Server Side
Pie graph

```
package ServerSide;

import java.awt.*;
import java.util.Map;
import java.util.TreeMap;
import java.awt.event.*;
import javax.swing.*;
import java.util.*;
import java.lang.*;
import javax.swing.JPanel;
import java.sql.*;

class Piegraph extends JFrame {
        PiegraphBuild pie1;
           public Piegraph() {
                       buildWindow();
                   }

             protected void buildWindow() {

               Container con=getContentPane();
                     pie1=new PiegraphBuild();

                                addWindowListener(new WindowAdapter()
         {
             public void windowClosing(WindowEvent e)
             {
                 System.exit(0);
              }
          });

          con.add(pie1);

                      setTitle("PIE-CHART");
         setSize(800,600);
         setVisible(true);
       }

    public static void main(String []args) {

      new Piegraph();

          }
  }

class PiegraphBuild extends JPanel
{

           Connection con;
              Statement stmt;
```

```
            ResultSet rs;
            String sql,sql1;
                  private Map<String,Integer> pieData;

                  PiegraphBuild()
        {

          pieData = new TreeMap<String,Integer>();
          getConnection();
          initializeData();
          setBackground(Color.white);

 }

        //////////////////////getdata//////////////////////

protected void getConnection()
    {


            System.out.println("DataBase is connected");
            try
             {
                                Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                con=DriverManager.getConnection("jdbc:odbc:voting");
             }

                  catch(Exception e)
           {
                System.out.println("JDBC:ODBC driver failed to load.");
                return;
           }
          try
          {

          stmt = con.createStatement();
          }

          catch(Exception e)
          {
                   e.printStackTrace();
                return;
          }



    }



//////////////////////////////////////
```

```
//*************** To Initialize the Object ************************************//
        protected void initializeData() {


            try {


            sql="select * from Result";
             stmt = con.createStatement();
                    rs=stmt.executeQuery(sql);
                    while(rs.next())
                    {

                            String s1=rs.getString(2);
                            int n=Integer.parseInt(rs.getString(4));
                            pieData.put(s1,n);


                    }
                    }
                    catch(SQLException e)
                    {
                            System.out.println("fgsdgsg");

                    }



  }


            int Tvotes ;
            double percentage;
            int Vote1,Vote,Iangle=90,n=0;
            int color1=25 ,color2=50,color3=75,X_axixs=80;

        public void paintComponent(Graphics g)
            {
                    super.paintComponents(g);
                    Set<String> keySet = pieData.keySet();
            Iterator<String> keyIterator = keySet.iterator();
            while(keyIterator.hasNext())                                    //To Get the Total
Number Of Votes
            {
            String partyName = keyIterator.next();
                    pieData.get(partyName);
                    Vote1 = pieData.get(partyName);
                    Tvotes=Tvotes+Vote1;
      //Formula For Total Number of Votes
            }
```

```
            Set<String> keySet1 = pieData.keySet();
            Iterator<String> keyIterator1 = keySet1.iterator();
            while(keyIterator1.hasNext())                                   //To
Calculate the Degrees & Draw the Pie Diagram.
            {
                double Tvotes2=(double)Tvotes;
                String partyName1 = keyIterator1.next();
                pieData.get(partyName1);                                    //For
Accessing the Partyname from Mapped Set
                System.out.println(partyName1);

                Vote = pieData.get(partyName1);                             //For
Accessing the Votes of Parties from Mapped Set
                System.out.println("Vote \t" +Vote);
                System.out.println("Total Votes \t" +Tvotes2);

                percentage=(Vote/Tvotes2)*100;                              //To
Calculate the Percentage of Votes.
                System.out.println("Percentage \t" +percentage);

                double deg1=(percentage/100)*360;                           //To Calculate the
Degrees from Percentage of Votes.
              int degree1=(int) deg1;
                System.out.println("DEGREE \t\t"+degree1);


                n=n+1;
                if(n==1)
                        g.setColor(Color.BLACK);
                if(n==2)
                        g.setColor(Color.ORANGE);
                if(n==3)
                        g.setColor(Color.BLUE);
                if(n==4)
                        g.setColor(Color.DARK_GRAY);
                if(n==5)
                        g.setColor(Color.RED);
                if(n==6)
                        g.setColor(Color.PINK);
                        if(n==7)
                        g.setColor(Color.GREEN);

                  if(n>8)
                 {
                Color color4=new Color(color1,color2,color3);
                          g.setColor(color4);
                         }
                         g.drawString(partyName1,X_axixs,500);              //For
Printing the Party Names of Respective Pie-Diagram.
                     X_axixs=X_axixs+80;
```

```
                    g.fillArc(300,200,201,201,Iangle,degree1);          //For Drawing the
Pie Diagram of Respective Political Parties.
                    Iangle=Iangle+degree1;

               color1=color1+25;
          //For Filling the Pie-Diagram with Different Colors Shades.
               color2=color2+25;
               color3=color3+25;

                System.out.println("\n\n");
              }

               String label="PIE-CHART";
               Font font=new Font("ArialBlack",Font.ITALIC,20);
               g.drawRect(30,50,720,460);
                    g.setColor(Color.black);
                    g.setFont(font);
                    g.drawString(label,360,150);
           }


          }
```

Candidates

```
if(ae.getSource()==storeEditBtn[1]) {

                    SetEditable(1);
                     candNameArray[1] =canName[1].getText();
                     candDscArray[1] = canDsc[1].getText();
                      System.out.println(candNameArray[1]);

               }
               if(ae.getSource()==storeEditBtn[2]) {

                    SetEditable(2);
                     candNameArray[2] =canName[2].getText();
                     candDscArray[2] = canDsc[2].getText();
                      System.out.println(candNameArray[2]);
               }
               if(ae.getSource()==storeEditBtn[3]) {

                    SetEditable(3);
                     candNameArray[3] =canName[3].getText();
                     candDscArray[3] = canDsc[3].getText();
                      System.out.println(candNameArray[3]);
               }
               if(ae.getSource()==storeEditBtn[4]) {

                    SetEditable(4);
                     candNameArray[4] =canName[4].getText();
                     candDscArray[4] = canDsc[4].getText();
                      System.out.println(candNameArray[4]);
               }
```

```
            if(ae.getSource()==storeEditBtn[5]) {

                    SetEditable(5);
                     candNameArray[5] =canName[5].getText();
                     candDscArray[5] = canDsc[5].getText();
                      System.out.println(candNameArray[5]);
            }
            if(ae.getSource()==storeEditBtn[6]) {

                    SetEditable(6);
                     candNameArray[6] =canName[6].getText();
                     candDscArray[6] = canDsc[6].getText();
                      System.out.println(candNameArray[6]);
            }
            if(ae.getSource()==storeEditBtn[7]) {

                    SetEditable(7);
                     candNameArray[7] =canName[7].getText();
                     candDscArray[7] = canDsc[7].getText();
                      System.out.println(candNameArray[7]);
            }
            if(ae.getSource()==storeEditBtn[8]) {

                    SetEditable(8);
                     candNameArray[8] =canName[8].getText();
                     candDscArray[8] = canDsc[8].getText();
                      System.out.println(candNameArray[8]);
            }
            if(ae.getSource()==storeEditBtn[9]) {

                    SetEditable(9);
                     candNameArray[9] =canName[9].getText();
                     candDscArray[9] = canDsc[9].getText();
                      System.out.println(candNameArray[9]);
            }


  } /* End of Action Performed*/

}  /* End of main Class*/
```

Client Side

Login

```
public class Login extends JFrame implements ActionListener
{
      JLabel name, pass;
      JTextField nameText;
      JPasswordField passText;
      JButton login, end;
      static int attempt=0;

      public Login()
      {
```

```
        name = new JLabel("Name:",JLabel.RIGHT);
        pass = new JLabel("Password:",JLabel.RIGHT);

        nameText = new JTextField(20);
        passText = new JPasswordField(20);

        login = new JButton("Login");
        end = new JButton("End");

        login.addActionListener(this);
        end.addActionListener(this);

        Container c = getContentPane();
        c.setLayout(new GridLayout(3,2));
        c.add(name);
        c.add(nameText);
        c.add(pass);
        c.add(passText);
        c.add(login);
        c.add(end);

        setTitle("Login Check");
        setSize(400,200);
        setLocation((800-400)/2,(600-165)/2);
        pack();
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae)
    {
        JButton btn = (JButton)ae.getSource();
        if(btn == end)
        {
            System.exit(0);
        }
        if(btn == login)
        {
            try
            {
                String user = nameText.getText();
                String pass = new String(passText.getPassword());
                if(user.compareTo(pass)==0)
                {
                    JOptionPane.showMessageDialog(null,"Login
Successful","Login",JOptionPane.INFORMATION_MESSAGE);
                    System.exit(0);
                }
                else
                {
                    throw new InvalidPasswordException();
                }
```

```
                }
                catch(Exception e)
                {
                        attempt++;
                        JOptionPane.showMessageDialog(null,"Login
Failed","Login",JOptionPane.ERROR_MESSAGE);
                        nameText.setText("");
                        passText.setText("");
                        nameText.requestFocus();
                        if(attempt == 3)
                        {
                                JOptionPane.showMessageDialog(null,"3                    Attempts
Over","Login",JOptionPane.ERROR_MESSAGE);
                                System.exit(0);
                        }
                }
        }
    }

    public static void main(String args[])
    {
        new Login();
    }
}
```

Register

```
public class Register extends JFrame implements ActionListener
 {
   private JMenu mnuHelp;
        private JMenu mnuFiles;
        private JMenuBar menubar;
        private JMenuItem mnuForm;
        private JMenuItem mnuAbt;
        private JMenuItem mnuExit;
        private JMenuItem Help;
        private JButton button[]=new JButton[20];
        private JTextField text[]=new JTextField[20];
        private JComboBox nationCombo = new JComboBox();
   private JLabel label[]=new JLabel[20];
        private int i=0,j=0,k=0;


        public Register()
        {


                super("Voting System - Voter Registration");
                Container con = getContentPane();
                con.setLayout(null);


       menubar=new JMenuBar();
       setJMenuBar(menubar);
```

```
                mnuHelp = new JMenu("Help");
                mnuHelp.setMnemonic('H');
                mnuFiles = new JMenu("Files");
                mnuFiles.setMnemonic('F');


                mnuForm=new JMenuItem("New Registration Form");
                mnuForm.setMnemonic('R');
                mnuFiles.add(mnuForm);
                mnuForm.addActionListener(this);

                mnuAbt=new JMenuItem("About");
                mnuAbt.setMnemonic('A');
                mnuFiles.add(mnuAbt);
                mnuAbt.addActionListener(this);

                mnuExit=new JMenuItem("Exit");
                mnuExit.setMnemonic('E');
            mnuFiles.add(mnuExit);
            mnuExit.addActionListener(this);

            Help=new JMenuItem(" About help F1");
        //Help.addActionListener(this);
            Help.setMnemonic('F');
            mnuHelp.add(Help);

            menubar.add(mnuFiles);
                menubar.add(mnuHelp);

          label[i]=new JLabel("VOTER    REGISTRATION    PROCESS");
                label[i].setFont(new Font("TimesNewRoman",Font.BOLD,20));
                label[i].setBounds(220,20,650,30);
                con.add(label[i]);
                i=i+1;

                label[i]=new JLabel("# A User has to Register Himself before He is ellegibe for
Voting.");
                label[i].setFont(new Font("MonoSpaced",Font.BOLD,15));
                label[i].setBounds(45,63,750,30);
                con.add(label[i]);
                i=i+1;

                label[i]=new JLabel("# A User Has to fill following Entries in order to Register
Himself as Voter. ");
                label[i].setFont(new Font("MonoSpaced",Font.BOLD,15));
                label[i].setBounds(45,88,750,30);
                con.add(label[i]);
                i=i+1;

                label[i]=new JLabel("# A User Has to fill Correct Entries in  Corresponding
TextFields in order to Register Himself as Voter. ");
```

```
        label[i].setFont(new Font("MonoSpaced",Font.BOLD,12));
        label[i].setBounds(45,120,750,30);
        con.add(label[i]);
        i=i+1;


        label[i]=new JLabel("1.Voter Name:");
        label[i].setFont(new Font("Arial",Font.ITALIC,15));
        label[i].setBounds(15,165,750,30);
        con.add(label[i]);
        i=i+1;

        text[j]=new JTextField(20);
        text[j].setBounds(160, 165, 210, 20);//(z,y,w,h)
text[j].setFont(new Font("Arial",Font.BOLD,13));
con.add(text[j]);


        text[j].addKeyListener(new KeyAdapter() {
                    public void keyTyped(KeyEvent e) {
                      char c = e.getKeyChar();

                      if (!((c >= 'A' ||c >= 'a' ) && (c <='Z'||c <='z' ) ||
                        (c == KeyEvent.VK_BACK_SPACE) ||
                         (c == KeyEvent.VK_DELETE))) {
                        getToolkit().beep();
                        e.consume();
                      }
                    }
                  });
                        j=j+1;
label[i]=new JLabel("3.VoterID:");
        label[i].setFont(new Font("Arial",Font.ITALIC,15));
        label[i].setBounds(15,200,750,30);
        con.add(label[i]);
        i=i+1;

text[j]=new JTextField(20);
        text[j].setBounds(160, 200, 210, 20);//(z,y,w,h)
        text[j].setFont(new Font("Arial",Font.BOLD,13));
con.add(text[j]);

        text[j].addKeyListener(new KeyAdapter() {
                    public void keyTyped(KeyEvent e) {
                      char c = e.getKeyChar();

                      if (!((c >= 'A' ||c >= 'a' ||c >= '0' ||c >= '9' ) && (c <='Z'||c <='z' ) ||
                        (c == KeyEvent.VK_BACK_SPACE) ||
                         (c == KeyEvent.VK_DELETE))) {
                        getToolkit().beep();
                        e.consume();
                      }
```

```
                  }
                });


j=j+1;

label[i]=new JLabel("5.Password:");
      label[i].setFont(new Font("Arial",Font.ITALIC,15));
      label[i].setBounds(15,250,750,30);
      con.add(label[i]);
      i=i+1;

text[j]=new JPasswordField(20);
      text[j].setBounds(160, 250, 210, 20);//(z,y,w,h)
      text[j].setFont(new Font("Arial",Font.BOLD,13));
con.add(text[j]);
j=j+1;

label[i]=new JLabel("6.Voter Phone No.:");
      label[i].setFont(new Font("Arial",Font.ITALIC,15));
      label[i].setBounds(15,300,750,30);
      con.add(label[i]);
      i=i+1;

text[j]=new JTextField(20);
      text[j].setBounds(160, 300, 210, 20);//(z,y,w,h)
      text[j].setFont(new Font("Arial",Font.BOLD,13));
con.add(text[j]);

text[j].addKeyListener(new KeyAdapter() {
                    public void keyTyped(KeyEvent e) {
                      char c = e.getKeyChar();

                      if(!((c >= '0' && c <='9' )||
                        (c == KeyEvent.VK_BACK_SPACE) ||
                         (c == KeyEvent.VK_DELETE))) {
                        getToolkit().beep();
                        e.consume();
                      }
                    }
                  });




j=j+1;

label[i]=new JLabel("7.Voter Adderess:");
      label[i].setFont(new Font("Arial",Font.ITALIC,15));
      label[i].setBounds(15,350,750,30);
      con.add(label[i]);
```

```
        i=i+1;

text[j]=new JTextField(20);
        text[j].setBounds(160, 350, 210, 20);//(z,y,w,h)
        text[j].setFont(new Font("Arial",Font.BOLD,13));
con.add(text[j]);
text[j].addKeyListener(new KeyAdapter() {
                    public void keyTyped(KeyEvent e) {
                      char c = e.getKeyChar();

                      if (!((c >= 'A' ||c >= 'a' ||c >= '0' ||c >= '9' ) && (c <='Z'||c <='z' ) ||
                        (c == KeyEvent.VK_BACK_SPACE) ||
                         (c == KeyEvent.VK_DELETE))) {
                        getToolkit().beep();
                        e.consume();
                      }
                    }
                  });
j=j+1;

label[i]=new JLabel("8.Date of Birth:");
        label[i].setFont(new Font("Arial",Font.ITALIC,15));
        label[i].setBounds(15,400,750,30);
        con.add(label[i]);
        i=i+1;

        text[j]=new JTextField(20);
        text[j].setBounds(160, 400, 210, 20);//(z,y,w,h)
        text[j].setFont(new Font("Arial",Font.BOLD,13));
con.add(text[j]);

text[j].addKeyListener(new KeyAdapter() {
                    public void keyTyped(KeyEvent e) {
                      char c = e.getKeyChar();

                      if (!((c >= 'A' ||c >= 'a' ||c >= '0' ||c >= '9' ) && (c <='Z'||c <='z' )||
                        (c == KeyEvent.VK_BACK_SPACE) ||
                         (c == KeyEvent.VK_DELETE))) {
                        getToolkit().beep();
                        e.consume();
                      }
                    }
                  });
j=j+1;



label[i]=new JLabel("9.Nationality:");
        label[i].setFont(new Font("Arial",Font.ITALIC,15));
        label[i].setBounds(15,450,750,30);
        con.add(label[i]);
        i=i+1;
```

```
nationCombo.addItem("India");
nationCombo.addItem("USA");
nationCombo.addItem("England");
nationCombo.addItem("Pakistan");
nationCombo.addItem("Russia");
nationCombo.addItem("Japan");
nationCombo.addItem("China");
nationCombo.addItem("Brazil");
nationCombo.addItem("Germany");
nationCombo.setBounds(160, 450, 210, 20);
con.add(nationCombo);

j=j+1;

label[i]=new JLabel("2.Voter Middle Name:");
      label[i].setFont(new Font("Arial",Font.ITALIC,15));
      label[i].setBounds(400,165,750,30);
      con.add(label[i]);
      i=i+1;

text[j]=new JTextField(20);
      text[j].setBounds(560, 165, 210, 20);//(z,y,w,h)
text[j].setFont(new Font("Arial",Font.BOLD,13));
con.add(text[j]);


      text[j].addKeyListener(new KeyAdapter() {
                  public void keyTyped(KeyEvent e) {
                    char c = e.getKeyChar();

                    if (!((c >= 'A' ||c >= 'a' ) && (c <='Z'||c <='z' ) ||
                      (c == KeyEvent.VK_BACK_SPACE) ||
                       (c == KeyEvent.VK_DELETE))) {
                      getToolkit().beep();
                      e.consume();
                    }
                  }
                });
j=j+1;

label[i]=new JLabel("4.Voter Last Name:");
      label[i].setFont(new Font("Arial",Font.ITALIC,15));
      label[i].setBounds(400,200,750,30);
      con.add(label[i]);
      i=i+1;

text[j]=new JTextField(20);
      text[j].setBounds(560, 200, 210, 20);//(z,y,w,h)
      text[j].setFont(new Font("Arial",Font.BOLD,13));
con.add(text[j]);
```

```
            text[j].addKeyListener(new KeyAdapter() {
                        public void keyTyped(KeyEvent e) {
                          char c = e.getKeyChar();

                          if (!((c >= 'A' ||c >= 'a' ) && (c <='Z'||c <='z' ) ||
                            (c == KeyEvent.VK_BACK_SPACE) ||
                             (c == KeyEvent.VK_DELETE))) {
                            getToolkit().beep();
                            e.consume();
                          }
                        }
                       });
      j=j+1;

      button[i]=new JButton("Register");
      button[i].setBounds(560,480,100,30);
      con.add(button[i]);

      System.out.println("I: "+i);
            System.out.println("J: "+j);
            System.out.println("K: "+k);



            addWindowListener(new WindowAdapter()
{
      public void windowClosing(WindowEvent e)
      {
          System.exit(0);
      }
});


setSize(800,600);
setVisible(true);
}

public void actionPerformed(ActionEvent ae)
{




   if(ae.getSource()==mnuForm)
   {
     for(k=0;k<20;k++)
     {

            text[k].setText("");
            }
```

```
        }
         if(ae.getSource()==mnuExit)
         {

                                int ai= JOptionPane.showConfirmDialog(null, "Are you sure you want to exit?", " Client Model", JOptionPane.YES_NO_OPTION);
                                if(ai==0)
                                System.exit(0);
         }
                        if(ae.getSource()==mnuAbt)
                        {
                                String msg ;
                                msg = "#  This is a Voter Registration Panel \n";
                                msg += "# This pannel can be Used by User to Register Himself as Voter.\n";
                                msg += "#  The Voter id,phone no, should be a Integer value\n";
                                msg += "#  The Birth Date of a Voter should be in format of 26Sep1987\n";
                                JOptionPane.showMessageDialog(null, msg, " Client", JOptionPane.INFORMATION_MESSAGE );
                        }


       }

         public static void main(String args[])
      {
           new Register();
      }


         private class StrictInputVerifier extends InputVerifier {
            private String validString;

            public StrictInputVerifier(String validString) {
               this.validString = validString;
            }

            public boolean verify(JComponent input) {
                  System.out.println("FD");
               JTextField textField = (JTextField) input;
               boolean retCode = false;
               try {
                  Integer.parseInt(textField.getText());
                  retCode = true;
               }
               catch(Exception e) {

               }
               return retCode;

               }
```

Result Module
Bar-graph

```
class Bargraph extends JFrame {

                BuildBarGraph  bargraph;
            public Bargraph() {

                 buildWindow();
                }
          protected void buildWindow() {

            Container con=getContentPane();
                bargraph=new BuildBarGraph();

             addWindowListener(new WindowAdapter()
        {
           public void windowClosing(WindowEvent e)
            {
                System.exit(0);
            }
        });
      con.add(bargraph);
                setTitle("BAR-CHART");
      setSize(800,600);
      setVisible(true);
   }
   public static void main(String []args) {
                new Bargraph();
        }

}

        class BuildBarGraph extends JPanel
        {
        private Map<String,Integer> pieData;

                 BuildBarGraph()
                {
                  pieData = new TreeMap<String,Integer>();
             initializeData();
             setBackground(Color.white);
      }


                protected void initializeData() {
                pieData.put("A",270);
                pieData.put("B",150);
                pieData.put("C", 245);
                pieData.put("D",200);
                pieData.put("Others", 110);
        }
```

```
        int Tvotes ;
        double percentage;
        int Vote1,Vote,Iangle =90,n = 0;
        int color1 =20,color2 =25,color3 = 50,X_axixs = 100,Y_axixs = 300;
        public void paintComponent(Graphics g)
        {

                super.paintComponents(g);
                Set<String> keySet = pieData.keySet();
        Iterator<String> keyIterator = keySet.iterator();


        while(keyIterator.hasNext())
        {
        String partyName = keyIterator.next();
                pieData.get(partyName);
                Vote1 = pieData.get(partyName);

                Tvotes=Tvotes+Vote1;
        }


Set<String> keySet1 = pieData.keySet();
Iterator<String> keyIterator1 = keySet1.iterator();


while(keyIterator1.hasNext())
{
        double Tvotes2=(double)Tvotes;
        String partyName1 = keyIterator1.next();


        pieData.get(partyName1);
        System.out.println(partyName1);


        Vote = pieData.get(partyName1);
        System.out.println("Vote \t" +Vote);
        System.out.println("Total Votes \t" +Tvotes2);


        percentage=(Vote/Tvotes2)*100;
        System.out.println("Percentage \t" +percentage);


        double deg1=(percentage/100)*360;
    int degree1=(int) deg1;
        System.out.println("DEGREE \t\t"+degree1);
```

```
                n=n+1;
            if(n==1)
                g.setColor(Color.BLACK);
                if(n==2)
                g.setColor(Color.BLUE);
                if(n==3)
                g.setColor(Color.CYAN);
                if(n==4)
                g.setColor(Color.DARK_GRAY);
                if(n==5)
                g.setColor(Color.RED);
                if(n==6)
                g.setColor(Color.PINK);
                if(n==6)
                g.setColor(Color.PINK);
                if(n==7)
                g.setColor(Color.ORANGE);

              if(n>7)
              {
            Color color4=new Color(color1,color2,color3);
                g.setColor(color4);
                }



                g.drawString(partyName1,X_axixs,520);


                g.fillRect(X_axixs,500-Vote,50,Vote);
              X_axixs=X_axixs+80;

          color1=color1+25;
          color2=color2+25;
          color3=color3+25;

          System.out.println("\n\n");
        }

        String label="BAR-CHART";
        Font font=new Font("ArialBlack",Font.ITALIC,20);
        g.drawRect(30,60,700,500);
             g.setColor(Color.black);
             g.setFont(font);
             g.drawString(label,320,100);

    }/* End of paint Component Method */
 }/* End of BuildBarGraph classs*/
```

Pie Graph

```
class Piegraph extends JFrame {
```

```
        PiegraphBuild pie1;
          public Piegraph() {

                    buildWindow();
                }


           protected void buildWindow() {

             Container con=getContentPane();
                  pie1=new PiegraphBuild();

                          addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent e)
            {
                System.exit(0);
             }
        });

        con.add(pie1);

                  setTitle("PIE-CHART");
        setSize(800,600);
        setVisible(true);
       }

    public static void main(String []args) {

     new Piegraph();

         }
  }

class PiegraphBuild extends JPanel
{
          private Map<String,Integer> pieData;

          PiegraphBuild()
          {
           pieData = new TreeMap<String,Integer>();
           initializeData();
           setBackground(Color.white);

    }

            protected void initializeData() {
                  pieData.put("A",1170);
                  pieData.put("B",170);
                  pieData.put("C", 170);
                  pieData.put("D",170);
                  pieData.put("Others", 170);
```

```
}


        int Tvotes ;
        double percentage;
        int Vote1,Vote,Iangle=90,n=0;
        int color1=25 ,color2=50,color3=75,X_axixs=80;

    public void paintComponent(Graphics g)
        {
                super.paintComponents(g);
                Set<String> keySet = pieData.keySet();
        Iterator<String> keyIterator = keySet.iterator();

        while(keyIterator.hasNext())
        {
        String partyName = keyIterator.next();
                pieData.get(partyName);
                Vote1 = pieData.get(partyName);


                Tvotes=Tvotes+Vote1;
        }


     Set<String> keySet1 = pieData.keySet();
     Iterator<String> keyIterator1 = keySet1.iterator();


      while(keyIterator1.hasNext())
     {
        double Tvotes2=(double)Tvotes;
        String partyName1 = keyIterator1.next();


        pieData.get(partyName1);
        System.out.println(partyName1);


         Vote = pieData.get(partyName1);
        System.out.println("Vote \t" +Vote);
        System.out.println("Total Votes \t" +Tvotes2);


        percentage=(Vote/Tvotes2)*100;
        System.out.println("Percentage \t" +percentage);


        double deg1=(percentage/100)*360;
```

```
        int degree1=(int) deg1;
           System.out.println("DEGREE \t\t"+degree1);


           n=n+1;
           if(n==1)
                    g.setColor(Color.BLACK);
           if(n==2)
                    g.setColor(Color.ORANGE);
           if(n==3)
                    g.setColor(Color.BLUE);
           if(n==4)
                    g.setColor(Color.DARK_GRAY);
           if(n==5)
                    g.setColor(Color.RED);
           if(n==6)
                    g.setColor(Color.PINK);
                    if(n==7)
                    g.setColor(Color.GREEN);

             if(n>8)
             {
           Color color4=new Color(color1,color2,color3);
                       g.setColor(color4);
                      }


                        g.drawString(partyName1,X_axixs,500);
                 X_axixs=X_axixs+80;


                   g.fillArc(300,200,201,201,Iangle,degree1);
                   Iangle=Iangle+degree1;


        color1=color1+25;
        color2=color2+25;
        color3=color3+25;

         System.out.println("\n\n");
       }

       String label="PIE-CHART";
       Font font=new Font("ArialBlack",Font.ITALIC,20);
       g.drawRect(30,50,720,460);
              g.setColor(Color.black);
              g.setFont(font);
              g.drawString(label,360,150);
 }

}
```