

```
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.JTextArea;
import javax.swing.table.DefaultTableModel;
import java.util.*;

public class JTableRow extends JFrame {
    public static void main (String args[]){
        JFrame frame = new JFrame();

        JTable table = new JTable();

        Object[] columns = {"Medicalcode","Medicalname","Service_description","Quantity_available","Cost"};
        DefaultTableModel model = new
        DefaultTableModel();
        model.setColumnIdentifiers(columns);
        table.setModel(model);
        table.setBackground(Color.LIGHT_GRAY);
        table.setForeground(Color.black);
        Font font = new Font("",1,22);
        table.setFont(font);
        table.setRowHeight(30);

        JTextField textMedicalcode = new JTextField();
```

```

JTextField textMedicalname = new JTextField();

JTextField textService_description = new JTextField();

JTextField textQuantity_available = new JTextField();

JTextField textCost = new JTextField();

JTextArea billarea = new JTextArea(" ***** APOLLOMEDICALS *****
MEDICALBILL");

JButton btnAdd = new JButton("Add");

JButton btnDelete = new JButton("Delete");

JButton btnUpdate = new JButton("Update");

JButton sel_item = new JButton("Select_item");

JButton res = new JButton("reset");

JButton prin = new JButton("print");

textMedicalcode.setBounds(20, 220, 100, 25);

textMedicalname.setBounds(20, 250, 100, 25);

textService_description.setBounds(20, 280, 100,
25);

textQuantity_available.setBounds(20, 310, 100, 25);

textCost.setBounds(20,340,100,25);

sel_item.setBounds(100,490,120,25);

res.setBounds(220,490,100,25);

prin.setBounds(300,490,100,25);

btnAdd.setBounds(150, 220, 100, 25);

btnUpdate.setBounds(150, 265, 100, 25);

btnDelete.setBounds(150, 310, 100, 25);

billarea.setBounds(400,200,385,300);

// create JScrollPane

JScrollPane pane = new JScrollPane(table);

pane.setBounds(0, 0, 880, 200);
```

```

frame.setLayout(null);

frame.add(pane);

// add JTextFields to the JFrame
frame.add(textMedicalcode);
frame.add(textMedicalname);
frame.add(textService_description);
frame.add(textQuantity_available);
frame.add(textCost);

// add JButtons to the JFrame
frame.add(btnAdd);
frame.add(btnDelete);
frame.add(btnUpdate);
frame.add(sel_item);
frame.add(res);
frame.add(prin);
frame.add(billarea);

Date obj = new Date();
String date = obj.toString();

billarea.setText(billarea.getText()+"\n"+"t\t" + "Date::"+date+"\n");

billarea.setText(billarea.getText()+" MEDICAL_NAME      "+
"SERVICE_DESCRIPTION      "+ "      COST"+"n");

// create an array of objects to set the row data
Object[] row = new Object[5];

// button add row
btnAdd.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e) {
        row[0] = textMedicalcode.getText();
        row[1] = textMedicalname.getText();
    }
});

```

```

row[2] = textService_description.getText();
row[3] =textQuantity_available.getText();

row[4]=textCost.getText();
// add row to the model
model.addRow(row);
}
});
// button remove row
btnDelete.addActionListener(new ActionListener(){
@Override
public void actionPerformed(ActionEvent e) {
// i = the index of the selectedrow
int i = table.getSelectedRow();
if(i >= 0){
// remove a row from jtable
model.removeRow(i);
}
else{
System.out.println("Delete Error");
}
}
});
// get selected row data From table to textfields
table.addMouseListener(new MouseAdapter(){
@Override
public void mouseClicked(MouseEvent e){
// i = the index of the selected row
int i = table.getSelectedRow();

```

```

textMedicalcode.setText(model.getValueAt(i,
0).toString());
textMedicalname.setText(model.getValueAt(i,

1).toString());
textService_description.setText(model.getValueAt(i, 2).toString());
textQuantity_available.setText(model.getValueAt(i, 3).toString());
textCost.setText(model.getValueAt(i, 4).toString());
}
});
// button update row
btnUpdate.addActionListener(new ActionListener(){
@Override
public void actionPerformed(ActionEvent e) {
// i = the index of the selected row
int i = table.getSelectedRow();
if(i >= 0)
{
model.setValueAt(textMedicalcode.getText(
), i, 0);
model.setValueAt(textMedicalname.getText(
), i, 1);
model.setValueAt(textService_description.getText(), i, 2);
model.setValueAt(textQuantity_available.getText(), i, 3);
model.setValueAt(textCost.getText(), i, 4);
}
else{
System.out.println("Update Error");
}
}

```

```

}

});

sel_item.addActionListener(new ActionListener(){

    @Override

    public void actionPerformed(ActionEvent ae)

    {

        int i=0,total_amount=0,sum=525;

        int []a =new int[args.length];

        if(i<args.length)

        {

            int t=Integer.parseInt(args[0]);

            int q=Integer.parseInt(textCost.getText());

            a[i]=t*q;

            billarea.setText(billarea.getText()+"\n"+" 1 "+"    amoxicillin      "+"

            "infection caused by bacteria  " + "          525" );

            billarea.setText(billarea.getText()+"\n\n"+" 2  "+" + textMedicalname.getText() + "          "+"

            textService_description.getText() +"          " + a[i]);

            i++;

        }

        if(i==args.length-1){

            for(int p=0;p<args.length;p++){

                sum=sum+a[p];

            }

            billarea.setText(billarea.getText()+"\n\n\n"+"t\t"+"    TOTAL_AMOUNT :"+ sum +"\n");

        }

    }

});

res.addActionListener(new ActionListener(){

```

```

public void actionPerformed(ActionEvent e)
{
    billarea.setText(" ***** APOLLOMEDICALS ***** MEDICAL BILL");
    billarea.setText(billarea.getText()+"\n"+"t\t" + "Date: "+date+"\n");
    billarea.setText(billarea.getText()+" MEDICAL_NAME           "+
    "SERVICE_DESCRIPTION           "+ " COST"+"n");
}

});
prin.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent ae){
        try
        {
            billarea.setText(billarea.getText()+"\n\n\n"+"    The good physician treats the disease the great
            physician treats \n\t the patient who has the disease \n\t*****THANK YOU*****");
            billarea.print();
        }
        catch(Exception e)
        {
        }
    }
});
frame.setSize(1000,1000);
frame.setLocationRelativeTo(null);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setVisible(true);
}
}

```