Programming course

# **INTRODUCTION**

This Coursework is a continuation of the coursework assigned during the previous semester. The previous coursework was aimed at introducing and making students comfortable in basic Object-Oriented Programming concepts such as Classes, Methods, Constructors etc. The current coursework takes that a step further by making us add a user-friendly GUI(Graphical User Interface) on top of the previous project. It is aimed at making students comfortable with designing basic GUI interfaces, Event Handling, Object Casting, Importing Java packages, Abstraction and Implementation and a more advance use of constructors.

We have been instructed to create an additional class in the previous project, which handles all the GUI and Event Handling. In total we will have four classes in this project. BankCard, DebitCard, CreditCard and BankCardGUI. The BankCardGUI class calls the appropriate methods from DebitCard and CredtCard classes through Object Casting and passes the parameters if any. All the work relating to GUI has been done inside the BankCardGUI constructor, which is then later used to create a new object through the main method. Event Handling was done by implementing the ActionListener abstract class and overriding the ActionListener method. More details about how various methods were implemented have been given throughout this report.

The tools used in this project were similar to the ones used in the previous project.

* JDK: Java Development Kit is a software development environment for developing applications using java (GeeksForGeeks, 2022). JDK is an absolute for any kind of java programming
* JRE: Java Runtime Environment provides environment for java code to run. Unlike JDK, JRE provide a development environment (GeeksForGeeks, 2022).
* JVM: Java Virtual Machine executes the compiled byte code line by line and runs the java program (GeeksForGeeks, 2022).
* BlueJ: BlueJ is a development environment for Java programs, created by the Kings College, London (BlueJ, 2023). It is a very good entry point to Java IDEs for students and people just starting out on Java Programming
* DrawIO: Draw.io is a website used to create UML(Unified Modal Language) diagrams. This website was used to create the class diagrams for all four classes.
* MS-Word: MS-Word is a word processing software developed by the Microsoft Corporation (Byju's, 2020). It was used to develop this report.
* Github: Github is a code hosting platform with version control using Git (GitHub, 2023). It was used to save any major changes periodically to prevent any data loss incase of any hardware or software failures.

# Class Diagrams

A class diagram is a static structure diagram that describes the structure of a system graphically. It shows the systems classes, attributes, methods etc. and its relationship with other classes and/or objects (Visual Paradigm, 2022). Class diagrams contain a graphical representation of all the attributes, methods, and constructors within a certain class. It is used in designing and modelling software to describe classes and their relationships (Agile Education Research group, University of Helsinki, 2019). Class diagrams have a set of standards that needs to be followed, this helps us to quickly understand basic things about the source code. It also helps us understand the relationship between the various classes.

A minus sign( - ) before an attribute or method denotes that, the said attribute or method is private whereas a plus sign( + ) denotes the method or attribute to be public. The data type of an attribute is written at the end of its name, separated by a colon ( : ). Constructors are denoted using the <<constructor>> symbol. The parameters of methods and constructors are given within the parenthesis along with their data types. At last, the return type of the method is denoted at the end. Relationships between various classes are denoted by arrows, dotted lines and a method is underlined with dotted lines to denote that it is a static method.

## BankCard Class

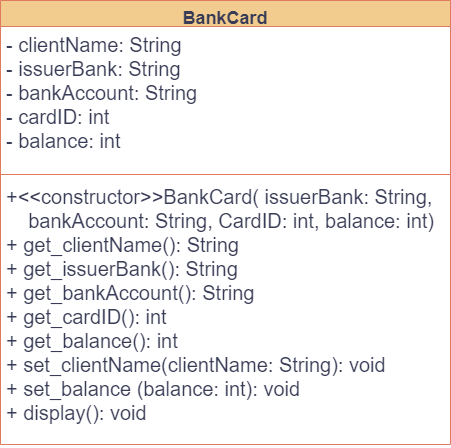


Figure : Bank Card Class Diagram

## Debit Card Class

A screenshot of a computer code

Description automatically generated with low confidence

Figure : Debit Card Class Diagram

## Credit Card Class Diagram

A screenshot of a computer program

Description automatically generated with medium confidence

Figure : Credit Card Class Diagram

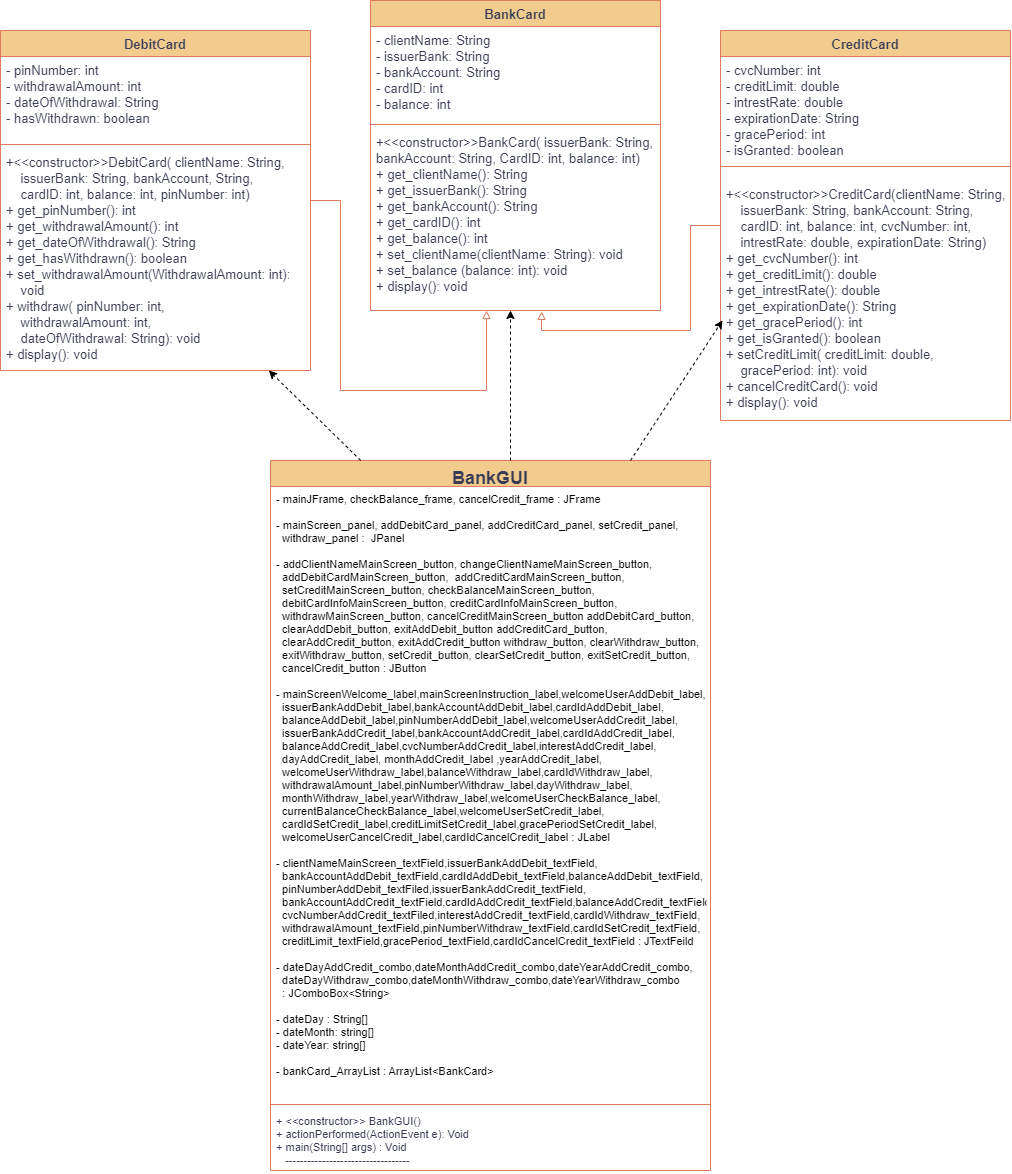
## BankGUI Class Diagram

A picture containing screenshot, text

Description automatically generated

Figure : BankGUI Class Diagram

## Combined Class Diagram



Dependency Relationship

Figure : Combined Class Diagram

The BankCard Class is the parent class of DebitCard and CreditCard class. BankGUI class has no inheritance relationship but has a dependency relationship between the other three classes as shown by the dotted line. Dependency is a weak relationship between two classes (Class Draw, 2008).

# PseudoCode for BankGUI Class

**IMPORT** the java swing class

**IMPORT** java awt Event class

**IMPORT** the java awt color class

**IMPORT** the java awt font class

**IMPORT** the java util ArrayList class

**CREATE** the BankGUI class

**DECLARE** mainJFrame, checkBalance\_frame and cancelCredit\_frame

**JFrames** with private access modifiers.

**DECLARE** JPanels with private access modifiers

**DECLARE** the required JButtons with private access modifiers

**DECLARE** the required JLabels with private access modifiers

**DECLARE** the required JTextFields with private access modifiers

**DECLARE** JComboBox that holds the String data type with private access modifiers

**DECLARE** and **INITIALIZE** three array’s that store String data type for storing the day, month, and year for the combo box with private access modifiers.

**DECLARE**  a ArrayList bankcard\_ArrayList that stores objects of the BankCard class with private access

**DECLARE** and **INITIALIZE** all the required icons with private access

**CREATE** a **CONSTRUCTOR** for BankGUI class with no parameters

**DO**

**INITIALIZE** the bankcard\_ArrayList

**INITIALIZE** the required JFrames

**INITIALIZE** the required Fonts

**INITIALIZE** all the required JLabel components with their appropriate texts

**SET** the font of all the JLabel components according to their use

**INITIALIZE** all the required JButton components with the appropriate label and icons

**ADD** action listener to all the JButtons

**INITIALIZE** all the required JTextFields

**INITIALIZE** the combo-box for storing date for withdrawal and credit card that displays the dates from previously initialized arrays

**POSITION** all the JLabel components

**POSITION** all the JTextField components

**POSITION** all the JButton components

**POSITION** all the JComboBox components

**SET** the **visibility** of change user button to false by default

**ADD** the respective components to the mainScreen\_panel

**ADD** the respective components to the addDebitCard\_panel

**ADD** the respective components to the addCreditCard\_panel

**ADD** the respective components to the withdraw\_panel

**ADD** the respective components to the setCredit\_panel

**ADD** the respective components to the checkBalance\_frame

**ADD** the respective components to the cancelCredit\_frame

**SET** the **layout** of mainScreen\_panel to null

**SET** the **size** ofmainScreen\_panel

**SET** the **visibility** of mainScreen\_panel to **true**

**SET** the **layout** of addDebitCard\_panel to null

**SET** the **size** ofaddDebitCard \_panel

**SET** the **visibility** of addDebitCard \_panel to **false**

**SET** the **layout** of addCreditCard\_panel to null

**SET** the **size** ofaddCreditCard \_panel

**SET** the **visibility** of addCreditCard \_panel to **false**

**SET** the **layout** of setCredit\_panel to null

**SET** the **size** ofsetCredit \_panel

**SET** the **visibility** of setCredit\_panel to **false**

**SET** the **layout** of withdraw\_panel to null

**SET** the **size** ofwithdraw\_panel

**SET** the **visibility** of withdraw\_panel to **false**

**ADD** mainScreen\_panel to mainJFrame

**ADD** addDebitCard\_panel to mainJFrame

**ADD** addCreditCard\_panel to mainJFrame

**ADD** setCredit\_panel to mainJFrame

**ADD** withdraw\_panel to mainJFrame

**ADD** setCredit\_panel to mainJFrame

**SET** the default close operation for mainJFrame

**SET** the layout of mainJFrame to **null**

**SET** the size of mainJFrame

**SET** the relative location of mainJFrame to **null**

**SET** the resizable value of mainJFrame to **false**

**SET** the visibility of mainJFrame to **true**

**SET** the layout of checkBalance\_frame to **null**

**SET** the size of checkBalance\_frame

**SET** the relative location of checkBalance\_frame to **null**

**SET** the resizable value of checkBalance\_frame to **false**

**SET** the visibility of checkBalance\_frame to **false**

**SET** the layout of cancelCredit\_frame to **null**

**SET** the size of cancelCredit \_frame

**SET** the relative location of cancelCredit \_frame to **null**

**SET** the resizable value of cancelCredit \_frame to **false**

**SET** the visibility of cancelCredit\_frame to **false**

**END**

**CREATE** a public method actionPerformed with the ActionEvent e parameter with no return type

**DO**

**INITIALIZE** a string named getClientName that stores the name written in the ClientName\_textField

**IF** addClientNameMainScreen\_button is pressed

**IF** clientName is empty

**DISPLAY** an error message

**ELSE**

**SET** the client’s name for all the panels

**SET** the visibility of add client button to **false**

**SET** the visibility of instruction label to **false**

**SET** the visibility of clientName JTextField to **false**

**SET** the visibility of change client button to **true**

**ELSE IF** changeClientNameMainScreen\_button is pressed

**SET**  text of welcomeUserMainScreen\_lablel back to default

**SET** the visibility of add client button and text field to **true**

**SET** the visibility of change client name button to **false**

**SET** the value of clientName to an empty string

**ELSE IF** addDebitCardMainScreen\_button is pressed

**IF** clientName is empty

**DISPLAY** an error message

**ELSE**

**SET** the visibility of the mainScreen\_panel to **false**

**SET** the visibility of addDebitCard\_panel to **true**

**ELSE IF** setCreditLimitMainScreen\_button is pressed

**INITIALIZE** creditExists Boolean as **false**

**IF** clientName is empty

**DISPLAY** an error message

**ELSE**

**ITERATE** over BankCard object in bankcard\_ArrayList

**IF** the object is of CreditCard Class

**SET** creditExists to **true**

**IF** creditExists is **true**

**SET** the visibility of the mainScreen\_panel to **false**

**SET** the visibility of setCredit\_panel to **true**

**ELSE**

**DISPLAY** an error message

**ELSE IF** checkBalanceMainScreen\_button is pressed

**INITIALIZE** debitExists Boolean as **false**

**IF** clientName is empty

**DISPLAY** an error message

**ELSE**

**ITERATE** over BankCard object in bankcard\_ArrayList

**IF** the object is of DebitCard Class

**CAST** a BankCard object as a DebitCard object and assign it to the DebiCard obj

**SET** the text of balance amount in withdraw screen using the get\_balance method in DebitCard class

**SET** debitExists to **true**

**IF** debitExists is **false**

**Display** an error message

**Else**

**SET** check setBalance\_frame visibility to **true**

**ELSE IF** debitCardInfoMainScreen\_button is pressed

**INITIALIZE** debitExists Boolean to **false**

**IF** clientName is empty

**DISPLAY** an error message

**ELSE**

**ITERATE** over BankCard object in bankcard\_ArrayList

**IF** the object is of DebitCard Class

**CAST** a BankCard object as a DebitCard object and assign it to the DebiCard obj

**CALL** the display method from DebitCard class using the DebitCard Object

**SET** debitExists to **true**

**IF** debitExists is **false**

**DISPLAY** an error message

**ELSE IF** creditCardInfoMainScreen\_button is pressed

**INITIALIZE** cardExists Boolean to **false**

**IF** clientName is empty

**DISPLAY** an error message

**ELSE**

**ITERATE** over BankCard object in bankcard\_ArrayList

**IF** the object is of CreditCard Class

**CAST** a BankCard object as a CreditCard object and assign it to the CreditCard obj

**CALL** the display method from CreditCard class using the CrediCard Object

**SET** creditExists to **true**

**IF** creditExists is **false**

**DISPLAY** an error message

**ELSE IF** withdrawMainScreen\_button is pressed

**INITIALIZE** debitExists Boolean to **false**

**IF** clientName Is empty

**DISPLAY** an error message

**ELSE**

**ITERATE** over BankCard object in bankcard\_ArrayList

**IF** the object is of DebitCard Class

**CAST** a BankCard object as a DebitCard object and assign it to the DebiCard obj

**SET** balance label to current balance using the get balance method

**SET** debitExists to **true**

**IF** debitExists is **true**

**SET** visibility of mainScreen\_panel to **false**

**SET** visibility of withdrawScreen\_panel to **true**

**ELSE**

**DISPLAY** an error message

**ELSE IF** exitAddDebit\_button is pressed

**SET** the visibility of addDebit\_panel to **false**

**SET** the visibility of mainScreen\_panel to **true**

**ELSE IF** clearAddDebit\_button is pressed

**SET** the text of all text fields to empty

**Display** success option pane

**ELSE IF** exitAddCredit\_button is pressed

**SET** the visibility of addCredit\_panel to **false**

**SET** the visibility of mainScreen\_panel to **true**

**ELSE IF** clearAddCredit\_button is pressed

**SET** the text of all text fields to empty

**Display** success option pane

**ELSE IF** setCredit\_button is pressed

**SET** the visibility of setCredit\_panel to **false**

**SET** the visibility of mainScreen\_panel to **true**

**ELSE IF** clearSetCredit\_button is pressed

**SET** the text of all text fields to empty

**Display** success option pane

**ELSE IF** exitWithdraw\_button is pressed

**SET** the visibility of Withdraw\_panel to **false**

**SET** the visibility of mainScreen\_panel to **true**

**ELSE IF** clearWithdraw\_button is pressed

**SET** the text of all text fields to empty

**Display** success option pane

**ELSE IF** addDebitCard\_button is pressed

**TRY**

**GET** the values of issuer bank and bank account using t he getText method

**GET** the value of cardID, balanceAmount and pinNumber using the getText method and convert it to integer using the parseInt method

**SET** the value of debitExists Boolean as **false**

**IF** any fields are empty

**DISPLAY** an error message

**ELSE**

**ITERATE** over BankCard object in bankcard\_ArrayList

**IF** the object is of DebitCard Class

**DISPLAY** debit exists error

**SET** debitExists to **true**

**IF** debitExists is false

**CREATE** a new debit card object and call the DebitCard constructor

**ADD** the object to the ArrayList

**DISPLAY** success dialog

**SET** addDebitCard\_panel visible to **false**

**SET** mainScreen\_panel visible to **true**

**SET** debitExists to **true**

**DISPLAY** a success message

**CATCH** any numberFormatException

**DISPLAY** an error message

**ELSE IF** withdraw\_button is pressed

**TRY**

**GET** the values of cardId, withdrawalAmount and pinNumber from their respective textFields and convert them to integer using pareInt

**GET** the selected day, month and year values from the combo box and concatenate it to a single string date

**INITIALIZE** a DebitCard object debitCard\_obj as **null**

**ITERATE** over BankCard object in bankcard\_ArrayList

**IF** the object is of DebitCard Class **and** user input cardId is equal to the DebitCard cardId

**CAST** a BankCard object as a DebitCard object and assign it to the DebiCard obj

**IF** the debitCard\_obj is **null**

**IF** user input pin number is equal to DebitCard pin number **and** withdrawal amount is less than or equal to balance amount

**CALL** the withdraw method using debitCard\_obj

**DISPLAY** the success message

**ELSE**

**DISPLAY** an error message

**ELSE**

**DISPLAY** debit not found error

**CATCH** any numberFormatError

**DISPLAY** an error message

**ELSE IF** cancelCreditMainScreen\_button is pressed

**IF** clientName is empty

**DISPLAY** an error message

**ELSE**

**SET** the visibility of cancelCredit\_frame to **true**

**ELSE IF** addCreditCard\_button

**TRY**

**GET** the values of issuerBank and bank account from their text fields

**GET** the values of card Id, balance amount and CVC number from their text fields and convert it to integer using the parseInt method

**GET** the value of interest rate from its text field and convert it to a double using the parseDouble method

**GET** the day, month and year from their combo box and concatenate to a single String date

**INITIALIZE** the value of Boolean creditExists as **false**

**IF** issuer bank or bank account is **empty**

**DISPLAY** an error message

**ELSE**

**ITERATE** over BankCard object in bankcard\_ArrayList

**IF** the object is of CreditCard Class and card Id is equal to the BankCard class

**DISPLAY** card already exists dialog

**SET** creditExists to **true**

**IF** creditExists is **false**

**INITIALIZE** a new credit card object and pass the required parameters to the constructor

**ADD** the object to the Array List

**DISPLAY** success dialog

**SET** the visibility of add Credit panel to **false**

**SET** the visibility of mainScreen panel to **true**

**CATCH** any numberFormatException

**DISPLAY** an error message

**ELSE IF** setCredit\_button is pressed

**INITIALIZE** the Boolean value of creditIsGranted as **false**

**TRY**

**GET** the values of cardID, credit limit and grace period from the text field and convert them to integer using parseInt

**ITERATE** over BankCard object in bankcard\_ArrayList

**IF** the object is of CreditCard Class and card Id is equal to the BankCard class

**CAST** a BankCard object as a CreditCard object and assign it to the CreditCard obj

**CALL** the setCredit method from CreditCard class using the CrediCard Object

**SET** creditIsGranted to **true**

**IF** creditIsGranted is **false**

**DISPLAY** an error messge

**ELSE**

**DISPLAY** success message

**SET** the visibility of setCredit panel to **false**

**SET** the visibility of mainScreen\_panel to **true**

**CATCH** any number format exception

**DISPLAY** an error message

**ELSE IF** cancelCredit\_button is pressed

**TRY**

**GET** the value of card id from the textfield and convert it into

Integer using the parseInt method

**INITIALIZE** a creditCard\_obj as **null**

**ITERATE** over BankCard object in bankcard\_ArrayList

**IF** the object is of CreditCard Class and card Id is equal to the BankCard class

**CAST** a BankCard object as a CreditCard object and assign it to the CreditCard obj

**IF** creditCard\_obj is not **null**

**CALL** the cancelCreditCard method

**DISPLAY** a success dialog

**SET** the visible of cancel credit frame to **false**

**ELSE**

**DISPLAY** card id not found error

**CATCH** any number format exception

**DISPLAY** an error message

**END DO**

**DECLARE** the static main method

**DO**

**CREATE** a new BankGUI object

**END DO**

**END**

# Method Descriptions

A method is collection of statements, or a block of code that does a specific task and runs only when called. We can pass data to methods, also known as parameters