

BANK PROJECT (2.1)

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;

public class JavaBank extends JFrame {

    /**
     *
     */
    private static final long serialVersionUID = 1L;
    // Make these variables publicly available
    public String Name;
    public int Accountnum;
    public int Balance;

    // JPanel for user inputs
    private JPanel inputDetailJPanel;

    // JLabel and JTextField for account name
    private JLabel NameJLabel;
    private JTextField NameJTextField;

    // JLabel and JTextField for account number
    private JLabel AccountnumJLabel;
    private JTextField AccountnumJTextField;

    // JLabel and JTextField for balance
    private JLabel BalanceJLabel;
    private JTextField BalanceJTextField;

    // JLabel and JTextField for withdraw
    private JLabel DepositJLabel;
    private JTextField DepositJTextField;

    // JLabel and JTextField for Withdraw
    private JLabel WithdrawJLabel;
    private JTextField WithdrawJTextField;

    // JButton to create account
    private JButton CreateAccountJButton;
```

```

// JButton to delete account
private JButton DeleteAccountJButton;

// JButton to make transaction
private JButton TransactionJButton;

// JButton to display account
private JButton DisplayJButton;

// JLabel and JTextArea to display account details
private JLabel displayJLabel;
private static JTextArea displayJTextArea;

// constants
//public final static Maximum Accounts that can be created;
public final static int MaxAccounts = 10;

// one-dimensional array to store Account names as Empty or Used
static String AccountNames[] = new String[MaxAccounts];

// two-dimensional array to store Account details

static Account myAccounts[] = new Account[MaxAccounts];

static int noAccounts = 0;

// constructor

public JavaBank() {
    for (int i=0; i <10; i++) {
        AccountNames[i] = "EMPTY";
        //System.out.println(AccountNames[i]);

    }
    createUserInterface();
}

// create and position GUI components; register event handlers
private void createUserInterface() {
    // get content pane for attaching GUI components
    Container contentPane = getContentPane();

    // enable explicit positioning of GUI components

```

```
contentPane.setLayout(null);

// set up inputDetailJPanel
inputDetailJPanel = new JPanel();
inputDetailJPanel.setBounds(16, 16, 208, 250);
inputDetailJPanel.setBorder(new TitledBorder("Input Details"));
inputDetailJPanel.setLayout(null);
contentPane.add(inputDetailJPanel);

// set up NameJLabel
NameJLabel = new JLabel();
NameJLabel.setBounds(8, 32, 90, 23);
NameJLabel.setText("Name:");
inputDetailJPanel.add(NameJLabel);

// set up NameJTextField
NameJTextField = new JTextField();
NameJTextField.setBounds(112, 32, 80, 21);
NameJTextField.setHorizontalAlignment(JTextField.RIGHT);
inputDetailJPanel.add(NameJTextField);

// set up AccountnumJLabel
AccountnumJLabel = new JLabel();
AccountnumJLabel.setBounds(8, 56, 100, 23);
AccountnumJLabel.setText("Account Number:");
inputDetailJPanel.add(AccountnumJLabel);

// set up AccountnumJTextField
AccountnumJTextField = new JTextField();
AccountnumJTextField.setBounds(112, 56, 80, 21);
AccountnumJTextField.setHorizontalAlignment(JTextField.RIGHT);
inputDetailJPanel.add(AccountnumJTextField);

// set up BalanceJLabel
BalanceJLabel = new JLabel();
BalanceJLabel.setBounds(8, 80, 60, 23);
BalanceJLabel.setText("Balance:");
inputDetailJPanel.add(BalanceJLabel);

// set up BalanceJTextField
BalanceJTextField = new JTextField();
BalanceJTextField.setBounds(112, 80, 80, 21);
BalanceJTextField.setHorizontalAlignment(JTextField.RIGHT);
inputDetailJPanel.add(BalanceJTextField);

// set up DepositJLabel
```

```

DepositJLabel = new JLabel();
DepositJLabel.setBounds(8, 104, 80, 23);
DepositJLabel.setText("Deposit:");
inputDetailJPanel.add(DepositJLabel);

// set up DepositJTextField
DepositJTextField = new JTextField();
DepositJTextField.setBounds(112, 104, 80, 21);
DepositJTextField.setHorizontalAlignment(JTextField.RIGHT);
inputDetailJPanel.add(DepositJTextField);

// set up WithdrawJLabel
WithdrawJLabel = new JLabel();
WithdrawJLabel.setBounds(8, 128, 60, 23);
WithdrawJLabel.setText("Withdraw:");
inputDetailJPanel.add(WithdrawJLabel);

// set up WithdrawJTextField
WithdrawJTextField = new JTextField();
WithdrawJTextField.setBounds(112, 128, 80, 21);
WithdrawJTextField.setHorizontalAlignment(JTextField.RIGHT);
inputDetailJPanel.add(WithdrawJTextField);

// set up CreateAccountButton
CreateAccountJButton = new JButton();
CreateAccountJButton.setBounds(112, 152, 80, 24);
CreateAccountJButton.setText("Create");
inputDetailJPanel.add(CreateAccountJButton);
CreateAccountJButton.addActionListener(

new ActionListener() {
    // event handler called when CreateAccountJButton
    // is clicked
    public void actionPerformed(ActionEvent event) {
        CreateAccountJButtonActionPerformed(event);
    }
}

); // end call to addActionListener

// set up DeleteAccountButton
DeleteAccountJButton = new JButton();
DeleteAccountJButton.setBounds(16, 152, 80, 24);
DeleteAccountJButton.setText("Delete");
inputDetailJPanel.add(DeleteAccountJButton);

```

```

DeleteAccountJButton.addActionListener(

new ActionListener() // anonymous inner class
{
    // event handler called when DeleteAccountJButton
    // is clicked
    public void actionPerformed(ActionEvent event) {
        DeleteAccountJButtonActionPerformed(event);
    }
}

); // end call to addActionListener

// set up TransactionJButton
TransactionJButton = new JButton();
TransactionJButton.setBounds(16, 180, 176, 24);
TransactionJButton.setText("Make Transaction");
inputDetailJPanel.add(TransactionJButton);
TransactionJButton.addActionListener(

new ActionListener() // anonymous inner class
{
    // event handler called when TransactionJButton
    // is clicked
    public void actionPerformed(ActionEvent event) {
        TransactionJButtonActionPerformed(event);
    }
} // end anonymous inner class

); // end call to addActionListener

// set up DisplayJButton
DisplayJButton = new JButton();
DisplayJButton.setBounds(16, 208, 176, 24);
DisplayJButton.setText("Display Accounts");
inputDetailJPanel.add(DisplayJButton);
DisplayJButton.addActionListener(

new ActionListener() // anonymous inner class
{
    // event handler called when TransactionJButton
    // is clicked
    public void actionPerformed(ActionEvent event) {

```

```

        DisplayJButtonActionPerformed(event);
    }

    } // end anonymous inner class

    ); // end call to addActionListener

// set up displayJLabel
displayJLabel = new JLabel();
displayJLabel.setBounds(240, 16, 150, 23);
displayJLabel.setText("Account Details:");
contentPane.add(displayJLabel);

// set up displayJTextArea
displayJTextArea = new JTextArea();
JScrollPane scrollPane = new JScrollPane(displayJTextArea);
scrollPane.setBounds(240,48,402,184);

scrollPane.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
contentPane.add(scrollPane);
displayJTextArea.setText("Welcome to Java Bank - There are currently no Accounts
created");

// clear other JTextFields for new data
NameJTextField.setText(" ");
AccountnumJTextField.setText("0");
BalanceJTextField.setText("0");
DepositJTextField.setText("0");
WithdrawJTextField.setText("0");

// set properties of application's window
setTitle("Java Bank"); // set title bar string
setSize(670, 308); // set window size
setVisible(true); // display window

} // end method createUserInterface

private void CreateAccountJButtonActionPerformed(ActionEvent event) {
    // System.out.println("Create Account Button Clicked");

    displayJTextArea.setText("");

```

```

Name = "";

//Get Name from Text Field
Name = NameJTextField.getText();

//Get Accountnum from Text Field and convert to int unless blank then set to 0
if (AccountnumJTextField.getText() == "0") {
    Accountnum = 0;
}
else {
    Accountnum = Integer.parseInt(AccountnumJTextField.getText());
}

//Get Balance from Text Field and convert to int unless blank then set to 0
if (BalanceJTextField.getText() == "0") {
    Balance = 0;
}
else {
    Balance = Integer.parseInt(BalanceJTextField.getText());
}

//int emptyAccount = 11;

if ((noAccounts <= 9) & (Name != "") & (Accountnum != 0)) {
    myAccounts[noAccounts] = new Account(Name,Accountnum,Balance);

    AccountNames[noAccounts] = "USED";
    //System.out.println(myAccounts[noAccounts].getaccountname());
    //emptyAccount = i;

    displayJTextArea.setText(myAccounts[noAccounts].getaccountname() + " " +
myAccounts[noAccounts].getaccountnum() + " " + myAccounts[noAccounts].getbalance());
    noAccounts++;
    System.out.println(noAccounts);
}
else {
    displayJTextArea.setText("Both the Name field and Account Number must be
completed");
}

if (noAccounts == 10) {
    // Once account 10 is created. All accounts full.
    displayJTextArea.setText("All Accounts Full!");
}

```

```

    }

    // clear other JTextFields for new data
    NameJTextField.setText(" ");
    AccountnumJTextField.setText("0");
    BalanceJTextField.setText("0");
    DepositJTextField.setText("0");
    WithdrawJTextField.setText("0");

}

private void DeleteAccountJButtonActionPerformed(ActionEvent event) {

    displayJTextArea.setText("Oops this isnt coded in this version!");
    //Name = NameJTextField.getText();
    //System.out.println("Delete Account: " + Name);

    // Enter code to delete here

    // clear JTextFields for new data

    NameJTextField.setText(" ");
    AccountnumJTextField.setText("0");
    BalanceJTextField.setText("0");
    DepositJTextField.setText("0");
    WithdrawJTextField.setText("0");

}

private void TransactionJButtonActionPerformed(ActionEvent event) {

    displayJTextArea.setText("");

    if (noAccounts == 0) {
        displayJTextArea.setText("No Accounts currently created");
    } else {

        // get user input
        int Accountnum = Integer.parseInt(AccountnumJTextField.getText());
        int Deposit = Integer.parseInt(DepositJTextField.getText());
        int Withdraw = Integer.parseInt(WithdrawJTextField.getText());
    }
}

```



```

        for (int i=0; i<noAccounts; i++) {
            if ((myAccounts[i].getaccountnum() == Accountnum) && (Deposit>0)) {
                myAccounts[i].setbalance(myAccounts[i].getbalance()+Deposit);
                displayJTextArea.setText(myAccounts[i].getaccountname() + " " +
myAccounts[i].getaccountnum() + " " + myAccounts[i].getbalance());
            }

            if ((myAccounts[i].getaccountnum() == Accountnum) && (Withdraw>0)) {

                myAccounts[i].setbalance(myAccounts[i].getbalance()-Withdraw);
                displayJTextArea.setText(myAccounts[i].getaccountname() + " " +
myAccounts[i].getaccountnum() + " " + myAccounts[i].getbalance());
            }

        }
    }

    // clear other JTextFields for new data
    NameJTextField.setText(" ");
    AccountnumJTextField.setText("0");
    BalanceJTextField.setText("0");
    DepositJTextField.setText("0");
    WithdrawJTextField.setText("0");

}

private void DisplayJButtonActionPerformed(ActionEvent event) {

    Name = NameJTextField.getText();
    displayJTextArea.setText("");

    if (noAccounts == 0) {
        displayJTextArea.setText("No Accounts currently created");
    } else {
        for (int i=0; i<noAccounts; i++) {

            displayJTextArea.append(myAccounts[i].getaccountname() + " " +
myAccounts[i].getaccountnum() + " " + myAccounts[i].getbalance() + "\n");

        }
    }

    // clear other JTextFields for new data

```

```

        NameJTextField.setText(" ");
        AccountnumJTextField.setText("0");
        BalanceJTextField.setText("0");
        DepositJTextField.setText("0");
        WithdrawJTextField.setText("0");

    }

    public static void main(String[] args) {
        // Populate arrays with the word EMPTY
        // so we can check to see if the values are empty later

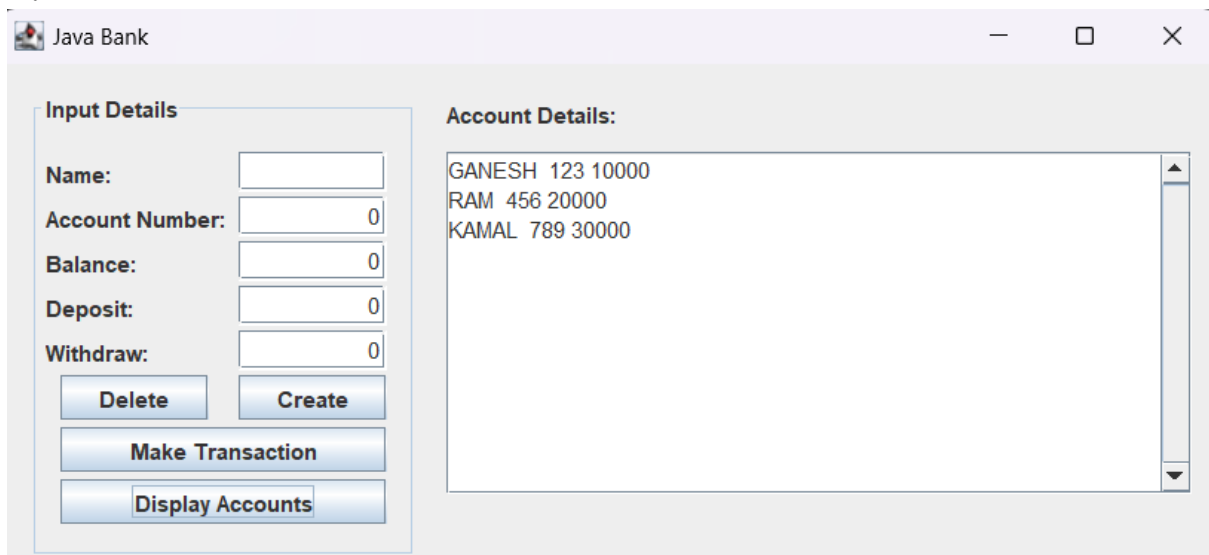
        JavaBank application = new JavaBank();
        application.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

```

QUESTIONS:

1. DISPLAY ACCOUNTS

A:



The screenshot shows a Java Swing window titled "Java Bank". On the left, under "Input Details", there are five text fields for Name, Account Number, Balance, Deposit, and Withdraw, each with a "0" in the Account Number, Balance, Deposit, and Withdraw fields. Below these fields are four buttons: "Delete", "Create", "Make Transaction", and "Display Accounts". The "Display Accounts" button is highlighted. On the right, under "Account Details:", a text area displays the following text:

GANESH	123	10000
RAM	456	20000
KAMAL	789	30000

2. CREATE ACCOUNTS

A:

Java Bank

Input Details

Name:

Account Number:

Balance:

Deposit:

Withdraw:

Delete Create

Make Transaction

Display Accounts

Account Details:

ABBURI 147 60000

3. DELETE ACCOUNTS

A: THIS CODE CANNOT DO THE DELETE OPERATION

4.MAKE A WITHDRAWAL TRANSACTION

A: YES IT CAN DO THE WITHDRAWAL TRANSACTION

5.MAKE A DEPOSIT TRANSCATION

A: YES IT CAN DO THE DEPOSIT TRANSACTION

6. CAN YOU DISPLAY ACCOUNTS BEFORE ANY ARE CREATED?

A: NO WE CAN'T DISPLAY ACCOUNTS BEFORE CREATED

7. CAN YOU CREATE AN ACCOUNT WITHOUT ENTERING ANYTHING IN THE FIELDS?

A: NO WE CAN'T CREATE AN ACCOUNT WITHOUT ENTERING ANYTHING IN THE FIELDS

8. CAN YOU MAKE A WITHDRAWAL TRANSACTION WITH NO AMOUNT IN THE WITHDRAWAL FIELD?

A: NO WE CAN'T MAKE A WITHDRAWAL TRANSACTION WITH NO AMOUNT IN THE WITHDRAWAL FIELD

9. CAN YOU DO A DEPOSIT TRANSACTION WITH NO AMOUNT IN THE DEPOSIT FIELD?

A: NO WE CAN'T DO DEPOSIT TRANSACTION WITHOUT NO AMOUNT IN THE FIELD

10. WHAT OTHER QUESTIONS DO YOU HAVE ABOUT THE JAVABANK APPLICATION?

A: NO OTHER QUESTIONS

11. WHAT CHANGES WOULD YOU MAKE TO THE CURRENT APPLICATION TO MAKE IT FUNCTION BETTER

A: WANT TO CREATE THE DELETE ACCOUNTS IN THE BANK OPTION

12. WHAT ADDITIONS WOULD YOU MAKE TO THE CURRENT APPLICATION TO INCREASE ITS FUNCTIONALITY?

A: WANT TO MAKE JOINT ACCOUNT'S OF THE FAMILY PERSONS MAKE CHANGES IN THE ABOVE APPLICATION TO INCREASE ITS FUNCTIONALITY