘除法，将上边随机数的范围改为0-2。

3)加减有无负数：在求加减法的函数中加入switch语句，产生0-3的随机数，然后通过判断随机数来确定是否为负数；

4）除法有无余数：在求除法的函数中添加判断求余是否为0；

5）是否有多位四则运算题：这里把此问题与上述其他选择分开，计算比较简单的整数四则运算。

3.输出

在综合方法中调用输出函数，参数分别是参与运算的数，输出结果；

二：源程序：

package sizeyunsuan;

import java.util.Random;

import java.util.Scanner;

public class FourOperations {

public static void main(String[] args){

Scanner in=new Scanner(System.in);

FourOperations FO=new FourOperations();

Random rand=new Random();

int num;

System.out.println("请输入题目的数量：");

num=in.nextInt();

System.out.println("是否有乘除法？（1：有 ，0：没有）");

int select1=in.nextInt();

System.out.println("加减有无负数？（1：有 ，0：没有）");

int select2=in.nextInt();

System.out.println("除法有无余数？（1：有 ，0：没有）");

int select3=in.nextInt();

System.out.println("请输入取值范围上限：");

int random=in.nextInt();

int n[]={0};

for(int i=0;i<num+n[0];i++)

{

int key=rand.nextInt();

if(Math.abs(key%2)==0)

FO.ZhengShu(n,select1,select2,select3,random);

if(Math.abs(key%2)==1)

FO.FenShu(n,select1,select2,select3,random);

}

in.close();

}

public String JiaJian(){

String sign=null;

Random rand=new Random();

int mark=rand.nextInt();

switch(Math.abs(mark%2))

{

case 0:sign = "+";break;

case 1:sign = "-";break;

}

return sign;

}

public String ChengChu(){

String sign=null;

Random rand=new Random();

int mark=rand.nextInt();

switch(Math.abs(mark%4))

{

case 0:sign = "+";break;

case 1:sign = "-";break;

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

case 3:sign = "/";break;

}

return sign;

}

public void OutWay(String dividend,String sign,String divider){ //输出整数或分数运算的算式

if(sign=="+")

System.out.println(dividend + " + " + divider+"=?");

if(sign=="-")

System.out.println(dividend + "-" + "(" + divider + ")");

if(sign=="\*")

System.out.println(dividend + " \* " + divider);

if(sign=="/")

System.out.println(dividend + " / " + divider);

}

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

Scanner in=new Scanner(System.in);

String result;

String sum=null;

if(sign=="+")

sum=String.valueOf(dividend+divider);

if(sign=="-")

sum=String.valueOf(dividend-divider);

if(sign=="\*")

sum=String.valueOf(dividend\*divider);

if(sign=="/"){

if(dividend%divider==0)

sum=String.valueOf(dividend/divider);

else

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

}

System.out.println("请输入正确结果：");

result=in.nextLine();

if(result.equals(sum))

System.out.println("True!");

else

System.out.println("False! The result is:"+sum);

}

public void 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

Scanner in=new Scanner(System.in);

String result;

String sum=null;

int dividend=0,divider=0;

int gcd;

if(sign=="+")

{

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

divider=divider1\*divider2;

gcd=GCD(dividend,divider);

dividend=dividend/gcd;

divider=divider/gcd;

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

}

if(sign=="-")

{

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

divider=divider1\*divider2;

gcd=GCD(dividend,divider);

dividend=dividend/gcd;

divider=divider/gcd;

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

}

if(sign=="\*")

{

dividend=dividend1\*dividend2;

divider=divider1\*divider2;

gcd=GCD(dividend,divider);

dividend=dividend/gcd;

divider=divider/gcd;

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

}

if(sign=="/"){

dividend=dividend1\*divider2;

divider=divider1\*dividend2;

gcd=GCD(dividend,divider);

dividend=dividend/gcd;

divider=divider/gcd;

}

if(dividend%divider==0)

sum=String.valueOf(dividend);

else

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

System.out.println("请输入正确结果：");

result=in.nextLine();

if(result.equals(sum))

System.out.println("True!");

else

System.out.println("False! The result is:"+sum);

}

/\*public void FO\_Sum(String suanshi){ //求四则运算算式的结果并输出运算结果

Scanner in=new Scanner(System.in);

int length=suanshi.length();

int number[]=new int[(length+1)/2];

char sign[]=new char[(length-1)/2];

for(int b=0;b<length;b++){

number[b]=suanshi.charAt(2\*b);

sign[b]=suanshi.charAt(2\*b+1);

}

int i=0,j=0,k=1;

int sum1[]=new int[length];

int sum2[]=new int[length];

for(int a=0;a<length;a++){

sum1[a]=0;

sum2[a]=0;

}

while(true)

{

if(sign[j]=='+')

{

sum1[k]=number[j]+sum1[k-1];

k++;

}

if(sign[j]=='-')

{

sum1[k]=number[j]-sum1[k-1];

k++;

}

if(sign[j]=='\*')

{

sum2[i]=number[j]\*number[j+1];

number[j+1]=sum2[i];

i++;

}

if(sign[j]=='/')

{

sum2[i]=number[j]/number[j+1];

number[j+1]=sum2[i];

i++;

}

j++;

if(j>(length-1)/2)

break;

}

int result;

System.out.println("请输入正确结果：");

result=in.nextInt();

if(result==sum1[k-1])

System.out.println("True!");

else

System.out.println("False! The result is:"+sum1[k-1]);

}\*/

public void FenShu(int n[],int s1,int s2,int s3,int random){

String dividend=null;

String divider=null;

FourOperations FO=new FourOperations();

String sign=null;

Random rand=new Random();

int dividend1,divider1,dividend2,divider2;

boolean index=false;

dividend1=rand.nextInt(random);

divider1=rand.nextInt(random);

dividend2=rand.nextInt(random);

divider2=rand.nextInt(random);

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

if((divider1==0)||(divider2==0)){

n[0]++;

index=true;

}

else{

int gcd;

gcd=GCD(dividend1,divider1);

dividend1=dividend1/gcd;

divider1=divider1/gcd;

gcd=GCD(dividend2,divider2);

dividend2=dividend2/gcd;

divider2=divider2/gcd;

if(s1==1)

sign=FO.ChengChu();

else

sign=FO.JiaJian();

if(s2==1){

if(sign=="+"||sign=="-"){

int key1=rand.nextInt();

switch(Math.abs(key1%3)){

case 0:{dividend1=-dividend1;break;}

case 1:{dividend1=-dividend1;dividend2=-dividend2;break;}

case 2:{dividend2=-dividend2;break;}

}

}

}

dd1=String.valueOf(dividend1);

dd2=String.valueOf(dividend2);

dr1=String.valueOf(divider1);

dr2=String.valueOf(divider2);

dividend=dd1+"/"+dr1;

divider=dd2+"/"+dr2;

if(sign=="/"){

if(dividend2==0){

n[0]++;

index=true;

}

if(s3==0){

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

n[0]++;

index=true;

}

}

}

if(!index)

{

OutWay(dividend,sign,divider);

FS\_Sum(dividend1,divider1,sign,dividend2,divider2);

}

}

}

public void ZhengShu(int n[],int s1,int s2,int s3,int random){

String divider=null;

String dividend=null;

FourOperations FO=new FourOperations();

String sign=null;

Random rand=new Random();

int dividend1,divider1;

dividend1=rand.nextInt(random);

divider1=rand.nextInt(random);

boolean index=false;

if(s1==1)

sign=FO.ChengChu();

else

sign=FO.JiaJian();

if(s2==1){

if((sign=="+")||(sign=="-")){

int key1=rand.nextInt();

switch(Math.abs(key1%3)){

case 0:{dividend1=-dividend1;break;}

case 1:{divider1=-divider1;break;}

case 2:{dividend1=-dividend1;divider1=-divider1;break;}

}

}

}

dividend=String.valueOf(dividend1);

divider=String.valueOf(divider1);

if(sign=="/"){

if(divider1==0){

n[0]++;

index=true;

}

if(s3==0){

if((dividend1%divider1)!=0){

n[0]++;

index=true;

}

}

}

if(!index){

OutWay(dividend,sign,divider);

ZS\_Sum(dividend1,sign,divider1);

}

}

/\*public void FOperation(int n[],int random){ //输出四则运算的算式

Random rand=new Random();

String sign=null;

String suanshi=null;

boolean index=false;

int isgo=1,mark;

int dividend,divider;

dividend=rand.nextInt(random);

divider=rand.nextInt(random);

suanshi=String.valueOf(dividend);

while(true){

mark=rand.nextInt();

switch(Math.abs(mark%4))

{

case 0:sign = "+";break;

case 1:sign = "-";break;

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

case 3:sign = "/";break;

}

if((sign=="/")&&(divider==0))

{

n[0]++;

index=true;

}

else if((sign=="/")&&(dividend%divider!=0))

{

dividend=divider;

n[0]++;

index=true;

}

else

{

suanshi=suanshi+sign+String.valueOf(divider);

}

if(Math.abs(isgo)==0)

{

if(!index)

{

System.out.println(suanshi);

FO\_Sum(suanshi);

break;

}

else

break;

}

else

{

divider=rand.nextInt(random);

isgo=rand.nextInt();

}

}

}\*/

public int GCD(int a,int y){

int x=Math.abs(a);

int r;

while(y>0){

r=x%y;

x=y;

y=r;

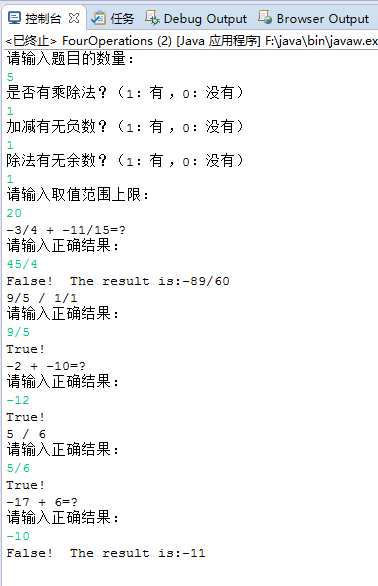
}

return x;

}

}

三：结果截图：



四.总结

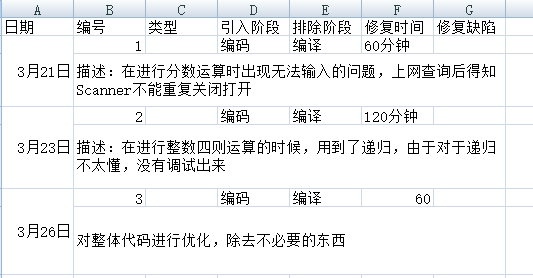
1项目计划总结：



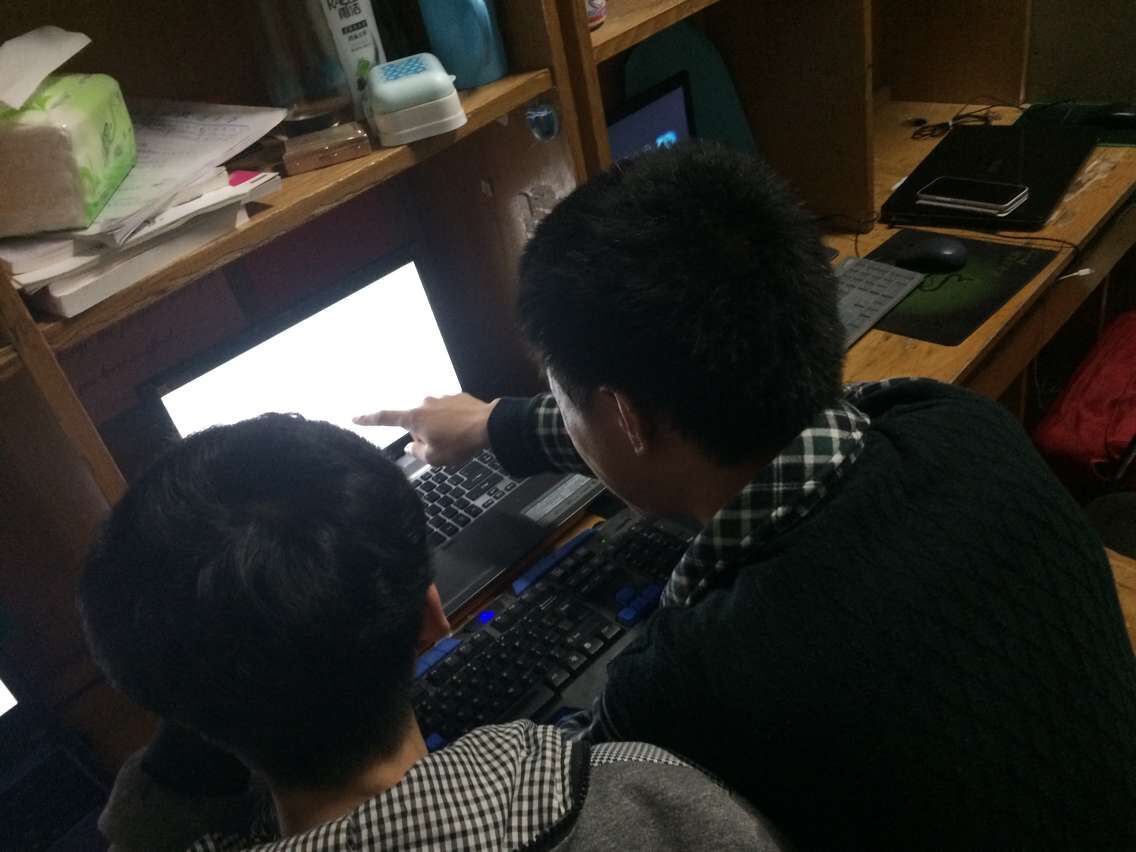
2时间记录日志：



3.缺陷记录日志：



五.工作照片



博客地址：http://www.cnblogs.com/java-meng/p/5323267.html