Name: Joshua Jadge Garcia

JTransfer.java

```java
...4 lines
package prjbank2;

import java.util.ArrayList;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**...4 lines */
public class JTransfer extends javax.swing.JFrame {

    /** Creates new form JTransfer ...3 lines */

    DefaultTableModel tbl;
    Connect conn;
    private String uname;
    public JTransfer(String username) {
        initComponents();
        tbl = (DefaultTableModel) tbl_account.getModel();
        conn = new Connect();
        uname = username;
        displayTable(username);
        btn_TransBal.setVisible( aFlag: false);
        lbl_error.setVisible( aFlag: false);
        usernametag.setText("<<" + username + ">>");
    }

    public JTransfer() {
    }
```

```java
    public void displayTable(String username){
       ArrayList<Account> acc = conn.displayAccount(username);
       for(Account c : acc){
           String data[]={c.getAccountNumber(),Double.toString( d: c.getBalance())};
            tbl.addRow( rowData: data);
       }

   }
```

```java
    private void tbl_accountMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        btn_TransBal.setEnabled( b: true);
        btn_TransBal.setVisible( aFlag: true);
    }
```

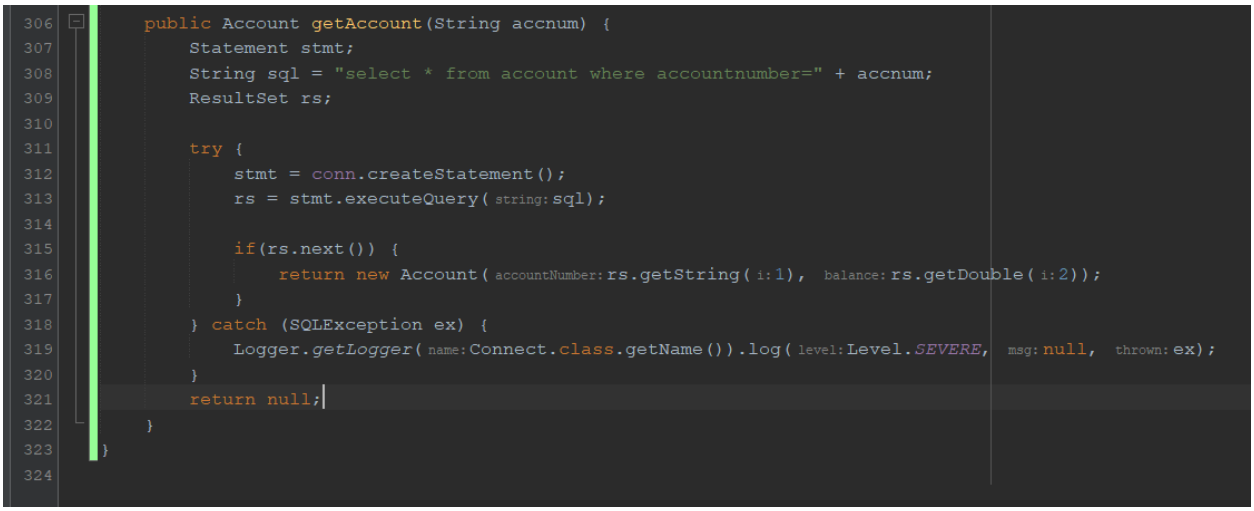
```
    private void btn_TransBalActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

        try{
            String accnum = tf_accnum.getText();
            double bal = Double.parseDouble( s:tf_bal.getText());
            double src_update_bal = 0;
            Account source_Account, target_Account;

            if(bal < 0) {
                throw new IllegalArgumentException( s:"Negative input");
            }

            if(accnum.isBlank() || tf_bal.getText().isBlank()) {
                JOptionPane.showMessageDialog( parentComponent:null,  message:"Fields should not be empty");
            } else {
                int index=tbl_account.getSelectedRow();
                String acc_src = (String) tbl_account.getValueAt( row:index, column:0);
                source_Account = conn.getAccount( accnum:acc_src);
                target_Account = conn.getAccount(accnum);
                if(target_Account == null) {
                    lbl_error.setVisible( aFlag:true);
                } else {
```

```
                } else {
                    src_update_bal = source_Account.getBalance() - bal;
                    if(src_update_bal < 0) {
                        throw new Exception( message:"Invalid Input");
                    }
                    lbl_error.setVisible( aFlag:false);
                    source_Account.setBalance( balance:src_update_bal);
                    target_Account.setBalance(target_Account.getBalance()+ bal);
                    if(conn.instantUpdate( acc:source_Account) && conn.instantUpdate( acc:target_Account)) {
                        JOptionPane.showMessageDialog( parentComponent:null,  message:"Transferred Successfully");
                        tbl.setRowCount( rowCount:0);
                        displayTable( username:uname);
                        btn_TransBal.setVisible( aFlag:false);
                        btn_TransBal.setEnabled( b:false);
                    } else {
                        JOptionPane.showMessageDialog( parentComponent:null,  message:"Failed Transfer");
                    }
                }
            }
        } catch(NumberFormatException e) {
            JOptionPane.showMessageDialog( parentComponent:null,  message:"Invalid Balance Input");
            tf_bal.setText( t:"");
        }catch(IllegalArgumentException e) {
            JOptionPane.showMessageDialog( parentComponent:null,  message:"Input must not be negative");
        } catch(Exception e) {
            JOptionPane.showMessageDialog( parentComponent:null,  message:"Invalid Input");
        }
    }
```

Connect.instantUpdate

```
287    public boolean instantUpdate(Account acc) {
288        Statement stmt;
           String sql=null;
           ResultSet rs=null, rs2 = null;
291        try {
292            stmt = conn.createStatement();
293            sql="select * from account where accountnumber='"+acc.getAccountNumber()+"'";
294            rs =stmt.executeQuery( string: sql);
295            if(rs.next()){
296                sql="update account set balance=" + acc.getBalance() + " where accountnumber=" + acc.getAccountNumber();
297                stmt.executeUpdate( string: sql);
298                return true;
299            }
300        } catch (SQLException ex) {
301            Logger.getLogger( name: Connect.class.getName()).log( level: Level.SEVERE,  msg: null,  thrown: ex);
302        }
303        return false;
304    }
```

Connect.getAccount



```
306    public Account getAccount(String accnum) {
307        Statement stmt;
308        String sql = "select * from account where accountnumber=" + accnum;
309        ResultSet rs;
310
311        try {
312            stmt = conn.createStatement();
313            rs = stmt.executeQuery( string: sql);
314
315            if(rs.next()) {
316                return new Account( accountNumber: rs.getString( i: 1),  balance: rs.getDouble( i: 2));
317            }
318        } catch (SQLException ex) {
319            Logger.getLogger( name: Connect.class.getName()).log( level: Level.SEVERE,  msg: null,  thrown: ex);
320        }
321        return null;
322    }
323}
324
```