1. 目的

電卓を作成することでjava言語の理解を深める

1. プログラミング説明

ボタン0～9が押されたとき

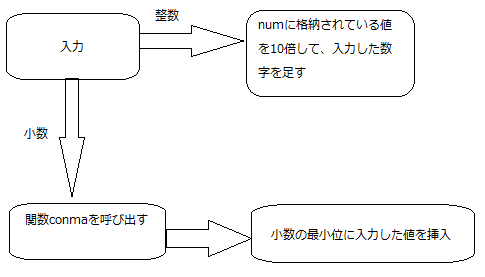


図１.数字ボタン

演算子ボタンとイコールボタン

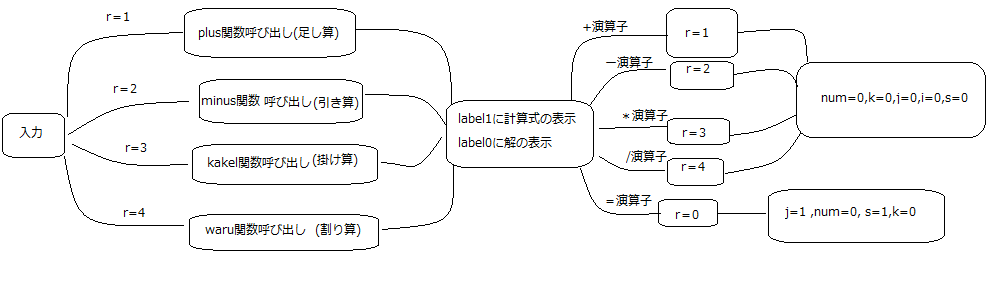


図2.演算子ボタンとイコールボタン

Cボタン：全ての変数とlabelを初期値にする

CEボタン：label0と変数iを0にする。

小数点ボタン:変数i=0のときi=0

変数num：入力した値の格納

変数num1：計算結果の格納

変数str：今までの計算式の格納

変数i：整数入力か小数入力かの判断。ボタンを入力したときi=0のときは整数入力i>0のときは小数入力となる

変数k：演算子の連続入力の防止。k=0のとき計算を行う関数を呼び出さない。k=1のとき計算を行う関数を呼び出す

変数j：演算子の連続入力の防止。j=0のときは演算子が連続入力のときである。このとき前に入力された演算子をなかったことにする。j=1とき数字ボタンが入力された後に演算子が入力されたことにな

変数r：計算方法の決定。r=1のとき足し算を行う。r=2のとき引き算を行う。r=3のとき掛け算を行う。r=4のとき割り算を行う

変数s：イコールボタンが押されたあとも連続で計算するときのエラー防止。

canma関数：i!=のとき呼び出される。これは小数第何位かを判断する。

plus関数：足し算を行う

minus関数：引き算を行う

kakel関数：掛け算を行う

waru関数：割り算を行う

レイアウト

5つのhboxを用意しそれぞれに[CE,C]、[７,８,９,/],[4,5,6,\*]、[1,2,3,-]、[0,.,=,+]を格納する。そしてvboxに[label1,label0,hbox1,hbox2,hbox3,hbox4,hbox5]を格納した。

1. 仕様
   1. 繰り返し何度も計算ができること

本プログラムでは繰り返し計算ができるよう二つの変数「num」「num1」を用いた。「num」は入力した値を格納する変数、「num1」は計算結果を格納する変数である。このように変数を二つの変数を用いることで、前に行った計算結果を「num1」に記憶したままボタン入力をすることが可能になる。

図3には8+2\*を押したときの表示である。これは足し算の後に掛け算ができることを示している。

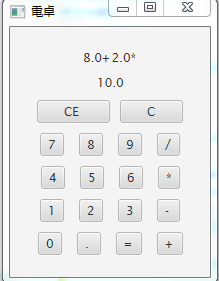


図3.繰り返し計算(1)

この後に3を入力し＝ボタンを押した後の結果を図4に示す。

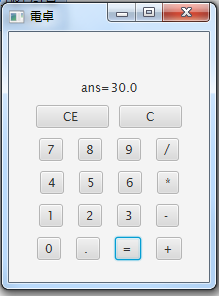


図4.繰り返し計算(2)

このように掛け算の計算結果が正確に表示されていることが確認できる。この後さらに演算子が入力されて計算を続けられる様子を図5に示す。

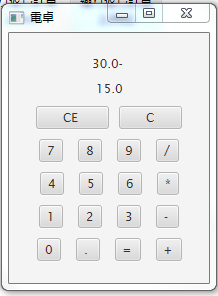


図5.繰り返し計算(3)

このよう複数の演算子を含む計算や＝で表示された後の計算の継続も可能であることが確認できる。

* 1. 小数点付きの数を使用できること

小数点付きの数を使用するために変数「i」を用いた。これは小数点ボタンを押したときの整数入力から小数入力の切り替えとなるものである。

「i」の初期値は0である。i=0のとき数字ボタンを押すと整数入力となる。このとき小数点ボタンを押すとi=1となる。i>0の場合に数字ボタンを押すと変数「num」にはが代入される。そして数字ボタンを押していくとiに1が足されていく。これで小数点付きの数の実現が可能となった。

図6には小数点入力の結果を示す。

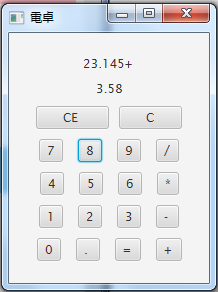


図6.小数点(1)

このように小数点入力が可能であることを示している。図7には小数の計算結果を示す

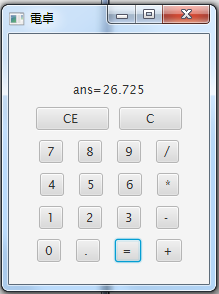


図7.小数点(2)

このように小数を含む計算結果が正確であること確認できる。

* 1. 加減乗除が可能なこと

どの計算を行うかの判断を変数rを用いて行う。＋演算子を押したときr=1、－はr=2、\*はr=3、/はr=4を代入する。それぞれrの値を比べてr=1ならばplus関数を呼び出し足し算を行う。r=2ならばminus関数を呼び出し引き算を行う。r=3ならばkakel関数を呼び出し掛け算を行う。r=4ならばwaru関数を呼び出し割り算を行う

図8～図11により加減乗除の計算結果を示す。

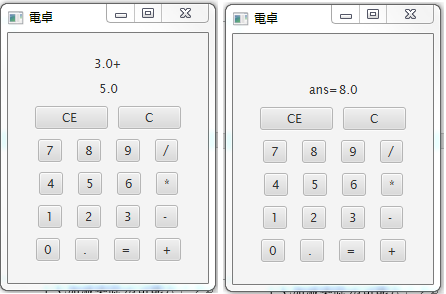


図8.足し算

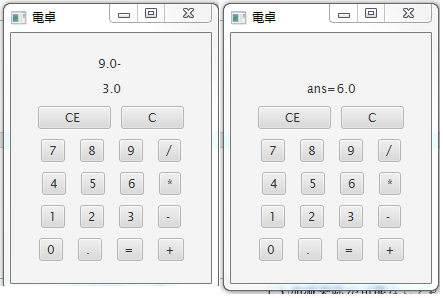


図9.引き算



図10.掛け算

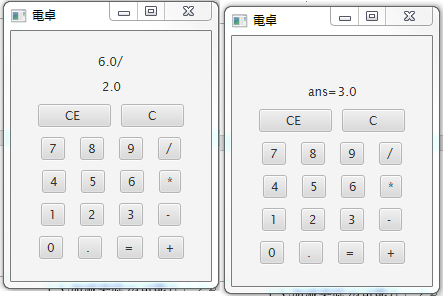


図11.割り算

* 1. 不正入力を行っても再起動なしに通常の計算に復帰できる

1. 0で割ったとき



図12.不正入力(1)

0で割ると｢0で割れません｣を表示される。このとき数字を押しても変化は無いがCをおすことで初期状態にもどるため通常通りの計算が可能である。

1. 演算子を連続入力

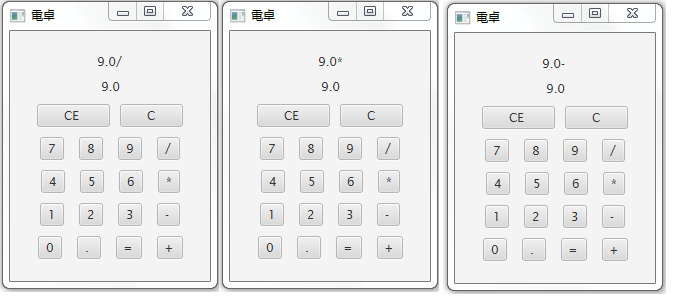


図13.不正入力(2)

演算子を連続入力した場合は前に入力した演算子はキャンセルされて最後に入力された演算子が表示される。

1. 小数の状態で小数点を押す

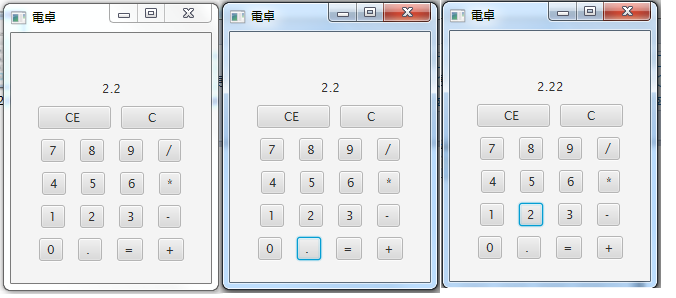


図14.不正入力(3)

小数の状態で小数点を押すと押された小数点は無視されるため続けて計算ができる。

1. 考察

今回のプログラムでは電卓作成を行った。

今回は計算に関係あるプログラミング部分を関数にした。これによって全行数を450から400未満にまで減らすことに成功した。関数を効率的に用いることで簡潔なプログラミングを書けることができると理解できた。

電卓における不正入力を回避するために変数と条件分岐を使用した。そのため精度の高い電卓をつくるためにはより複雑なプログラミングにならざるを得なくなった。さらにどの変数がどの不正入力を回避するためのものかが非常にわかりにくい。これを解決するためにはコメントを効率よく使うべきである。

小数の入力時に同じ数を一定数入力すると、図15のような表示になる。

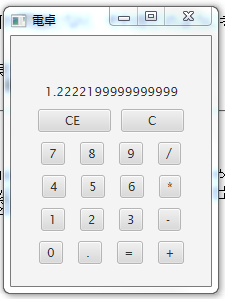


図15.エラー

これは本来なら「1.22222」と表示されるはずである。このまま計算した場合、答えには小さな誤差が生じる。しかし入力したはずの値と表示された値の差は1×10-10%以下である。誤差が起こるのは電卓としては欠陥であるがこの程度の差ならば問題は無い。

1. プログラム

package dentaku;

import javafx.application.Application;

import javafx.event.ActionEvent;

import javafx.event.EventHandler;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.layout.StackPane;

import javafx.stage.Stage;

import javafx.scene.control.Label;

import javafx.scene.layout.HBox;

import javafx.scene.layout.VBox;

/\*\*

\*

\* @author Owner

\*/

public class Dentaku extends Application {

Stage stage;

Scene scene;

Label label0,label1,label2;

Button btn0,btn1,btn2,btn3,btn4,btn5,btn6,btn7, btn8, btn9,

point,plus,minus,kakel,waru,ans,del,aldel,back,plumai;

double num,num1,num2,b;

int i,j,k,r,s;

//i 小数判断　j 演算子を連続入力したときの変換

String str;

@Override

public void start(Stage primaryStage) {

i=0;

k=0;

j=1;

s=0;

num=0;

stage=primaryStage;

//ボタンの表示

btn0=new Button("0"); btn1=new Button("1");

btn2=new Button("2"); btn3=new Button("3");

btn4=new Button("4"); btn5=new Button("5");

btn6=new Button("6"); btn7=new Button("7");

btn8=new Button("8"); btn9=new Button("9");

plus=new Button("+"); minus=new Button("-");

kakel=new Button("\*"); waru=new Button("/");

aldel=new Button(" C "); del=new Button(" CE ");

point=new Button(". "); ans=new Button("=");

plumai=new Button("±");

label1=new Label("");

label0=new Label("0");

point.setOnAction(e->{

if(i==0)

i=i+1;

});

btn0.setOnAction(e->{

if(i==0)

num=num\*10+0;

else if(i>0){

b=canma(i,b);

num=num+(0/b);

i++;

}

k++;

j++;

label0.setText(String.valueOf(num));

});

btn1.setOnAction(e->{

if(i==0)

num=num\*10+1;

else if(i>0){

b=canma(i,b);

num=num+(1/b);

i++;

}

k++;

j++;

label0.setText(String.valueOf(num));

});

btn2.setOnAction(e->{

if(i==0)

num=num\*10+2;

else if(i>0){

b=canma(i,b);

num=num+(2/b);

i++;

}

k++;

j++;

label0.setText(String.valueOf(num));

});

btn3.setOnAction(e->{

if(i==0)

num=num\*10+3;

else if(i>0){

b=canma(i,b);

num=num+(3/b);

i++;

}

k++;

j++;

label0.setText(String.valueOf(num));

});

btn4.setOnAction(e->{

if(i==0)

num=num\*10+4;

else if(i>0){

b=canma(i,b);

num=num+(4/b);

i++;

}

k++;

j++;

label0.setText(String.valueOf(num));

});

btn5.setOnAction(e->{

if(i==0)

num=num\*10+5;

else if(i>0){

b=canma(i,b);

num=num+(5/b);

i++;

}

k++;

j++;

label0.setText(String.valueOf(num));

});

btn6.setOnAction(e->{

if(i==0)

num=num\*10+6;

else if(i>0){

b=canma(i,b);

num=num+(6/b);

i++;

}

k++;

j++;

label0.setText(String.valueOf(num));

});

btn7.setOnAction(e->{

if(i==0)

num=num\*10+7;

else if(i>0){

b=canma(i,b);

num=num+(7/b);

i++;

}

k++;

j++;

label0.setText(String.valueOf(num));

});

btn8.setOnAction(e->{

if(i==0)

num=num\*10+8;

else if(i>0){

b=canma(i,b);

num=num+(8/b);

i++;

}

k++;

j++;

label0.setText(String.valueOf(num));

});

btn9.setOnAction(e->{

if(i==0)

num=num\*10+9;

else if(i>0){

b=canma(i,b);

num=num+(9/b);

i++;

}

k++;

j++;

label0.setText(String.valueOf(num));

});

ans.setOnAction(e->{

if(r==0)

num1=num;

if(k!=0) {

num1=calcu(num1, num, r);

}

label1.setText("");

if(r==4 && num==0)

label0.setText("0では割れません");

else

label0.setText("ans="+String.valueOf(num1));

str=(String.valueOf(num1));

j=1; num=0; s=1; k=0;

});

del.setOnAction(e->{

i=0;

k=0;

num=0;

label0.setText(String.valueOf(0));

});

aldel.setOnAction(e->{

num=0; num1=0; r=0; k=0; j=1;i=0;

label0.setText(String.valueOf(0));

label1.setText("");

str=("");

});

plus.setOnAction(e->{

if(r==0)

num1=num;

if(k!=0) {

num1=calcu(num1, num, r);

}

if(j==0 || s==1)

label1.setText(str+"+");

else{

str=(label1.getText().trim()+String.valueOf(num));

label1.setText(str+"+");

if(r==4 && num==0)

label0.setText("0では割れません");

else

label0.setText(String.valueOf(num1));

}

num=0; r=1; k=0; j=0; i=0; s=0;

});

minus.setOnAction(e->{

if(r==0)

num1=num;

if(k!=0) {

num1=calcu(num1, num, r);

}

if(j==0 || s==1)

label1.setText(str+"-");

else{

str=(label1.getText().trim()+String.valueOf(num));

label1.setText(str+"-");

if(r==4 && num==0)

label0.setText("0では割れません");

else

label0.setText(String.valueOf(num1));

}

num=0; r=2; k=0; j=0; i=0; s=0;

});

kakel.setOnAction(e->{

if(r==0)

num1=num;

if(k!=0) {

num1=calcu(num1, num, r);

}

if(j==0 || s==1)

label1.setText(str+"\*");

else{

str=(label1.getText().trim()+String.valueOf(num));

label1.setText(str+"\*");

if(r==4 && num==0)

label0.setText("0では割れません");

else

label0.setText(String.valueOf(num1));

}

num=0; r=3; k=0; j=0; i=0; s=0;

});

waru.setOnAction(e->{

if(r==0)

num1=num;

if(k!=0) {

num1=calcu(num1, num, r);

}

if(j==0 || s==1)

label1.setText(str+"/");

else{

str=(label1.getText().trim()+String.valueOf(num));

label1.setText(str+"/");

if(r==4 && num==0)

label0.setText("0では割れません");

else

label0.setText(String.valueOf(num1));

}

num=0; r=4; k=0; j=0; i=0; s=0;

});

//レイアウト

HBox hbox1,hbox2,hbox3,hbox4,hbox5;

hbox1=new HBox(10,del,aldel);

hbox2=new HBox(15,btn7,btn8,btn9,waru);

hbox3=new HBox(15,btn4,btn5,btn6,kakel);

hbox4=new HBox(15,btn1,btn2,btn3,minus);

hbox5=new HBox(15,btn0,point,ans,plus);

hbox1.setAlignment(Pos.CENTER);

hbox2.setAlignment(Pos.CENTER);

hbox3.setAlignment(Pos.CENTER);

hbox4.setAlignment(Pos.CENTER);

hbox5.setAlignment(Pos.CENTER);

VBox pane=new VBox(10,label1,label0,hbox1,hbox2,hbox3,hbox4,hbox5);

pane.setAlignment(Pos.CENTER);

scene =new Scene(pane,200,250);

primaryStage.setScene(scene);

primaryStage.setTitle("電卓");

primaryStage.show();

}

//小数

public static double canma(int i,double b)

{

b=1;

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

b=b\*10;

}

return b;

}

//どの計算方法かの判断

public static double calcu(double num1,double num2,int r)

{

if(r==1)

num1=plus(num1,num2);

else if(r==2)

num1=minus(num1,num2);

else if(r==3)

num1=kakel(num1,num2);

else if(r==4)

if(num2==0);

else

num1=waru(num1,num2);

return num1;

}

public static double plus(double num1,double num2)

{

num1=num1+num2;

return num1;

}

public static double minus(double num1,double num2)

{

num1=num1-num2;

return num1;

}

public static double kakel(double num1,double num2)

{

num1=num1\*num2;

return num1;

}

public static double waru(double num1,double num2)

{

num1=num1/num2;

return num1;

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

launch(args);

}

}