NATIONAL INSTITUTE OF TECHNOLOGY RAIPUR

HARSH KUMAR CHOUDHARY

19118034, IT 3rd Sem.

JAVA – CALCULATOR

DECEMBER 13, 2020.

ABSTRACT

Java is a beautiful programming language, as it gives a lot of features such as synchronization, graphical user interface and object orientation. So in this paper, we will be using Java to create one of the wonderful device which is always necessary for mathematical calculations. It's a calculator made by using features of Java AWT, SWING, METHODS and many other effects.

We will be discussing the features of java used in this paper and finally at last will be analyzing it's OOM concepts with some diagrams. We will also be using some basic methods in java such as "log" and "sin" and some others.

INTRODUCTION

Java is robust and well known programming language, used today for object oriented programming, android development and other components of web such as applet and GUI.

Java has a huge library of classes and a very large number of methods are available in each classes. It has classes that supports GUI and we will be using this feature to create and handle our problem in this paper.

In this paper we will be using "Math", "Awt", and "Object" class and some other basics methods of strings etc. Also we will be developing basic algorithm to cope with the problem described.

Basic knowledge of Java and some of its classes is a prerequisite for understanding the code. Hence this project is of utmost importance to college students involved with higher calculations and many other engineering fields.

PROBLEM:

Recent studies has shown that on higher studies student requires to a lot of numerical calculations with very large data. This large data can only be handled by computers. Use of Scientific calculators is a must. But some basic mathematics operations are there which cannot be performed by a scientific calculator easily, we should use some trick for it. So performing fast calculation in such a competitive world is a big challenge.

So for the above problem, we require a calculator with which the below implementations can be done easily:

1) Operations like addition, subtraction, multiplication, Division. Other than this it should be able to find sine, cosine, tangent of any value, logarithm and factorials (only one number to be entered).
2) It should also find power of any number to any power (e.g. POW(2,3)==8). And NTHROOT of any number (e.g. NTHROOT(81,4)=3). It should also be capable of finding HCF and LCM of any two given numbers.
3) It should also tell whether the number is PRIME or whether the pair is COPRIME and also find COMBINATION and PERMUTATION of any pair of numbers.

METHODOLOGY:

Here we will be using AWT and Object class properties along with some basic Math class functions (for log and sin, tan, cos functions) in our calculator. They are defined below as:

AWT: Java AWT(Abstract Window Toolkit) is an API to develop GUI or Windows based applications in java. This package has components and classes such as TextField, TextArea, checkbox, List, Label and many other classes to create an interactive GUI application. There is another package known as "java.awt.event" which has many classes and Listener interface for providing dynamic event handling capacity to our static view of calculator.

In our calculator we will be using Action Listener which listens to the button pressed and perform calculations to show its dynamic behavior. Also we will using basic containers of awt such as Frame and Window to display our Buttons, TextFields, Labels and other items.

We are also using various methods involved in these classes as "setText();" "getText();" also we are overriding "actionPerformed" method of "ActionListener" class in the below code for event handling.

OBJECT CLASS : It is the parent class of all the classes in java by default. In other words it is the topmost class. We will be using one of its method i.e (toString()) for conveting the numeric values to String format to display them in TextField.

MATHS CLASS : We are also using some methods of math class for basic mathematical operations in calculator such as "sine"," tangent", "cosine", "log" etc.

So basically we are making a class named "**calculator**", in whose constructor we are declaring a Frame containing **List** of all the operations, our calculator can perform. we are basically involving 18 basic operations : + , - , * , / , Fact(Factorial), FIB(the respective Fibonacci term), PRIME (tells whether the number is prime or not), COMBINATION(named as COMBIN), PERMUTATION(named as PERMU), sin, cos, tan, log, NTHROOOT(nth root of any number), COPRIME, HCF, LCM, POW.

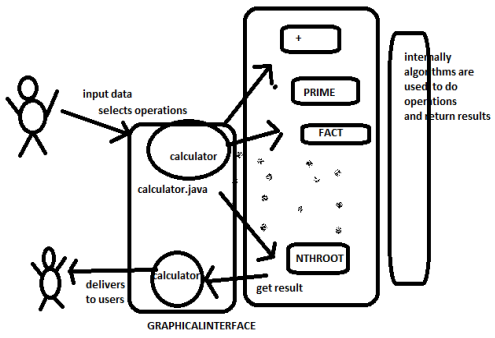
In this we will create two fields to get numeric input and one list of above 18 operations to be selected to perform respective operation and the result will be printed in result field. We will get input as string which can be converted to double data type using "parseDouble()" method. To get the input we will use "getText()", method of event handler class. Then we will also get the name of operation selected by the user from list using "getItem(getIndexSelected())" method.

After than when the calculate button will be pressed the event handling will be triggered and operation will be performed by switch operation to switch to that operation's algorithm which is selected by user, and respective operation is done by the algorithm, and the output is printed to result TextField using "setText()" method.

The algorithm are all basic one to calculate the fibonaaci number or to find nthroot as we use in normal coding. So basically we are here combining all basic mathematical operations to make our calculator work. In the below implementation we can see different cases for different algorithms such as for calculating sine, we will use "Math.sin()", method , for finding factorial we will use while loop for multiplication, for checking whether the number is prime or not, we will use "Sieve of Erastothenes" algorithm, for finding HCF, LCM we will use the idea that (HCF (a,b) = (a*b)/(LCM(a,b))) and so on .

We also describes a basic dispose() method at last to make **close** operation successful. Also we create an instance of calculator class in main method of Java. The below diagram is the **use case diagram** showing how calculator works. Here how user enters the data and how calculator switches to the case and perform calculations and gives the results to the user as shown.

Since this calculator performs a single number operations also so some basic instructions, following which we can get correct output, is given below.



USE CASE

1) For FIB, FACT,LOG, SIN,COS,TAN,PRIME, just enter only number in "firstnumber " TextField and calculate the result.

2) POW(2,3)=8, HCF(2,3)=1, and same cases for COMBIN, LCM , PERMU (permutation) use same standard (first,second)=result.

3) For NTHROOT(81,4)=3, follow same, above rule. It's algorritm is given in code.

Hence we created a calculator following these rules from the code shown below:

We are declaring our code with a public class named "calculator". (File named as "calculator.java").

IT IS WITTEN IN THIS WAY (WITHOUT INDENTATIONS) JUST TO AVOID WASTEAGE OF SPACE.

```
import java.lang.*;
import java.awt.*;
import java.awt.event.*;
public class calculator extends WindowAdapter {
      Button b1;      TextField t1,t2,t3;               Frame f;
    List l =new List(18,false);
    Label label2=new Label("See instruction given in code for single number operations");
    Label l1=new Label("CALCULATOR");          Label label5=new Label("firstnumber");
Label label3 =new Label("secondnumber");               Label label4=new Label("result");
    calculator(){
             b1=new Button("Calculate");        t1=new TextField();       t2=new TextField();
             t3 =new TextField();      f=new Frame("CALCULATOR");

             f.setVisible(true);       f.setLayout(null);        f.setSize(500,500);
             f.addWindowListener(this);         l.setBounds(10,200,70,300);

             l.add("+");      l.add("-");      l.add("*");      l.add("/");
             l.add("FACT");   l.add("FIB");    l.add("POW");    l.add("NTHROOT");
             l.add("PRIME");  l.add("SIN");    l.add("COS");    l.add("TAN");
             l.add("HCF");    l.add("LCM");    l.add("COPRM");  l.add("COMBIN");
             l.add("PERMU");  l.add("LOG");

             l1.setAlignment(Label.CENTER);     l1.setSize(500,100);
             label5.setBounds(300,100,70,20 );          label3.setBounds(300,150,80,20);
             label4.setBounds(300,250,70,20);           label2.setBounds(50,200,450,50);
```

```
t1.setBounds(100,100,150,20);              t2.setBounds(100,150,150,20);

t3.setBounds(100,250,150,20);       b1.setBounds(200,400,50,50);


f.add(l);         f.add(l1);        f.add(label3);            f.add(label4);

f.add(label5);    f.add(label2);    f.add(t1);        f.add(t2);

f.add(t3);        f.add(b1);

b1.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e)

                {

                        String s1=t1.getText();     Double x=Double.parseDouble(s1);

                        String s3=l.getItem(l.getSelectedIndex());

                switch(s3)

                {


case "+":         {


                String s2=t2.getText();            Double y=Double.parseDouble(s2);

                x=x+y;      s1=Double.toString(x);      t3.setText(s1);      break;

                        }

case "-":   {

                String s2=t2.getText();   Double y=Double.parseDouble(s2);

                x=x-y;   s1=Double.toString(x);    t3.setText(s1);   break;

                }

case "*":    {

                String s2=t2.getText();   Double y=Double.parseDouble(s2);

                x=x*y;   s1=Double.toString(x);    t3.setText(s1);  break;

                        }

    case "/":   {
```

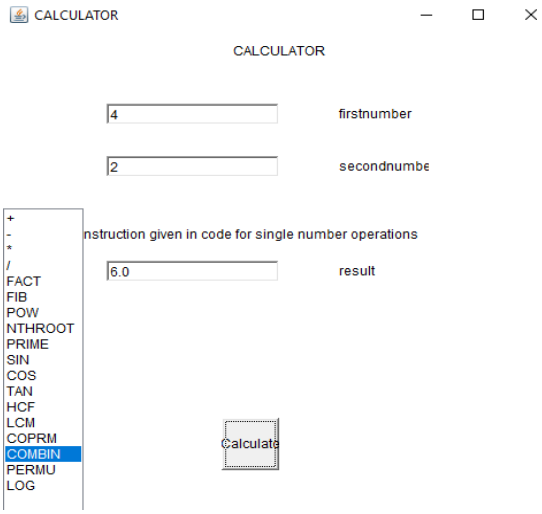
```
                                String s2=t2.getText();   Double y=Double.parseDouble(s2);
                                x=x/y; s1=Double.toString(x);      t3.setText(s1);   break;
                                        }
    case "FACT": {
                                int ans=1;       while(x>1){ans*=x;--x;}   s1=Double.toString(ans);
                                t3.setText(s1);   break;
                        }
case "FIB": {
                                int a=0,b=1;     int y=0;         for(int i=1;i<x;++i){y=a+b;b=a;a=y;}
                                s1=Integer.toString(y);   t3.setText(s1);   break;
                        }
    case "POW": {
                                String s2=t2.getText();   int y =Integer.parseInt(s2);
                                Double r=1.0;     while(y>0){r*=x;--y;}     s1=Double.toString(r);
                                t3.setText(s1);           break;
                        }
case "PRIME": {
                                int ok=0;
                                if(x!=1||x!=2)
                                for(int i=2;i<x;++i){if(x%i==0){ok=1;break;}}
                                if(x==2||ok==0) s1="YES";
                                else
                                s1="NO";
                                if(x==1)
                                s1="NO";
                                t3.setText(s1);   break;
                                }
case "NTHROOT":{
```

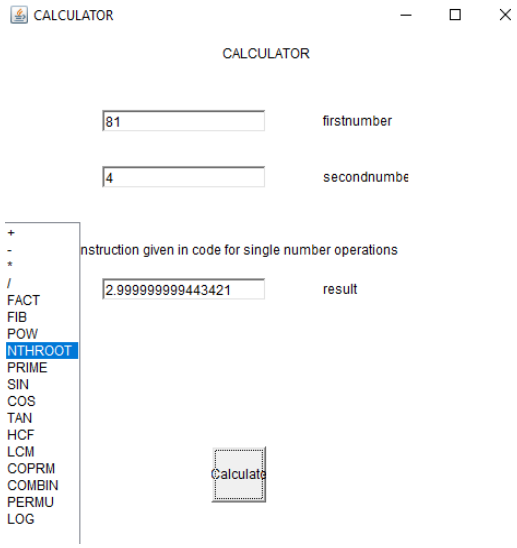
```
                Double y=  Double.parseDouble(t2.getText());  x=Math.log(x);
                  x/=y;    x=Math.pow(2.718281828,x);
            s1=Double.toString(x);    t3.setText(s1);   break;}

    case "LOG":{            x=Math.log(x);    s1=Double.toString(x);
                  t3.setText(s1);   break;
            }
case "SIN":{                x=Math.sin(x);    s1=Double.toString(x);    t3.setText(s1);
                  break;                }
case "COS":{                x=Math.cos(x);    s1=Double.toString(x);    t3.setText(s1);
            break;                }
case "TAN":{                x=Math.tan(x);    s1=Double.toString(x);    t3.setText(s1);
            break;   }
case "COPRM":{
                  Double a=x,b=Double.parseDouble(t2.getText());      Double y;
                  while(true)
                  {       if (a == 0){y=b;break;}   if(b == 0){y=a;break;}
                        if (a == b){y= a;break;}  if (a > b){a=a-b;}
                        b=b-a;            }
                  if(y==1)
                        s1="YES";
                  else
                        s1="NO";
                  t3.setText(s1);                   break;           }

    case "HCF":{            Double a=x,b=Double.parseDouble(t2.getText());      Double y;
                  while(true)
            {if (a == 0){y=b;break;}  if(b == 0){y=a;break;}
            if (a == b){y= a;break;}  if (a > b){a=a-b;} b=b-a;          }
```

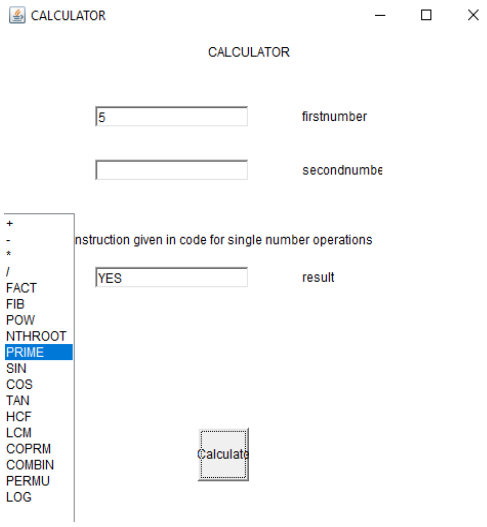
```
                    s1=Double.toString(y);    t3.setText(s1);   break;
            }
case "LCM":{
                            Double a =x,b=Double.parseDouble(t2.getText());     Double y;
                            while(true){if (a == 0){y=b;break;}        if(b == 0){y=a;break;}
                                        if (a == b){y= a;break;}  if (a > b)a=a-b;
                                        b=b-a;   }
                            y=(x*Double.parseDouble(t2.getText()))/y;   s1=Double.toString(y);
                            t3.setText(s1);   break;   }

case "COMBIN":{
                            Double z1,z2;z1=x;        Double ans=1.0;   while(x>1){ans*=x;--x;}
                            Double ans2=1.0;  Double y= Double.parseDouble(t2.getText());
                            z2=y; Double ans3=1.0;    while(y>1){ans2*=y;--y;} z1=z1-z2;
                            while(z1>1){ans3*=z1;--z1;}        y=ans/(ans3*ans2);
                            s1=Double.toString(y);    t3.setText(s1);   break;   }
    case "PERMU":{
                            Double ans=1.0;   while(x>1){ans*=x;--x;}   Double ans2=1.0;
                            Double y= Double.parseDouble(t2.getText()); while(y>1){ans2*=y;--y;}
                            y=ans/ans2;       s1=Double.toString(y);    t3.setText(s1);   break;
                            }
                    }
    }
});
}
public void windowClosing(WindowEvent e){       f.dispose();      }
public static void main(String[] args){         new calculator();}
} //END.
```

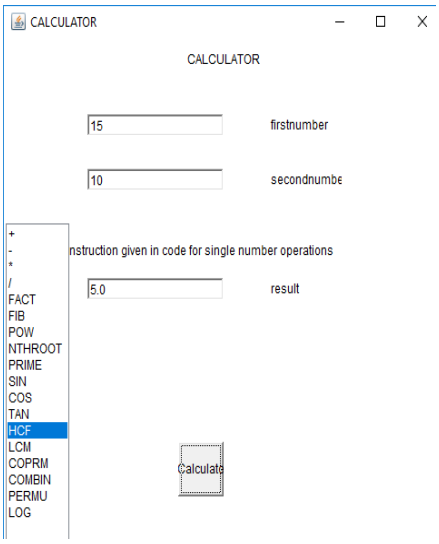
RESULTS:

Below some images are shown how calculator can perform operations and the way it will be displayed to users graphically.









The repository having real code (with comments, with instructions and proper indentations) can be found on my GITHUB account under name paper.java, follow this link to access it:

https://harshkumarchoudary.github.io/Java-Calculator/

CONCLUSION:

Hence we can see that our calculator is able to perform operations like finding HCF, LCM, checking primarility of a given number, finding NTHROOT, finding fibonacci term, factorials, checking whether the pair of numbers are coprime, easily. Hence it reduced the task of remembering the tricks to perform above operations and also long calculations can be performed easily (since DOUBLE data type is used).

We also found that using AWT for creating a graphical interface, can help user to understand the environment better than using shell interface. we can to learn several method of different classes of java packages. We found how an interrelationship among all components and methods and events will finally result in such a smart device.

We are also able to learn several new mathematical algorithm of different type to perform operations. This improves our coding skills. Use of loops and statements in Java is also learnt by us form this project.

BIBLIOGRAPHY:

- Geeks for geeks : java programming language and tutorials: https://www.geeksforgeeks.org/java/
- Java point AWT tutorial : https://www.javatpoint.com/java-awt
- Docs.oracle.com for java full documentation: https://docs.oracle.com/javase/7/docs/api/java/awt/package-summary.html
- And many other references from different books