

**Final Group Project**

**Machine Problem**

Construct a JAVA OOP program and a Java Swing GUI environment for Personal New Bank Account, that the user will Register the following information:

First Name:

Middle Name:

Last Name:

Birthdate:

Gender:

Address:

Father Name:

Mother Name:

Contact No:

Initial Deposit: 500

The program will store all the information **“NewAccount.txt”** for the storage of the bank account, the program has feature **of Deposit, Withdraw and Balance Inquiry**. The initial deposit is the constant value **500** in the registration. Account No. will start at this default value.

Note:

Use 1 method “**BankTransaction**” use the method of OOP polymorphism, inheritance, exception handling.

**Jeremiah C. Pagarigan**

**BSIT-MWA\_COM223**

**CODES:**  
import java.io.\*;

class BankAccount {

private static int accountNo = 1000;

private String firstName;

private String middleName;

private String lastName;

private String birthdate;

private String gender;

private String address;

private String fatherName;

private String motherName;

private String contactNo;

private double balance;

public BankAccount(String firstName, String middleName, String lastName, String birthdate, String gender,

String address, String fatherName, String motherName, String contactNo) {

this.firstName = firstName;

this.middleName = middleName;

this.lastName = lastName;

this.birthdate = birthdate;

this.gender = gender;

this.address = address;

this.fatherName = fatherName;

this.motherName = motherName;

this.contactNo = contactNo;

this.balance = 500.0;

accountNo++;

}

public void deposit(double amount) {

balance += amount;

}

public void withdraw(double amount) throws InsufficientBalanceException {

if (amount > balance)

throw new InsufficientBalanceException("Insufficient balance");

balance -= amount;

}

public double getBalance() {

return balance;

}

public String getAccountInfo() {

return "Account No: " + accountNo +

"\nName: " + firstName + " " + middleName + " " + lastName +

"\nBirthdate: " + birthdate +

"\nGender: " + gender +

"\nAddress: " + address +

"\nFather's Name: " + fatherName +

"\nMother's Name: " + motherName +

"\nContact No: " + contactNo +

"\nBalance: " + balance;

}

public void saveAccountInfo() {

String filename = "NewAccount.txt";

try {

FileWriter writer = new FileWriter(filename, true);

BufferedWriter bufferedWriter = new BufferedWriter(writer);

bufferedWriter.write(getAccountInfo());

bufferedWriter.newLine();

bufferedWriter.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class PersonalBankAccountGUI {

private JFrame frame;

private JTextField firstNameField;

private JTextField middleNameField;

private JTextField lastNameField;

private JTextField birthdateField;

private JTextField genderField;

private JTextField addressField;

private JTextField fatherNameField;

private JTextField motherNameField;

private JTextField contactNoField;

private JTextField depositField;

private JTextArea accountInfoArea;

private BankAccount bankAccount;

public PersonalBankAccountGUI() {

initialize();

}

private void initialize() {

frame = new JFrame();

frame.setTitle("Personal Bank Account");

frame.setBounds(100, 100, 450, 500);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.getContentPane().setLayout(null);

JLabel lblFirstName = new JLabel("First Name:");

lblFirstName.setBounds(10, 10, 80, 14);

frame.getContentPane().add(lblFirstName);

firstNameField = new JTextField();

firstNameField.setBounds(100, 10, 200, 20);

frame.getContentPane().add(firstNameField);

firstNameField.setColumns(10);

JLabel lblMiddleName = new JLabel("Middle Name:");

lblMiddleName.setBounds(10, 35, 80, 14);

frame.getContentPane().add(lblMiddleName);

middleNameField = new JTextField();

middleNameField.setBounds(100, 35, 200, 20);

frame.getContentPane().add(middleNameField);

middleNameField.setColumns(10);

JLabel lblLastName = new JLabel("Last Name:");

lblLastName.setBounds(10, 60, 80, 14);

frame.getContentPane().add(lblLastName);

lastNameField = new JTextField();

lastNameField.setBounds(100, 60, 200, 20);

frame.getContentPane().add(lastNameField);

lastNameField.setColumns(10);

JLabel lblBirthdate = new JLabel("Birthdate:");

lblBirthdate.setBounds(10, 85, 80, 14);

frame.getContentPane().add(lblBirthdate);

birthdateField = new JTextField();

birthdateField.setBounds(100, 85, 200, 20);

frame.getContentPane().add(birthdateField);

birthdateField.setColumns(10);

JLabel lblGender = new JLabel("Gender:");

lblGender.setBounds(10, 110, 80, 14);

frame.getContentPane().add(lblGender);

genderField = new JTextField();

genderField.setBounds(100, 110, 200, 20);

frame.getContentPane().add(genderField);

genderField.setColumns(10);

JLabel lblAddress = new JLabel("Address:");

lblAddress.setBounds(10, 135, 80, 14);

frame.getContentPane().add(lblAddress);

addressField = new JTextField();

addressField.setBounds(100, 135, 200, 20);

frame.getContentPane().add(addressField);

addressField.setColumns(10);

JLabel lblFatherName = new JLabel("Father's Name:");

lblFatherName.setBounds(10, 160, 87, 14);

frame.getContentPane().add(lblFatherName);

fatherNameField = new JTextField();

fatherNameField.setBounds(100, 160, 200, 20);

frame.getContentPane().add(fatherNameField);

fatherNameField.setColumns(10);

JLabel lblMotherName = new JLabel("Mother's Name:");

lblMotherName.setBounds(10, 185, 90, 14);

frame.getContentPane().add(lblMotherName);

motherNameField = new JTextField();

motherNameField.setBounds(100, 185, 200, 20);

frame.getContentPane().add(motherNameField);

motherNameField.setColumns(10);

JLabel lblContactNo = new JLabel("Contact No:");

lblContactNo.setBounds(10, 210, 80, 14);

frame.getContentPane().add(lblContactNo);

contactNoField = new JTextField();

contactNoField.setBounds(100, 210, 200, 20);

frame.getContentPane().add(contactNoField);

contactNoField.setColumns(10);

JLabel lblDeposit = new JLabel("Initial Deposit:");

lblDeposit.setBounds(10, 235, 80, 14);

frame.getContentPane().add(lblDeposit);

depositField = new JTextField();

depositField.setBounds(100, 235, 200, 20);

frame.getContentPane().add(depositField);

depositField.setColumns(10);

JButton btnRegister = new JButton("Register");

btnRegister.setBounds(10, 270, 100, 23);

frame.getContentPane().add(btnRegister);

btnRegister.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String firstName = firstNameField.getText();

String middleName = middleNameField.getText();

String lastName = lastNameField.getText();

String birthdate = birthdateField.getText();

String gender = genderField.getText();

String address = addressField.getText();

String fatherName = fatherNameField.getText();

String motherName = motherNameField.getText();

String contactNo = contactNoField.getText();

double initialDeposit = Double.parseDouble(depositField.getText());

bankAccount = new BankAccount(firstName, middleName, lastName, birthdate, gender,

address, fatherName, motherName, contactNo);

bankAccount.deposit(initialDeposit);

bankAccount.saveAccountInfo();

JOptionPane.showMessageDialog(frame, "Account registered successfully!");

clearFields();

}

});

JButton btnDeposit = new JButton("Deposit");

btnDeposit.setBounds(120, 270, 100, 23);

frame.getContentPane().add(btnDeposit);

btnDeposit.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (bankAccount == null) {

JOptionPane.showMessageDialog(frame, "Please register an account first.");

return;

}

String depositAmountString = JOptionPane.showInputDialog(frame, "Enter deposit amount:");

if (depositAmountString != null && !depositAmountString.isEmpty()) {

double depositAmount = Double.parseDouble(depositAmountString);

bankAccount.deposit(depositAmount);

JOptionPane.showMessageDialog(frame, "Deposit successful!");

} else {

JOptionPane.showMessageDialog(frame, "Invalid deposit amount!");

}

}

});

JButton btnWithdraw = new JButton("Withdraw");

btnWithdraw.setBounds(230, 270, 100, 23);

frame.getContentPane().add(btnWithdraw);

btnWithdraw.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (bankAccount == null) {

JOptionPane.showMessageDialog(frame, "Please register an account first.");

return;

}

String withdrawAmountString = JOptionPane.showInputDialog(frame, "Enter withdrawal amount:");

if (withdrawAmountString != null && !withdrawAmountString.isEmpty()) {

double withdrawAmount = Double.parseDouble(withdrawAmountString);

try {

bankAccount.withdraw(withdrawAmount);

JOptionPane.showMessageDialog(frame, "Withdrawal successful!");

} catch (InsufficientBalanceException ex) {

JOptionPane.showMessageDialog(frame, "Insufficient balance!");

}

} else {

JOptionPane.showMessageDialog(frame, "Invalid withdrawal amount!");

}

}

});

JButton btnBalance = new JButton("Balance Inquiry");

btnBalance.setBounds(10, 310, 150, 23);

frame.getContentPane().add(btnBalance);

btnBalance.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (bankAccount == null) {

JOptionPane.showMessageDialog(frame, "Please register an account first.");

return;

}

double balance = bankAccount.getBalance();

JOptionPane.showMessageDialog(frame, "Your balance is: $" + balance);

}

});

accountInfoArea = new JTextArea();

accountInfoArea.setEditable(false);

accountInfoArea.setBounds(10, 350, 400, 100);

frame.getContentPane().add(accountInfoArea);

frame.setVisible(true);

}

private void clearFields() {

firstNameField.setText("");

middleNameField.setText("");

lastNameField.setText("");

birthdateField.setText("");

genderField.setText("");

addressField.setText("");

fatherNameField.setText("");

motherNameField.setText("");

contactNoField.setText("");

depositField.setText("");

}

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

new PersonalBankAccountGUI();

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

}

class InsufficientBalanceException extends Exception {

public InsufficientBalanceException(String message) {

super(message);

}

}

A screenshot of a computer

Description automatically generated**THE RESULT:**  
1. Here is the main screen when you run the code

A screenshot of a computer

Description automatically generated2. Here is the screen when you register a new account, there is a 500 pesos deposit when you register a new account to the bank.

A screen shot of a computer screen

Description automatically generated with low confidenceA screenshot of a computer

Description automatically generated3. Here is the screen when you want to deposit in your new account, and you will see the successful when you click OK button

A screen shot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated4. Here is the screen when you want to withdraw in your new account, and you will see the successful when you click OK button

5. And when you want to see your balance into your account, just click the balance inquiry and you will see your balance to your bank account.

A screenshot of a computer

Description automatically generated

6. Lastly, the program will store all the information **“NewAccount.txt”** for the storage of the bank account.

A screenshot of a computer

Description automatically generated with medium confidence