1.

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class JavaBank {

private AbstractBankAccount[] myAccounts;

private JTextArea jTextArea;

private JTextField amountField;

private JComboBox<AbstractBankAccount> accountComboBox;

public JavaBank() {

myAccounts = new AbstractBankAccount[10]; // Example initialization

jTextArea = new JTextArea(); // Initialize the JTextArea

amountField = new JTextField(); // Initialize the JTextField for amount

accountComboBox = new JComboBox<>(); // Initialize the JComboBox for accounts

// Add some example accounts

myAccounts[0] = new CheckingAccount("12345", 1000.00);

myAccounts[1] = new SavingsAccount("67890", 2000.00);

for (AbstractBankAccount account : myAccounts) {

if (account != null) {

accountComboBox.addItem(account);

}

}

JButton makeTransactionButton = new JButton("Make Transaction");

makeTransactionButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

handleTransaction();

}

});

// Add components to the GUI and set up layout (not shown)

}

private void handleTransaction() {

try {

AbstractBankAccount selectedAccount = (AbstractBankAccount) accountComboBox.getSelectedItem();

if (selectedAccount == null) {

throw new IllegalArgumentException("No account selected.");

}

double amount = Double.parseDouble(amountField.getText());

if (amount <= 0) {

throw new IllegalArgumentException("Amount must be greater than zero.");

}

// Perform the transaction (deposit or withdrawal)

// Example: Assuming the transaction type is determined by some logic

// For demonstration, let's assume a deposit:

// selectedAccount.deposit(amount);

// If transaction is successful, update the text area

displayAccountDetails(selectedAccount);

} catch (NumberFormatException ex) {

jTextArea.setText("Invalid amount format. Please enter a valid number.");

} catch (IllegalArgumentException ex) {

jTextArea.setText(ex.getMessage());

} catch (Exception ex) {

jTextArea.setText("An unexpected error occurred: " + ex.getMessage());

}

}

private void displayAccountDetails(AbstractBankAccount account) {

jTextArea.setText(account.toString());

}

public static void main(String[] args) {

new JavaBank();

}

}

2.

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class JavaBank {

private AbstractBankAccount[] myAccounts;

private JTextArea jTextArea;

private JTextField amountField;

private JComboBox<AbstractBankAccount> accountComboBox;

public JavaBank() {

myAccounts = new AbstractBankAccount[10];

jTextArea = new JTextArea();

amountField = new JTextField();

accountComboBox = new JComboBox<>();

myAccounts[0] = new CheckingAccount("12345", 1000.00);

myAccounts[1] = new SavingsAccount("67890", 2000.00);

for (AbstractBankAccount account : myAccounts) {

if (account != null) {

accountComboBox.addItem(account);

}

}

JButton makeTransactionButton = new JButton("Make Transaction");

makeTransactionButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

handleTransaction();

}

});

// Add components to the GUI and set up layout (not shown)

}

private void handleTransaction() {

try {

AbstractBankAccount selectedAccount = (AbstractBankAccount) accountComboBox.getSelectedItem();

if (selectedAccount == null) {

throw new myException("No account selected.");

}

double amount = Double.parseDouble(amountField.getText());

if (amount <= 0) {

throw new myException("Amount must be greater than zero.");

}

// Perform the transaction (deposit or withdrawal)

// Example: Assuming the transaction type is determined by some logic

// For demonstration, let's assume a deposit:

// selectedAccount.deposit(amount);

displayAccountDetails(selectedAccount);

} catch (NumberFormatException ex) {

jTextArea.setText("Invalid amount format. Please enter a valid number.");

} catch (myException ex) {

jTextArea.setText(ex.getMessage());

} catch (Exception ex) {

jTextArea.setText("An unexpected error occurred: " + ex.getMessage());

}

}

private void displayAccountDetails(AbstractBankAccount account) {

jTextArea.setText(account.toString());

}

public static void main(String[] args) {

new JavaBank();

}

}

3.

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class JavaBank {

private AbstractBankAccount[] myAccounts;

private JTextArea jTextArea;

private JTextField amountField;

private JComboBox<AbstractBankAccount> accountComboBox;

public JavaBank() {

myAccounts = new AbstractBankAccount[10];

jTextArea = new JTextArea();

amountField = new JTextField();

accountComboBox = new JComboBox<>();

myAccounts[0] = new CheckingAccount("12345", 1000.00);

myAccounts[1] = new SavingsAccount("67890", 2000.00);

for (AbstractBankAccount account : myAccounts) {

if (account != null) {

accountComboBox.addItem(account);

}

}

JButton makeTransactionButton = new JButton("Make Transaction");

makeTransactionButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

handleTransaction();

}

});

// Add components to the GUI and set up layout (not shown)

}

private void handleTransaction() {

try {

AbstractBankAccount selectedAccount = (AbstractBankAccount) accountComboBox.getSelectedItem();

if (selectedAccount == null) {

throw new myException("No account selected.");

}

double amount = Double.parseDouble(amountField.getText());

if (amount <= 0) {

throw new myException("Amount must be greater than zero.");

}

// Perform the transaction (deposit or withdrawal)

// Example: Assuming the transaction type is determined by some logic

// For demonstration, let's assume a deposit:

// selectedAccount.deposit(amount);

displayAccountDetails(selectedAccount);

} catch (NumberFormatException ex) {

jTextArea.setText("Invalid amount format. Please enter a valid number.");

} catch (myException ex) {

jTextArea.setText(ex.getMessage());

} catch (Exception ex) {

myException newExc = new myException("An unhandled error occurred!!");

jTextArea.setText(newExc.getMessage());

}

}

private void displayAccountDetails(AbstractBankAccount account) {

jTextArea.setText(account.toString());

}

public static void main(String[] args) {

new JavaBank();

}

}

4.

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class JavaBank {

private AbstractBankAccount[] myAccounts;

private JTextArea jTextArea;

private JTextField amountField;

private JComboBox<AbstractBankAccount> accountComboBox;

public JavaBank() {

myAccounts = new AbstractBankAccount[10];

jTextArea = new JTextArea();

amountField = new JTextField();

accountComboBox = new JComboBox<>();

try {

// Example account creation

myAccounts[0] = new CheckingAccount("12345", 1000.00);

myAccounts[1] = new SavingsAccount("67890", 2000.00);

for (AbstractBankAccount account : myAccounts) {

if (account != null) {

accountComboBox.addItem(account);

}

}

} catch (myException ex) {

JOptionPane.showMessageDialog(null, ex.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

JButton makeTransactionButton = new JButton("Make Transaction");

makeTransactionButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

try {

handleTransaction();

} catch (myException ex) {

JOptionPane.showMessageDialog(null, ex.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

}

});

// Add components to the GUI and set up layout (not shown)

}

private void handleTransaction() throws myException {

AbstractBankAccount selectedAccount = (AbstractBankAccount) accountComboBox.getSelectedItem();

if (selectedAccount == null) {

throw new myException("No account selected.");

}

double amount = Double.parseDouble(amountField.getText());

if (amount <= 0) {

throw new myException("Amount must be greater than zero.");

}

// Perform the transaction (deposit or withdrawal)

// Example: Assuming the transaction type is determined by some logic

// For demonstration, let's assume a deposit:

// selectedAccount.deposit(amount);

displayAccountDetails(selectedAccount);

}

private void displayAccountDetails(AbstractBankAccount account) {

jTextArea.setText(account.toString());

}

public static void main(String[] args) {

new JavaBank();

}

}

5. import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class JavaBank {

private AbstractBankAccount[] myAccounts;

private JTextArea jTextArea;

private JTextField amountField;

private JComboBox<AbstractBankAccount> accountComboBox;

public JavaBank() {

myAccounts = new AbstractBankAccount[10];

jTextArea = new JTextArea();

amountField = new JTextField();

accountComboBox = new JComboBox<>();

try {

// Example account creation

myAccounts[0] = new CheckingAccount("12345", 1000.00);

myAccounts[1] = new SavingsAccount("67890", 2000.00);

for (AbstractBankAccount account : myAccounts) {

if (account != null) {

accountComboBox.addItem(account);

}

}

} catch (Exception ex) {

myException newExc = new myException("An unhandled error occurred!!");

JOptionPane.showMessageDialog(null, newExc.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

JButton makeTransactionButton = new JButton("Make Transaction");

makeTransactionButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

try {

handleTransaction();

} catch (Exception ex) {

myException newExc = new myException("An unhandled error occurred!!");

JOptionPane.showMessageDialog(null, newExc.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

}

});

// Add components to the GUI and set up layout (not shown)

}

private void handleTransaction() throws myException {

AbstractBankAccount selectedAccount = (AbstractBankAccount) accountComboBox.getSelectedItem();

if (selectedAccount == null) {

throw new myException("No account selected.");

}

double amount = Double.parseDouble(amountField.getText());

if (amount <= 0) {

throw new myException("Amount must be greater than zero.");

}

// Perform the transaction (deposit or withdrawal)

// Example: Assuming the transaction type is determined by some logic

// For demonstration, let's assume a deposit:

// selectedAccount.deposit(amount);

displayAccountDetails(selectedAccount);

}

private void displayAccountDetails(AbstractBankAccount account) {

jTextArea.setText(account.toString());

}

public static void main(String[] args) {

new JavaBank();

}

}