```
package stubs;

import java.awt.EventQueue;

public class TestFrame {

    private JFrame frmTest;
    static Connection conn;
    private JPanel panel_output;
    private JLabel lblConnStatus;
    static String host_add = "172.27.71.142";
    private JButton BtnConn;
    private TextArea text_result;
    private JLabel xlabel;
    private JLabel ylabel;
    private JLabel label_zero;
    private JLabel label_xmax;
    private JLabel label_ymax;
    private JPanel panel_red;
    private JPanel panel_blue;
    private JLabel lblDelayed;
    private JLabel lblNormal;
    private JLabel lblFnumber;
    private JLabel lblPleaseSelectItems;
    private JComboBox<String> select_factorY;
    private JTextField input_xmax;
    private JTextField input_ymax;
    private JTextField input_add;
    private JPanel led_conn;
    private String logstr = new String();
    private JTextField input_delay;
    private JLabel lblDelayedMoreThan;
    private JLabel lblMinutes;
    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    TestFrame window = new TestFrame();
                    window.frmTest.setVisible(true);

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

    }

    /**
```

```
     * Create the application.
     */
    public TestFrame() {
        initialize();
    }

    /**
     * Initialize the contents of the frame.
     */
    private void initialize() {
        frmTest = new JFrame();
        frmTest.setTitle("Test");
        frmTest.setBounds(100, 100, 799, 496);
        frmTest.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frmTest.getContentPane().setLayout(null);

        panel_output = new JPanel();
        panel_output.setBorder(new BevelBorder(BevelBorder.LOWERED, null, null,
null, null));
        panel_output.setBackground(Color.WHITE);
        panel_output.setBounds(406, 62, 350, 350);
        frmTest.getContentPane().add(panel_output);

        lblConnStatus = new JLabel("Status: Unconnected");
        lblConnStatus.setBounds(54, 25, 201, 14);
        frmTest.getContentPane().add(lblConnStatus);

        input_add = new JTextField();
        input_add.setHorizontalAlignment(SwingConstants.RIGHT);
        input_add.setColumns(10);
        input_add.setBounds(36, 52, 173, 20);
        frmTest.getContentPane().add(input_add);

        //====================== Connect to the database button
====================
        BtnConn = new JButton("Connect");
        BtnConn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                //Connect to the database
                String add = input_add.getText();
                String driver = "com.mysql.jdbc.Driver";
                String url = new String();
                if(add.length()>0)
                {
                    url = "jdbc:mysql://"+add+":3306/traffic_weather";
                    host_add = add;
                }
                else
                    url = "jdbc:mysql://"+host_add+":3306/traffic_weather";
                String user = "admin";
                String password = "123456";
```

```
            try{
                Class.forName(driver);
                conn = DriverManager.getConnection(url, user, password);
                if(!conn.isClosed())
                {
                    showConnectStatus(true);
                    logstr = "Connected.\n\n" + logstr;
                    text_result.setText(logstr);
                }
            }
            catch (Exception e1) {
                e1.printStackTrace();
            }
            query_create_weaview();
        }
    });
    BtnConn.setBounds(237, 51, 97, 23);
    frmTest.getContentPane().add(BtnConn);

    text_result = new TextArea();
    text_result.setBounds(30, 254, 339, 173);
    frmTest.getContentPane().add(text_result);

    xlabel = new JLabel("X");
    xlabel.setHorizontalAlignment(SwingConstants.RIGHT);
    xlabel.setFont(new Font("Tahoma", Font.BOLD, 11));
    xlabel.setBounds(637, 429, 130, 14);
    frmTest.getContentPane().add(xlabel);

    ylabel = new JLabel("Y");
    ylabel.setFont(new Font("Tahoma", Font.BOLD, 11));
    ylabel.setBounds(363, 41, 115, 14);
    frmTest.getContentPane().add(ylabel);

    label_zero = new JLabel("0");
    label_zero.setFont(new Font("Tahoma", Font.BOLD, 11));
    label_zero.setBounds(397, 413, 22, 14);
    frmTest.getContentPane().add(label_zero);

    label_xmax = new JLabel("100");
    label_xmax.setHorizontalAlignment(SwingConstants.CENTER);
    label_xmax.setFont(new Font("Tahoma", Font.BOLD, 11));
    label_xmax.setBounds(724, 413, 59, 14);
    frmTest.getContentPane().add(label_xmax);

    label_ymax = new JLabel("100");
    label_ymax.setHorizontalAlignment(SwingConstants.RIGHT);
    label_ymax.setFont(new Font("Tahoma", Font.BOLD, 11));
    label_ymax.setBounds(363, 55, 37, 14);
    frmTest.getContentPane().add(label_ymax);

    panel_red = new JPanel();
```

```
        panel_red.setBackground(Color.RED);
        panel_red.setBorder(null);
        panel_red.setBounds(609, 41, 10, 10);
        frmTest.getContentPane().add(panel_red);

        panel_blue = new JPanel();
        panel_blue.setBackground(Color.BLUE);
        panel_blue.setBounds(498, 41, 10, 10);
        frmTest.getContentPane().add(panel_blue);

        lblDelayed = new JLabel("Delayed");
        lblDelayed.setBounds(629, 37, 46, 14);
        frmTest.getContentPane().add(lblDelayed);

        lblNormal = new JLabel("Normal");
        lblNormal.setBounds(518, 37, 46, 14);
        frmTest.getContentPane().add(lblNormal);

        lblFnumber = new JLabel("Logs");
        lblFnumber.setFont(new Font("Tahoma", Font.BOLD, 11));
        lblFnumber.setBounds(36, 233, 59, 14);
        frmTest.getContentPane().add(lblFnumber);

        lblPleaseSelectItems = new JLabel("Gram of Delays");
        lblPleaseSelectItems.setFont(new Font("Tahoma", Font.BOLD, 11));
        lblPleaseSelectItems.setHorizontalAlignment(SwingConstants.CENTER);
        lblPleaseSelectItems.setBounds(406, 21, 350, 14);
        frmTest.getContentPane().add(lblPleaseSelectItems);

        JComboBox<String> select_factorX = new JComboBox<String>();
        select_factorX.setModel(new DefaultComboBoxModel<String>(new String[]
{"Humidity (%)", "Visibility (km)", "Wind_Speed (km/s)", "Precipitation (m)",
"Temperature (C)"}));
        select_factorX.setBounds(45, 138, 133, 20);
        frmTest.getContentPane().add(select_factorX);


        select_factorY = new JComboBox<String>();
        select_factorY.setModel(new DefaultComboBoxModel<String>(new String[]
{"Humidity (%)", "Visibility (km)", "Wind_Speed (km/s)", "Precipitation (m)",
"Temperature (C)"}));
        select_factorY.setBounds(45, 173, 133, 20);
        frmTest.getContentPane().add(select_factorY);

        input_xmax = new JTextField();
        input_xmax.setText("0");
        input_xmax.setHorizontalAlignment(SwingConstants.RIGHT);
        input_xmax.setBounds(322, 138, 46, 20);
        frmTest.getContentPane().add(input_xmax);
        input_xmax.setColumns(10);

        input_ymax = new JTextField();
```

```
        input_ymax.setText("0");
        input_ymax.setHorizontalAlignment(SwingConstants.RIGHT);
        input_ymax.setColumns(10);
        input_ymax.setBounds(322, 173, 46, 20);
        frmTest.getContentPane().add(input_ymax);



        JLabel lblMaxValue = new JLabel("max value");
        lblMaxValue.setHorizontalAlignment(SwingConstants.CENTER);
        lblMaxValue.setBounds(308, 123, 73, 14);
        frmTest.getContentPane().add(lblMaxValue);

        JLabel lblX = new JLabel("X");
        lblX.setHorizontalAlignment(SwingConstants.CENTER);
        lblX.setBounds(10, 141, 35, 14);
        frmTest.getContentPane().add(lblX);

        JLabel lblY = new JLabel("Y");
        lblY.setHorizontalAlignment(SwingConstants.CENTER);
        lblY.setBounds(10, 176, 35, 14);
        frmTest.getContentPane().add(lblY);

        JComboBox<String> select_loc = new JComboBox<String>();
        select_loc.setModel(new DefaultComboBoxModel(new String[] {"Departure",
"Arrival"}));
        select_loc.setBounds(218, 157, 94, 20);
        frmTest.getContentPane().add(select_loc);

        JLabel lblOf = new JLabel("On");
        lblOf.setHorizontalAlignment(SwingConstants.CENTER);
        lblOf.setBounds(188, 160, 22, 14);
        frmTest.getContentPane().add(lblOf);

        JComboBox<String> select_type = new JComboBox<String>();
        select_type.setModel(new DefaultComboBoxModel(new String[] {"Flight",
"Train"}));
        select_type.setBounds(61, 100, 59, 20);
        frmTest.getContentPane().add(select_type);

        JLabel lblType = new JLabel("Type");
        lblType.setHorizontalAlignment(SwingConstants.CENTER);
        lblType.setBounds(20, 103, 37, 14);
        frmTest.getContentPane().add(lblType);

        //=================================== Get Information and execute query
=========================================
        JButton btn_drawscatter = new JButton("Execute");
        btn_drawscatter.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String X[] = ((String)select_factorX.getSelectedItem()).split("[\
\(\\) ;]");
```

```
                double xmax = Double.valueOf(input_xmax.getText());
                String Y[] = ((String)select_factorY.getSelectedItem()).split("[\
\(\\) ;]");
                double ymax = Double.valueOf(input_ymax.getText());
                String type = new
String(((String)select_type.getSelectedItem()).substring(0,1));
                String loc = new
String(((String)select_loc.getSelectedItem()).substring(0,1));
                drawScatterGram(type,loc,X[0],X[2],xmax,Y[0],Y[2],ymax);
            }
        });
        btn_drawscatter.setBounds(104, 204, 191, 23);
        frmTest.getContentPane().add(btn_drawscatter);

        led_conn = new JPanel();
        led_conn.setBackground(Color.RED);
        led_conn.setForeground(Color.RED);
        led_conn.setBounds(20, 21, 20, 20);
        frmTest.getContentPane().add(led_conn);

        input_delay = new JTextField();
        input_delay.setText("0");
        input_delay.setHorizontalAlignment(SwingConstants.RIGHT);
        input_delay.setColumns(10);
        input_delay.setBounds(266, 100, 46, 20);
        frmTest.getContentPane().add(input_delay);

        lblDelayedMoreThan = new JLabel("Delayed more than");
        lblDelayedMoreThan.setHorizontalAlignment(SwingConstants.CENTER);
        lblDelayedMoreThan.setBounds(140, 103, 115, 14);
        frmTest.getContentPane().add(lblDelayedMoreThan);

        lblMinutes = new JLabel("minutes");
        lblMinutes.setHorizontalAlignment(SwingConstants.CENTER);
        lblMinutes.setBounds(314, 103, 67, 14);
        frmTest.getContentPane().add(lblMinutes);
    }

    public void showConnectStatus(boolean status){
        if(status==true)
        {
            this.lblConnStatus.setText("Status: "+host_add + " Connected");
            led_conn.setBackground(Color.GREEN);
        }
        else
            this.lblConnStatus.setText("Status: Unconnected");
    }

    //=================================== Generate and execute Query
==========================================
    public void drawScatterGram(String type, String loc, String axisX, String
unitX, Double maxX, String axisY, String unitY, Double maxY){
```

```
        String sql = "SELECT "+axisX+", "+axisY +", Delay\n"
                +"FROM "+type+"_DELAY_WEATHER_"+loc;
        String type_name = new String();
        xlabel.setText(axisX+"("+unitX+")");
        ylabel.setText(axisY+"("+unitY+")");
        label_xmax.setText(maxX.toString());
        label_ymax.setText(maxY.toString());
        if(type.equals("T"))type_name="Train";
        else if(type.equals("F"))type_name="Flight";
        lblPleaseSelectItems.setText(axisY + " and "+axisX+"'s Effect on
"+type_name+" Delay"+"("+loc+")");
        try
        {
            Statement statement = conn.createStatement();
            ResultSet rs = statement.executeQuery(sql);
            int w = panel_output.getWidth();
            double x = 0;
            double y = 0;
            int delay = 0;
            int thea = Integer.valueOf(input_delay.getText());
            Graphics2D dc = (Graphics2D) panel_output.getGraphics();
            dc.setColor(Color.WHITE);
            dc.fillRect(0, 0, w, w);
            dc.setColor(new Color(200,200,200));
            for(int i=w/10;i<w;i+=w/10)
            {
                dc.drawLine(0, i, w, i);
                dc.drawLine(i, 0, i, w);
            }
            while(rs.next())
            {
                x = rs.getDouble(axisX);
                y = rs.getDouble(axisY);
                if(x==0)x=0.01*maxX/w;
                if(y==0)y=0.01*maxY/w;
                delay = rs.getInt("Delay");
                if(delay>thea)
                    dc.setColor(Color.RED);
                else
                    dc.setColor(Color.BLUE);
                dc.fillOval((int)(x*w/maxX), (int)(w-y*w/maxY), 5, 5);
            }
            logstr = "View Query Executed:\n"+sql+";\n\n" + logstr;
            text_result.setText(logstr);
        }
        catch(SQLException e1) {
            e1.printStackTrace();
        }
    }
    public void query_create_weaview()
    {
        String sql = "CREATE OR REPLACE VIEW F_DELAY_WEATHER_D AS\n"
```

```
                +"SELECT FNumber, Dep_date, Temperature, Humidity, Visibility,
Wind_Speed, Precipitation, -TimeStampDiff(Minute,Act_Arrive,Sch_Arrive) AS
Delay\n"
                + "FROM flight JOIN Airport ON Depart_Acode=Airport_Code\n"
                + "\t"+"JOIN flight_operation ON FNumber=FNumber_o\n"
                + "\t"+"JOIN Weather ON ACity_code=WCity_code AND Dep_date=WDate
AND HOUR(Act_Depart)=HOUR(WTime)\n"
                + "GROUP BY FNumber, Dep_Date\n"
                + "HAVING COUNT(*)>=1;";

        logstr = "View Created:\n"+sql+";\n\n" + logstr;
        text_result.setText(logstr);

        String sql2 = "CREATE OR REPLACE VIEW T_DELAY_WEATHER_D AS\n"
                +"SELECT TNumber, Dep_date, Temperature, Humidity, Visibility,
Wind_Speed, Precipitation, -TimeStampDiff(Minute,Act_Arrive,Sch_Arrive) AS
Delay\n"
                + "FROM train JOIN Station ON Depart_Scode=Station_Code\n"
                + "\t"+"JOIN train_operation ON TNumber=TNumber_o\n"
                + "\t"+"JOIN Weather ON SCity_code=WCity_code AND Dep_date=WDate
AND HOUR(Act_Depart)=HOUR(WTime)\n"
                + "GROUP BY TNumber, Dep_Date\n"
                + "HAVING COUNT(*)>=1;";

        logstr = "View Created:\n"+sql2+";\n\n" + logstr;
        text_result.setText(logstr);

        String sql3 = "CREATE OR REPLACE VIEW F_DELAY_WEATHER_A AS\n"
                +"SELECT FNumber, Dep_date, Temperature, Humidity, Visibility,
Wind_Speed, Precipitation, -TimeStampDiff(Minute,Act_Arrive,Sch_Arrive) AS
Delay\n"
                + "FROM flight JOIN Airport ON Arrive_Acode=Airport_Code\n"
                + "\t"+"JOIN flight_operation ON FNumber=FNumber_o\n"
                + "\t"+"JOIN Weather ON ACity_code=WCity_code AND Arr_date=WDate
AND HOUR(Act_Arrive)=HOUR(WTime)\n"
                + "GROUP BY FNumber, Dep_Date\n"
                + "HAVING COUNT(*)>=1;";

        logstr = "View Created:\n"+sql3+";\n\n" + logstr;
        text_result.setText(logstr);

        String sql4 = "CREATE OR REPLACE VIEW T_DELAY_WEATHER_A AS\n"
                +"SELECT TNumber, Dep_date, Temperature, Humidity, Visibility,
Wind_Speed, Precipitation, -TimeStampDiff(Minute,Act_Arrive,Sch_Arrive) AS
Delay\n"
                + "FROM train JOIN Station ON Arrive_Scode=Station_Code\n"
                + "\t"+"JOIN train_operation ON TNumber=TNumber_o\n"
                + "\t"+"JOIN Weather ON SCity_code=WCity_code AND Arr_date=WDate
AND HOUR(Act_Arrive)=HOUR(WTime)\n"
                + "GROUP BY TNumber, Dep_Date\n"
                + "HAVING COUNT(*)>=1;";
```

```
            logstr = "View Created:\n"+sql4+";\n\n" + logstr;
            text_result.setText(logstr);
            try
            {
                Statement statement = conn.createStatement();
                statement.executeUpdate(sql1);
                statement.executeUpdate(sql2);
                statement.executeUpdate(sql3);
                statement.executeUpdate(sql4);
            }
            catch(SQLException e1) {
                e1.printStackTrace();
            }
        }
}
```