Phurba Lama

Professor: Luis Rizo

MAC 190

12/6/2017

Test 2

**import** javax.swing.table.DefaultTableModel;

**import** javax.swing.\*;

**import** java.awt.\*;

**import** java.awt.event.\*;

**public** **class** TableGui {

**public** **static** **void** main(String[] args)

{

JFrame frame = **new** JFrame();

JTable table = **new** JTable();

JTable purchase=**new** JTable();

JTable search=**new** JTable();

//columnHeader for search Table

//column Header for purchase TableModel

Object[]columnHeader= {"Name","Details","Price"};

//takes the Table and purchase TableHeader and set the font to Bold

table.getTableHeader().setFont(**new** Font("",Font.***BOLD***,18));

purchase.getTableHeader().setFont(**new** Font("",Font.***BOLD***,20));

Object[][] data1;

//column Headers for TableModel

Object[] columns = {"Name",

"Starting Inventory",

"Received",

"Shipped",

"On Hand",

"Minimum Required",

"Needs Reorder",

"Product Details",

"Price"};

//row of data

Object[][] data = {

{"Cisco Router", 60,30,20, 20,10,

"yes","Details",399},

{"Dell Server", 60,20, 22, 20,10,

"yes","Details",99},

{"Phurba Router", 60,30, 24, 20,10,"yes","Details",199},

};

//TableModel for the list

DefaultTableModel model = **new** DefaultTableModel();

model.setDataVector(data, columns);

table.setModel(model);

table.setBackground(Color.***gray***);

table.setForeground(Color.***white***);

Font font= **new** Font("",1,18);

table.setFont(font);

table.setRowHeight(30);

//TableModel for the purchase

DefaultTableModel model1= **new** DefaultTableModel();

model1.setColumnIdentifiers(columnHeader);

purchase.setModel(model1);

//tableModel for search

DefaultTableModel model2= **new** DefaultTableModel();

model2.setColumnIdentifiers(columns);

search.setModel(model2);

JTextField name= **new** JTextField();

JTextField startingInventory= **new** JTextField();

JTextField received= **new** JTextField();

JTextField shipped= **new** JTextField();

JTextField onHand= **new** JTextField();

JTextField minimumRequired= **new** JTextField();

JTextField reOrder= **new** JTextField();

JTextField details= **new** JTextField();

JTextField price= **new** JTextField();

JTextField grandTotal= **new** JTextField();

JTextField searchText=**new** JTextField();

//creaking ImageIcon object for JButton

ImageIcon addImage=**new** ImageIcon("C:\\Users\\b331-u24\\Desktop\\New Folder\\addIcon.png");

ImageIcon updateImage=**new** ImageIcon("C:\\Users\\b331-u24\\Desktop\\New folder\\updateIcon.png");

ImageIcon deleteImage=**new** ImageIcon("C:\\Users\\b331-u24\\Desktop\\New folder\\deleteIcon.png");

ImageIcon clearImage=**new** ImageIcon("C:\\Users\\b331-u24\\Desktop\\New folder\\clearIcon.png");

ImageIcon buyImage=**new** ImageIcon("C:\\Users\\b331-u24\\Desktop\\New folder\\cartIcon.png");

ImageIcon inventoryImage=**new** ImageIcon("C:\\Users\\b331-u24\\Desktop\\New folder\\inventoryLogo.png");

//creating button Object

JButton btnAdd = **new** JButton(addImage);

JButton btnDelete = **new** JButton (deleteImage);

JButton btnUpdate = **new** JButton(updateImage);

JButton btnBuy = **new** JButton(buyImage);

JButton btnClear= **new** JButton(clearImage);

JButton btnSearch=**new** JButton("Search");

//creating label object

JLabel l1 = **new** JLabel("Name");

JLabel l2 = **new** JLabel("Startinv Inventory");

JLabel l3 = **new** JLabel("Received");

JLabel l4 = **new** JLabel("Shipped");

JLabel l5 = **new** JLabel("On Hand");

JLabel l6 = **new** JLabel("Minimum Required");

JLabel l7 = **new** JLabel("Needs Reorder");

JLabel l8 = **new** JLabel("Product Details");

JLabel l9 = **new** JLabel("Price");

JLabel l10= **new** JLabel("Grand Total");

JLabel l11=**new** JLabel(inventoryImage);

l10.setFont(**new** Font("",Font.***BOLD***,20));

//setting the x,y,width and height of label

l1.setBounds(50, 220, 100, 25);

l2.setBounds(180, 220, 100, 25);

l3.setBounds(350, 220, 100, 25);

l4.setBounds(500, 220, 100, 25);

l5.setBounds(650, 220, 100, 25);

l6.setBounds(800, 220, 100, 25);

l7.setBounds(980, 220, 100, 25);

l8.setBounds(1130, 220, 100, 25);

l9.setBounds(1300, 220, 100, 25);

l10.setBounds(1050, 520, 200, 25);

l11.setBounds(500, 300, 500, 100);

//setting the size and the location of textfields

name.setBounds(30, 260, 100, 25);

startingInventory.setBounds(180, 260, 100, 25);

received.setBounds(350, 260, 100, 25);

shipped.setBounds(500, 260, 100, 25);

onHand.setBounds(650, 260, 100, 25);

minimumRequired.setBounds(800, 260, 100, 25);

reOrder.setBounds(980, 260, 100, 25);

details.setBounds(1130, 260, 100, 25);

price.setBounds(1270, 260, 100, 25);

grandTotal.setBounds(1170, 520, 200, 25);

searchText.setBounds(600,550,200,25);

//setting the size and the location of Buttons

btnAdd.setBounds(30, 320, 64, 64);

btnUpdate.setBounds(130, 320, 64, 64);

btnDelete.setBounds(230, 320, 64, 64);

btnBuy.setBounds(1270, 320, 64, 64);

btnClear.setBounds(1170, 320, 64, 64);

btnSearch.setBounds(600,520,200,25);

//adding all the elements to the JFrame

//scroll pane for the Table

JScrollPane pane= **new** JScrollPane(table);

pane.setBounds(0,0,1400,200);

JScrollPane pane1= **new** JScrollPane(purchase);

pane1.setBounds(0, 400, 1400, 100);

JScrollPane pane2=**new** JScrollPane(search);

pane2.setBounds(0,600,1400,100);

frame.setLayout(**null**);

frame.add(pane);

frame.add(pane1);

frame.add(pane2);

//adding textfield in the Frame

frame.add(name);

frame.add(startingInventory);

frame.add(received);

frame.add(shipped);

frame.add(onHand);

frame.add(minimumRequired);

frame.add(reOrder);

frame.add(details);

frame.add(price);

frame.add(grandTotal);

frame.add(searchText);

//adding label in the Frame

frame.add(l1);

frame.add(l2);

frame.add(l3);

frame.add(l4);

frame.add(l5);

frame.add(l6);

frame.add(l7);

frame.add(l8);

frame.add(l9);

frame.add(l10);

frame.add(l11);

//adding Buttons in the Frame

frame.add(btnAdd);

frame.add(btnDelete);

frame.add(btnUpdate);

frame.add(btnBuy);

frame.add(btnClear);

frame.add(btnSearch);

//creating object of row to add new values to the table

Object[] row = **new** Object[9];

//row1 is for purchase table

Object[] row1= **new** Object[3];

//row2 is for search

Object[] row2=**new** Object[9];

//when using presses ADD Button

btnAdd.addActionListener(**new** ActionListener() {

JFrame f= **new** JFrame();

JFrame f1=**new** JFrame();

@Override

**public** **void** actionPerformed(ActionEvent e)

{

**for** (**int** i = 0; i < model.getRowCount(); i++){

**if**( model.getValueAt(i,0).equals(name.getText()))

{

JOptionPane.*showMessageDialog*(f1,"Item Already Exist.","Alert",JOptionPane.***WARNING\_MESSAGE***);

**break**;

}

**else** {

row[0]= name.getText();

row[1]= startingInventory.getText();

row[2]= received.getText();

row[3]= shipped.getText();

row[4] = onHand.getText();

row[5]= minimumRequired.getText();

row[6]= reOrder.getText();

row[7]= details.getText();

row[8]= price.getText();

model.addRow(row);

JOptionPane.*showMessageDialog*(f,"Successfully Added.","Alert",JOptionPane.***WARNING\_MESSAGE***,addImage);

**break**;

}

}

}

});

//when user search item

btnSearch.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e)

{

**for**(**int** i=0;i<model.getRowCount();i++)

{

//checks the database

**if**( model.getValueAt(i,0).toString().equalsIgnoreCase(searchText.getText().toString()))

{

row2[0] = model.getValueAt(i, 0);

row2[1] = model.getValueAt(i, 1);

row2[2] = model.getValueAt(i, 2);

row2[3] = model.getValueAt(i, 3);

row2[4] = model.getValueAt(i, 4);

row2[5] = model.getValueAt(i, 5);

row2[6] = model.getValueAt(i, 6);

row2[7] = model.getValueAt(i, 7);

row2[8] = model.getValueAt(i, 8);

model2.addRow(row2);

}

}

}

});

//when user presses Buy button it will take the name,details and price and show it on tablemodel below

btnBuy.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e)

{

**int** i=table.getSelectedRow(); // chooses the selected row

**int** onHand= (Integer)(model.getValueAt(i, 4))-1; //changing object to integer and deducts it by 1

model.setValueAt((onHand), i, 4); // new value is set which changes onHand value

//adds name details and price to new table below

row1[0]= name.getText();

row1[1]=details.getText();

row1[2]=price.getText();

model1.addRow(row1); //addRow methods adds new row to the model

//Calculates the grandtotal of your purchase

**int** total=0;

**for** (**int** t = 0; t < model1.getRowCount(); t++){

total +=Integer.*parseInt*( model1.getValueAt(t, 2).toString() );// 3rd column . row column indexes are 0 based

}

grandTotal.setText(Integer.*toString*(total)); //sets the total to grandtotal textfield

}

});

//Clear Button for purchase

btnClear.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e)

{

**while**(model1.getRowCount()>0)

{

model1.removeRow(0);

}

}

});

//delete Button

btnDelete.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e)

{

JFrame f=**new** JFrame();

**int** i = table.getSelectedRow();

**if**(i>=0) {

model.removeRow(i);

JOptionPane.*showMessageDialog*(f,"Successfully Deleted.","Alert",JOptionPane.***WARNING\_MESSAGE***,deleteImage);

}

**else** {

System.***out***.println("Delete Error");

}

}

});

//adding mouse listener to take the value from the table and set it to textfield

table.addMouseListener(**new** MouseAdapter () {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**int** i = table.getSelectedRow();

name.setText(model.getValueAt(i, 0).toString());

startingInventory.setText(model.getValueAt(i, 1).toString());

received.setText(model.getValueAt(i, 2).toString());

shipped.setText(model.getValueAt(i, 3).toString());

onHand.setText(model.getValueAt(i, 4).toString());

minimumRequired.setText(model.getValueAt(i, 5).toString());

reOrder.setText(model.getValueAt(i, 6).toString());

details.setText(model.getValueAt(i, 7).toString());

price.setText(model.getValueAt(i, 8).toString());

}

});

//Updating the box

btnUpdate.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

**int** i = table.getSelectedRow();

JFrame f= **new** JFrame();

**if**(i>=0)

{

model.setValueAt(name.getText(), i, 0);

model.setValueAt(startingInventory.getText(), i, 1);

model.setValueAt(received.getText(), i, 2);

model.setValueAt(shipped.getText(), i,3);

model.setValueAt(onHand.getText(), i, 4);

model.setValueAt(minimumRequired.getText(), i, 5);

model.setValueAt(reOrder.getText(), i, 6);

model.setValueAt(details.getText(), i,7);

model.setValueAt(price.getText(), i, 8);

JOptionPane.*showMessageDialog*(f,"Successfully Updated.","Alert",JOptionPane.***WARNING\_MESSAGE***,updateImage);

}

**else** {

System.***out***.println("Update Error");

}

}

});

frame.setTitle("Inventory"); //gives title to the frame

frame.setSize(1420,800); //sets the size

frame.setLocationRelativeTo(**null**); //sets the location in the middle of the screen

frame.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***); //allows program to shut when closed

frame.setVisible(**true**); //makes the frame visible

}

}

**Output:**

