CS4001NI Programming

30% Individual Coursework

2022-23 autumn

Student Name: Apil Thapa

London Met ID: 22067753

College ID: NP01CP4A220164

Group:L1C8

Assignment Due Date: Friday, January 27, 2023

Assignment Submission Date: Friday, January 27, 2023

I confirm that I understand my coursework needs to be submitted online via MySecondTeacher under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

List of figures

Table of Tables

1 Introduction:

This is a java project in which we have to create a parent class **Bank card** with two sub classes **Debit card** and **credit card**. In our Bank card class instance variables like **card Id, client name**, **issuer bank**, **bank account**, and **Balance_Amount** are used likewise In Debit card class PIN number, Withdrawal Amount, date Of **Withdrawal**, **hasWithdrawn** and In credit card class **CVC number**, **Credit Limit**, **Interest Rate**, **Expiration Date, Grace period**, **isgranted** are used .simply we have to show process in which details regarding the use of Bank card, credit and debit card using object oriented programming in java. Simply in bank card class we have details regarding bank account similarly in debit card we need to withdraw a money using withdrawal method if there is sufficient amount and if pin entered is correct with pin in debit card only likewise in credit card class we can take loan from the bank and can maintain credit rate in case if we want to cancel our credit card we can but certain conditions need to be followed up such as we need to clear all our previous interest then we can apply for cancelling credit card(ubah, july 26 2021).

2 Tools used:

ms-word

Microsoft Word is a word processing program that was first developed in 1983. The most basic (and least expensive) suites also include Microsoft PowerPoint and Microsoft Excel. Additional suites exist and include other Office programs, such as Microsoft Outlook and Skype for Business. It is included in all of the Microsoft Office application suites (Ballew, 2021)**.**

Blue-j

Blue-j is a development environment that allows you to rapidly and simply create Java apps. Its essential characteristics are: simple interactive creative portable mature (gosling, 20 september 2022)**.**

3 Class diagram

A class diagram is used to represent the types of objects in a computer system and the relationships between them. A class is made up of objects, and it may inherit from other classes. It is also used to document various different aspects of the system, and to construct executable software code (booch, et al., 1994-95)**.**

Class diagram for bankcard

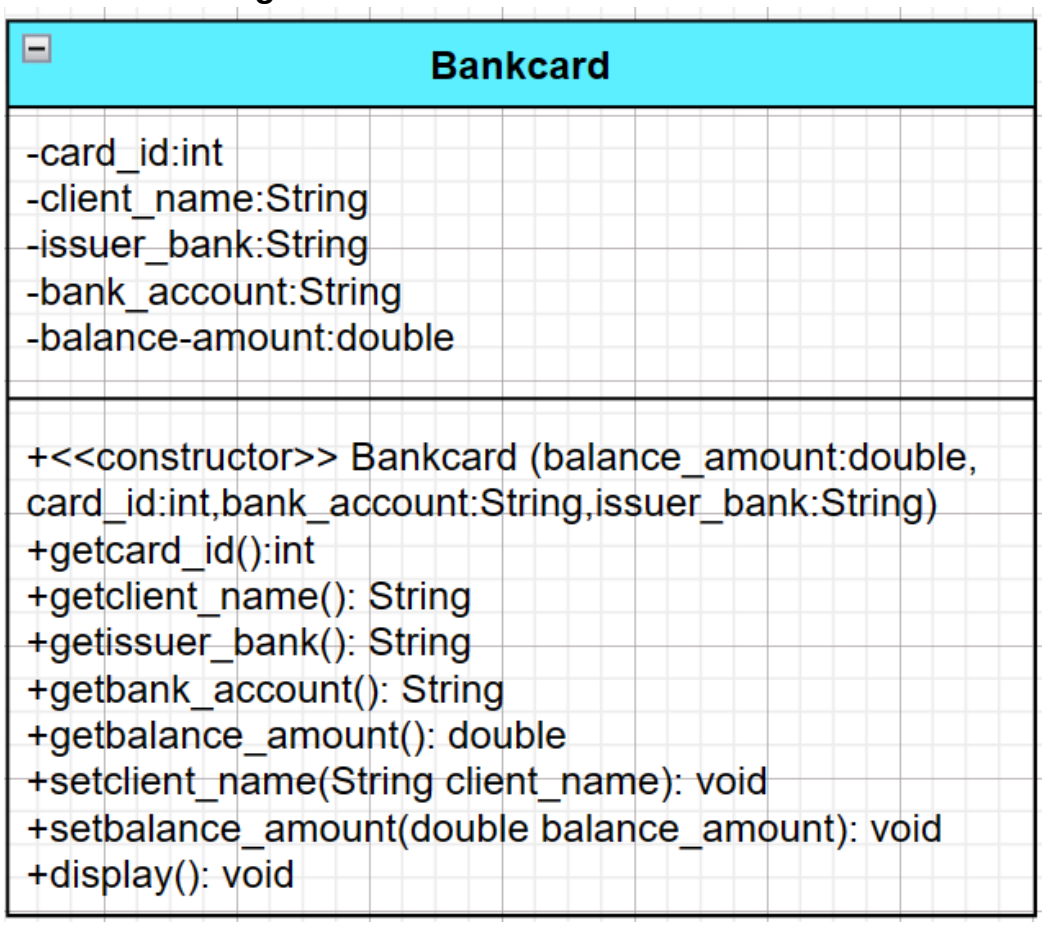


Figure 1bank card

Class diagram for debitcard

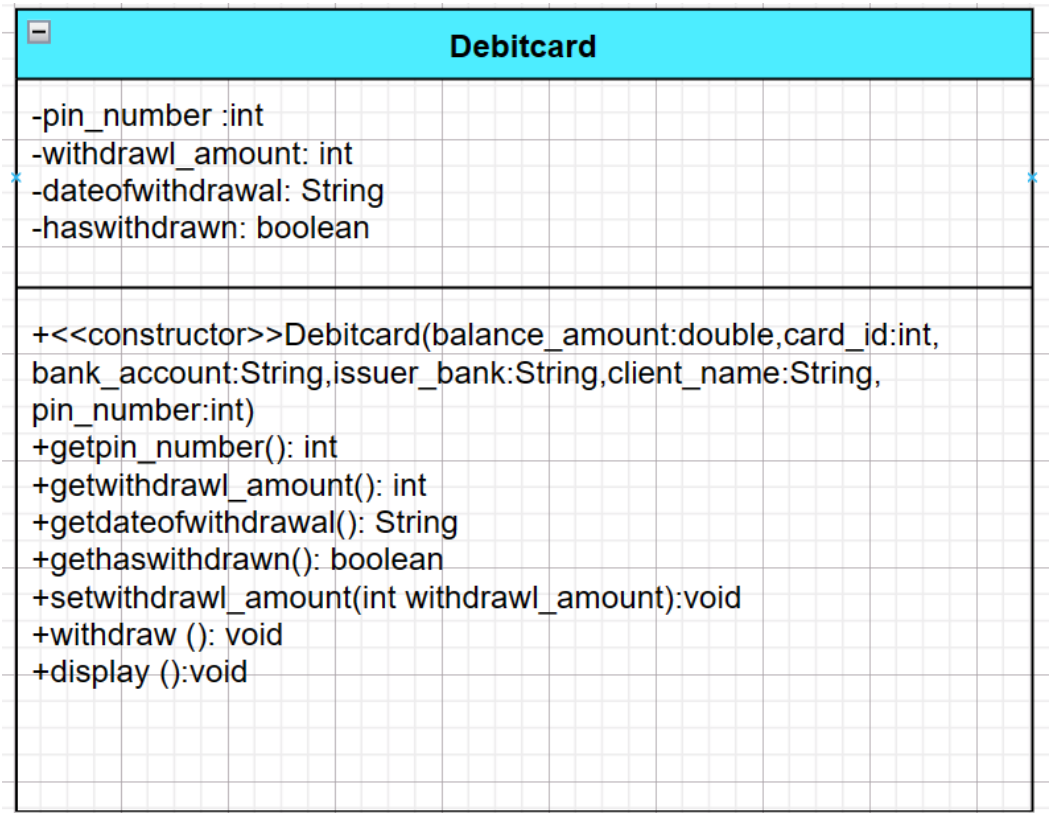


Figure 2 debit card

Class diagram for creditcard

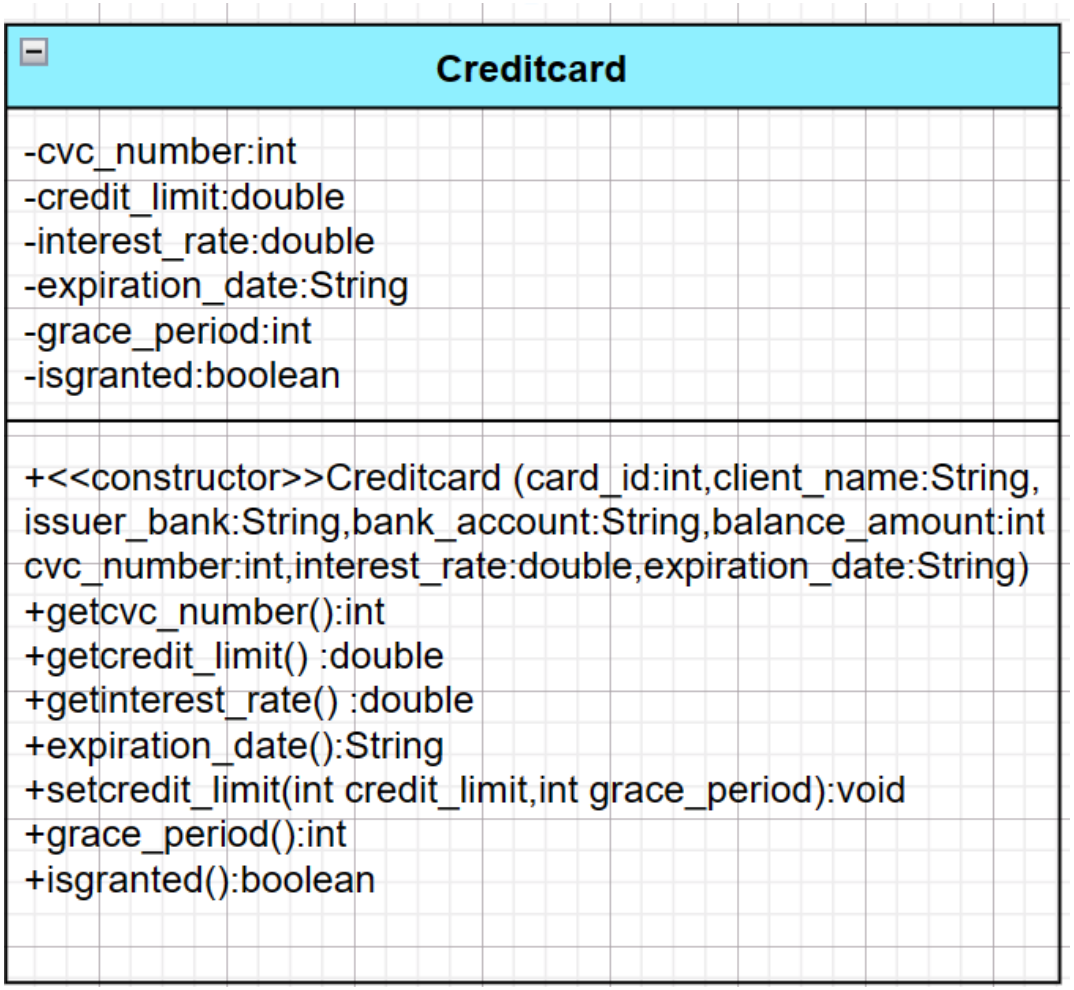


Figure 3 credit card

Class diagram for bankcard,debitcard and credit card

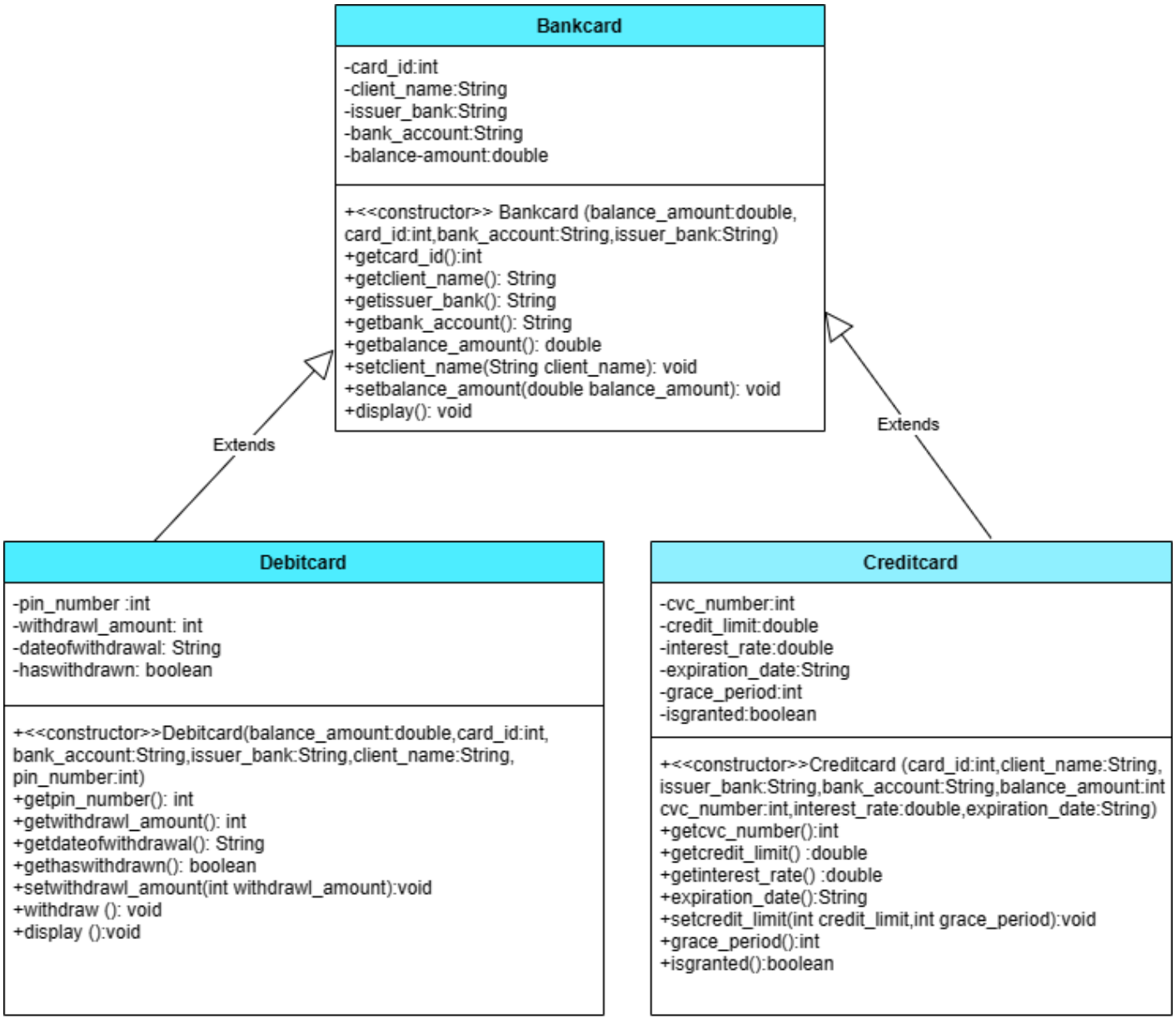

Bankcard
-card_id:int
-client_name:String
-issuer_bank:String
-bank_account:String
-balance-amount:double
+<<constructor>> Bankcard (balance_amount:double, card_id:int,bank_account:String,issuer_bank:String)
+getcard_id():int
+getclient_name(): String
+getissuer_bank(): String
+getbank_account(): String
+getbalance_amount(): double
+setclient_name(String client_name): void
+setbalance_amount(double balance_amount): void
+display(): void
Extends
Extends
Debitcard
-pin_number :int
-withdrawl_amount: int
-dateofwithdrawal: String
-haswithdrawn: boolean
+<<constructor>>Debitcard(balance_amount:double,card_id:int, bank_account:String,issuer_bank:String,client_name:String, pin_number:int)
+getpin_number(): int
+getwithdrawl_amount(): int
+getdateofwithdrawal(): String
+gethaswithdrawn(): boolean
+setwithdrawl_amount(int withdrawl_amount):void
+withdraw (): void
+display ():void
Creditcard
-cvc_number:int
-credit_limit:double
-interest_rate:double
-expiration_date:String
-grace_period:int
-isgranted:boolean
+<<constructor>>Creditcard (card_id:int,client_name:String, issuer_bank:String,bank_account:String,balance_amount:int cvc_number:int,interest_rate:double,expiration_date:String)
+getcvc_number():int
+getcredit_limit() :double
+getinterest_rate() :double
+expiration_date():String
+setcredit_limit(int credit_limit,int grace_period):void
+grace_period():int
+isgranted():boolean

Figure 4 parent class with two sub classes

4 Pseudo code:

It is a casual and artificial approach of developing programs where you convey the series of instructions and commands (also known as algorithms) in a way that is simple for people to grasp.The issue is that computers and people are quite different from one another.But with pseudocode, it works exactly the other way around. You set the guidelines. What language you choose to write your pseudocode is irrelevant (ubah, july 26 2021).

i. For Bank card class

Create a parent class Bankcard

DO

DECLARE instance variable **card_id** as **integer**

DECLARE instance variable **client_name** as **String**

DECLARE instance variable **issuer_bank** as **String**

DECLARE instance variable **bank_account** as **String**

DECLARE instance variable **balance_amount** as **double**

END DO

CREATE a constructor name Bankcard with parameter as balance_amount, card_id, bank_account and issuer_bank

DO

Initialize the value of instance variable **balance_amount**

Initialize the value of instance variable **card_id**

Initialize the value of instance variable **bank_amount**

Initialize the value of instance variable **issuer_bank**

Initialize the value of instance variable **client_name** and set it to an empty string

END DO

CREATE an accessor method **getcard_id** () with return type **integer**

DO

RETURN **card_id**

END DO

CREATE an accessor method **getclient_name ()** with return type **String**

DO

RETURN **client_name**

END DO

CREATE an accessor method **getissuer_bank ()** with return type **String**

DO

RETURN **issuer_bank**

END DO

CREATE an accessor method **getbank_account ()** with return type **String**

DO

RETURN **bank_account**

END DO

CREATE an accessor method **getbalance_amount ()** with return type **double**

DO

RETURN **balance_amount**

END DO

CREATE a mutator method **setclient_name** in which **client_name** is passed as parameter with no return type

DO

Initialize the value of instance variable **client_name**

END DO

CREATE a mutator method **setbalance_amount** in which **balance_amount** is passed as parameter with no return type

DO

Initialize the value of instance variable **balance_amount**

END DO

CREATE an instance method display with no return type

DO

IF client_name is equals to an empty **string**

print “please set **client_name”**

Else if

it will print “**card_id”**

Else if

it will print “ **client name”**

Else if

it will print “ **issuer bank”**

Else if

it will print” **bank account”.**

END DO

ii**. For Debit card class**

CREATE class Debitcard which extends Bankcard class

DO

DECLARE instance variable **pin_number** as **integer**

DECLARE instance variable **withdrawal_amount** as i**nteger**

DECLARE instance variable **dateofwithdrawal** as **String**

DECLARE instance variable **haswithdrawn** as **Boolean**

END DO

CREATE a constructor name Debitcard with parameter as **balance_amount**, **card_id**, **bank_account, issuer_bank, client_name** and **pin_number**

DO

Call constructor from parent class with parameter as **balance_amount**, **card_id**, **bank_account** and **issuer_bank**

Initialize the value of an instance_variable **pin_number**

Initialize the value of an instance_variable **haswithdrawn** which is set to false

END DO

CREATE an accessor method **getpin_number** () with return type i**nteger**

DO

RETURN **pin_number**

END DO

CREATE an accessor method **getwithdrawal_amount ()** with return type **integer**

DO

RETURN **withdrawal_amount**

END DO

CREATE an accessor method **getwithdrawal_amount ()** with return type **integer**

DO

RETURN **withdrawal_amount**

END DO

CREATE an accessor method **getdateofwithdrawal** () with return type **String**

DO

RETURN **dateofwithdrawal**

END DO

CREATE a mutator method **setwithdrawal_amount** in which **withdrawal_amount** is passed as parameter with no return type

DO

Initialize the value of an instance_variable **withdrawal_amount**

END DO

CREATE an instance method withdraw with parameter as withdrawal amount, **dateofwithdrawal** and **pin_number**

DO

If pin_number double equals to updated value of a **pin_number**

DO

if returned value of instance variable of parent class **balance_amount** is subtracted with **withdrawal_amount** which is greater than or equals to zero

DO

new value for instance variable of parent class is assign with parameter as return value of instance variable **balance_amount** subtracted by **withdrawal_amount**

initialize the value of an instance variable **withdrawal_amount**

initialize the value of an instance variable **dateofwithdrawal**

initialize the value of an instance variable **haswithdrawn**

it will print Withdrawal successful. New balance will be return value of an instance variable of parent class

Else if

It will print insufficient balance.

END DO

It will print invalid pin, please enter valid information

END DO

END DO

CREATE an instance method display

Call the parent class method called display using dot operator

DO

If haswithdrawn double equals to true

Print “pin number”.

ELSE if

Print “print **withdrawal amount”**

Else if

Print”**date of withdrawal”**

Else

Print”Transaction has not been carried out”.

iii. **for credit card class**

CREATE a public class Credit card which extends Bank card class

DO

DECLARE instance variable **cvc_number** as **integer**

DECLARE instance variable **credit_limit** as **double**

DECLARE instance variable **interest_rate** as **double**

DECLARE instance variable **expiration_date** as **string**

DECLARE instance variable **grace_period** as **integer**

DECLARE instance variable **isgranted** as **boolean**

END DO

CREATE a constructor name creditcard with parameter as **card_id, client_name, issuer_bank, bank_account ,balance_amount, cvc_number, interest_rate** and **expiration_date.**

DO

Call constructor from parent class with parameter as **balance_amount, card_id, bank_account** and **issuer_bank**

Initialize the value of an instance_variable **cvc_number**

The value of an instance variable **client_name** from parent class is assigned

Initialize the value of an instance_variable **interest_rate**

Initialize the value of an instance_variable **expiration_date**

Initialize the value of an instance_variable **isgranted** which is set to false

END DO

CREATE an accessor method **getcvc_number** () with return type **integer**

DO

RETURN **cvc_number**

END DO

CREATE an accessor method **getcredit_limit ()** with return type **double**

DO

RETURN **credit_limit**

END DO

CREATE an accessor method **getinterest_rate** () with return type **double**

DO

RETURN **withdrawl_amount**

END DO

CREATE an accessor method **getexpiration_date()** with return type **String**

DO

RETURN **expiration_date**

END DO

CREATE an accessor method **getgrace__period** () with return type **integer**

DO

RETURN **grace_period**

END DO

CREATE an accessor method **getisgranted ()** with return type **boolean**

DO

RETURN **isgranted**

END DO

CREATE a mutator method **setcreditlimit** in which parameter as **credit_limit** and **grace_period** are pass with no return type

DO

If credit_limit less than or equals to returned value of an instance variable of a parent class

Print “credit granted”

Initialize the value of an instance_variable **isgranted** which is set to true

Initialize the value of an instance_variable **credit_limit**

Initialize the value of an instance_variable **grace_period**

Else

Print”your credit cant be granted”

END DO

CREATE an instance method cancelcreditcard

DO

Cvc_number is set to zero

Grace_period is set to zero

Credit_limit is set to zero

Isgranted is set to false

END DO

CREATE an instance method display

Call the parent class method called display using dot operator

DO

If **isgranted** double equals to true

Print “**cvc_number**”.

ELSE if

Print “**credit_limit”**

Else if

Print”**grace_period**”

Else

Print”credit card is not granted”.

5 method description

Bank card class

Getcard_id()

This is a method in a Java class that returns the value of the instance variable "card_id" within the same class. The keyword "this" is used to reference the current instance of the class.

Getclient_name()

This is a method in a Java class that returns the value of the instance variable "client_name" within the same class. The keyword "this" is used to reference the current instance of the class.

Getissuerbank()

This is a method in a Java class that returns the value of the instance variable "issuer _bank " within the same class. The keyword "this" is used to reference the current instance of the class.

Getbank_account()

There is a method in a Java class that returns the value of the instance variable "bank _account " within the same class. The keyword "this" is used to reference the current instance of the class.

Getbalance_amount()

This is a method in a Java class that returns the value of the instance variable "balance _amount " within the same class. The keyword "this" is used to reference the current instance of the class.

Setclient_name(string client_name)

This is a Java method that sets the value of the instance variable "client_name" in the current object to the value passed as an argument. The keyword "this" is used to reference the current object. The method is public, meaning it can be accessed by other classes. The method takes one input argument, a String value which is assigned to the instance variable client_name.

Setbalance_amount(int balance_amount)

This is a Java method that sets the value of the instance variable "balance _amount " in the current object to the value passed as an argument. The keyword "this" is used to reference the current object. The method is public, meaning it can be accessed by other classes. The method takes one input argument, a int value which is assigned to the instance variable balance_amount.

Display()

The instance variables "card id," "client name," "issuer bank," "bank account," and "balance amount" values for the current object are shown by this Java method. The instance variable "client name" is checked before the values are displayed; if it is empty, the message "Please set client name:" is printed. Otherwise, the whole value of the instance variable is printed. The method is accessible to other classes because it is public.

Debit card class

Getpin_number()

This is a method in a Java class that returns the value of the instance variable "pin _number" within the same class. The keyword "this" is used to reference the current instance of the class.

Getwithdrawal_amount()

This is a method in a Java class that returns the value of the instance variable "withdrawal_amount " within the same class. The keyword "this" is used to reference the current instance of the class.

Getdateofwithdrawal()

This is a method in a Java class that returns the value of the instance variable " dateofwithdrawal " within the same class. The keyword "this" is used to reference the current instance of the class.

Gethaswithdrawn()

This is a method in a Java class that returns the value of the instance variable “haswithdrawn " within the same class. The keyword "this" is used to reference the current instance of the class.

Setwithdrawal_amount(int withdrawal_amount)

This is a Java method that sets the value of the instance variable "withdrawal _amount " in the current object to the value passed as an argument. The keyword "this" is used to reference the current object. The method is public, meaning it can be accessed by other classes. The method takes one input argument, a int value which is assigned to the instance variable balance_amount.

Withdraw()

This is a method to withdraw money from an account. It takes in three parameters: the withdrawal amount, the date of withdrawal, and a pin number. The method first checks if the entered pin number matches the account's pin number. If it does, it then checks if the withdrawal amount does not exceed the current balance. If it does not, the withdrawal is processed and the new balance is displayed. If the pin number is incorrect or the withdrawal exceeds the balance, an error message is displayed.

Display()

This is one way to take money out of an account. It requires three inputs: a pin number, the withdrawal amount, and the date of the transaction. The procedure initially verifies that the pin entered corresponds to the pin number associated with the account. If so, it then confirms that the withdrawal amount does not exceed the balance at hand. If not, the withdrawal is processed and the updated balance is shown. An error warning appears if the pin number is entered incorrectly or if the withdrawal amount is more than the balance.

Credit card class

Getcvc_number()

This is a method in a Java class that returns the value of the instance variable "cvc _number" within the same class. The keyword "this" is used to reference the current instance of the class.

Getcredit_limit ()

This is a method in a Java class that returns the value of the instance variable "credit _limit " within the same class. The keyword "this" is used to reference the current instance of the class.

Getinterest_rate ()

This is a method in a Java class that returns the value of the instance variable "interest _rate " within the same class. The keyword "this" is used to reference the current instance of the class.

Getexpiration_date ()

This is a method in a Java class that returns the value of the instance variable "expiration _date " within the same class. The keyword "this" is used to reference the current instance of the class.

Getgrace_period ()

This is a method in a Java class that returns the value of the instance variable "Grace _period " within the same class. The keyword "this" is used to reference the current instance of the class.

Getisgranted ()

This is a method in a Java class that returns the value of the instance variable "isgranted " within the same class. The keyword "this" is used to reference the current instance of the class.

Setcreditlimit (int credit_limit,int grace_period)

This is a method that sets a credit limit for an account and a grace period. It takes in two parameters: the credit limit and the grace period. The method checks if the requested credit limit is less than or equal to 2.5 times the current balance. If it is, it grants the credit and sets the credit limit and grace period for the account. If the credit limit is greater than 2.5 times the current balance, it prints a message saying that the credit cannot be granted.

Cancelcreditcard()

This is a method called cancelcreditlimit which cancel the credit limit in credit card class.Cvc_number grace period and credit_limit are set to zero whereas isgranted is set to false in this method.

Display()

Information about a credit card account is shown using this method. The parent class's equivalent method is called using the "super" keyword. Next, it examines the value of the isgranted variable to see if the credit card has been approved. The CVC number, Credit Limit, and Grace Period are printed if it is. A notice stating that the credit card is not approved is printed if not.

6 Testing

Table 1-Test 1:To inspect debit card class ,withdraw amount and re-inspect debit card class

Test No	1
Objective:	To inspect debit card class,withdraw amount and re-inspect debit card class:
Action:	➡Debit card class is called with following arguments: Balance_amount:5000 Card_id:12345 Bank_account:”current” Issuer bank:”nic asia” Client_name:”apil” Pin_number:9090 ➡ Inspection of the debitcard class. ➡ Void withdraw is called with following arguments: Withdrawal_amount:2000 Dateofwithdrawal:”2023/04/12” Pin_number:9090 ➡ Re- Inspection of the debitcard class.
Expected result:	Money is withdrawn or withdrawal amount is deducted from initial balance amount.
Actual result:	Money is successfully withdrawn.

Conclusion:	The test is successful.

Figure 5To inspect debit card class ,withdraw amount and re-inspect debit card class

Output result:

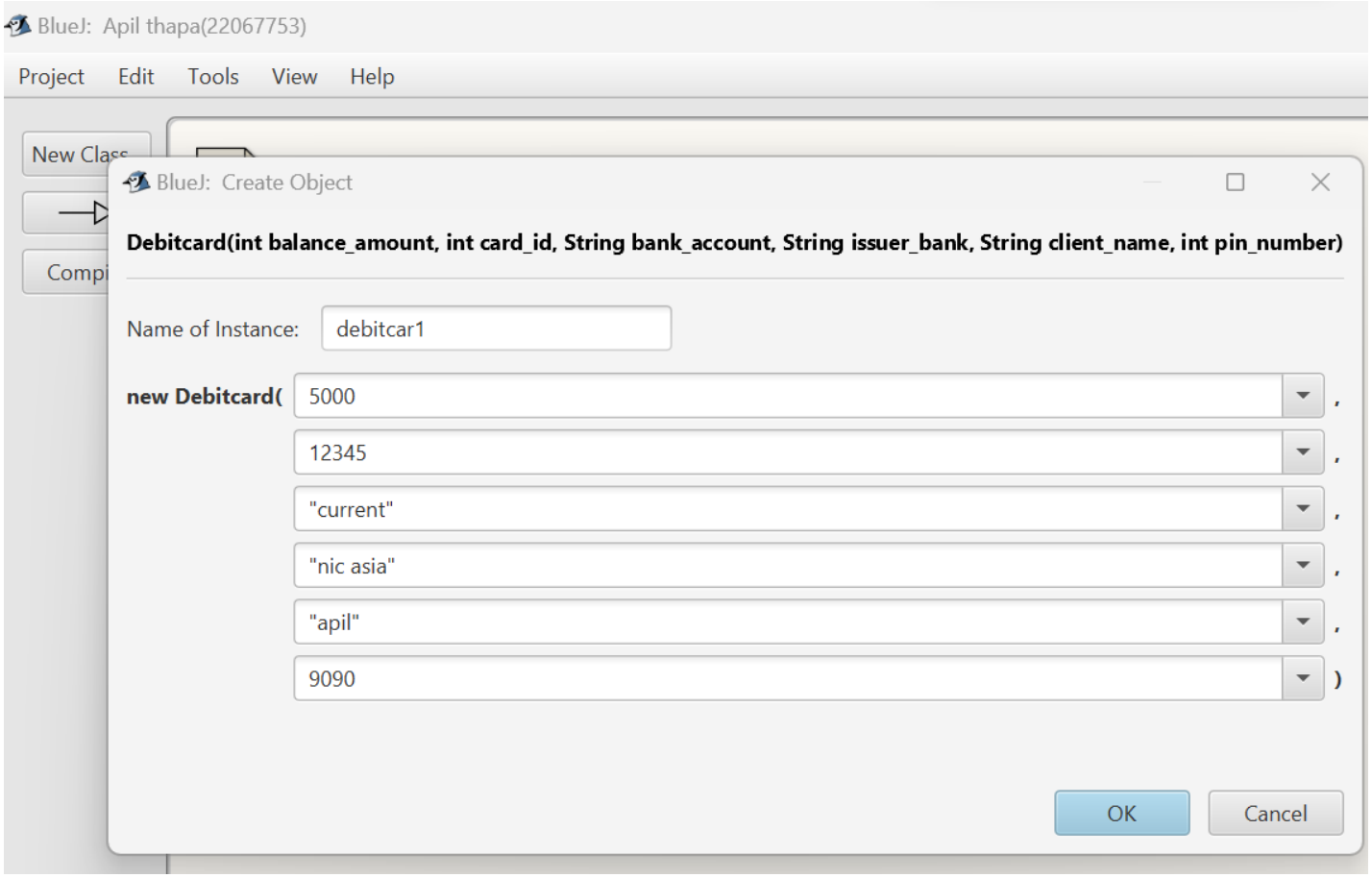


Figure 62screenshot of assigning the data in debit card class

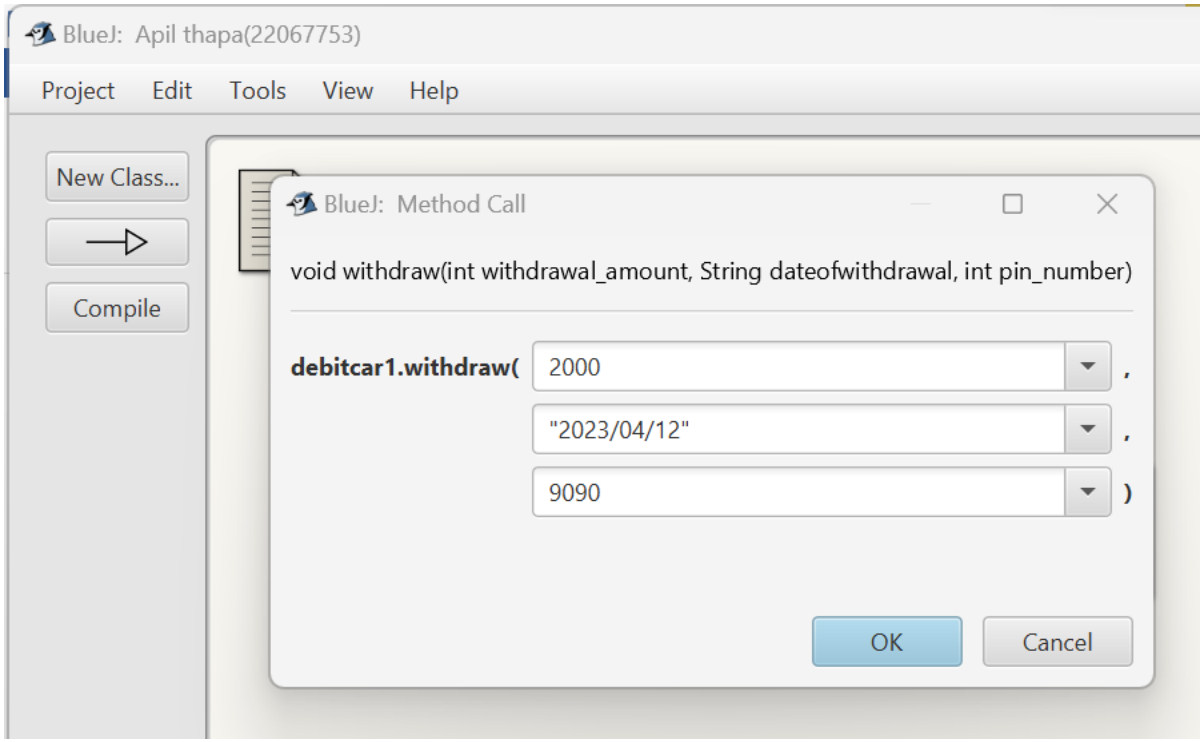


Figure 7screenshot of assigning values ofnwithdrawl amount,dateofwithdrawal and pin number

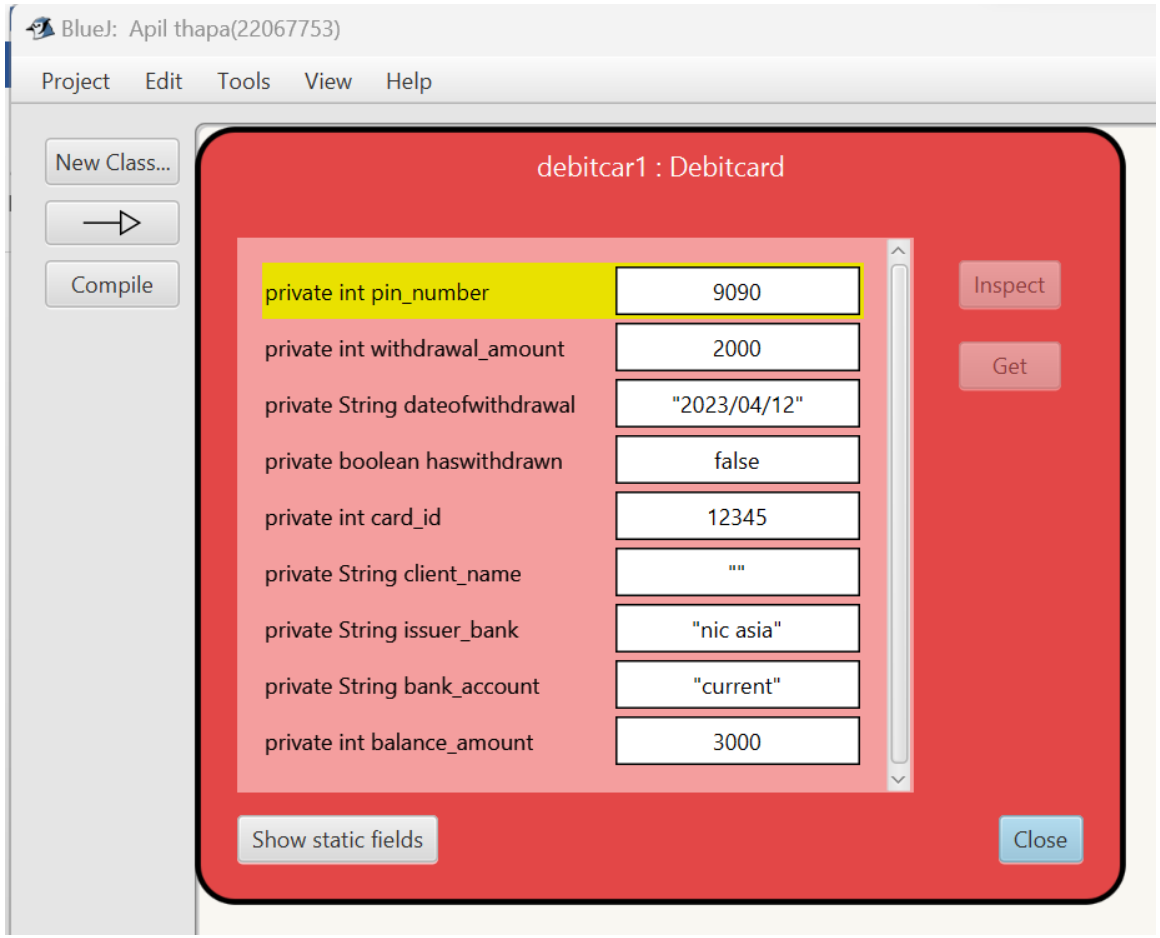


Figure 8screenshot of inspection of debit card class after withdrawing amount

Table 2-Test 2: To Inspect Credit Card class, set the credit limit and reinspect the Credit Card class

Test No:	2
Objective:	To Inspect Credit Card class, set the credit limit and reinspect the Credit Card class
Action:	➡ Credit card class is called with the following arguments: Card_id:56789 Client_name:”apil” Issuer_bank:”ncc” Bank_account:”saving” Balance_amount:1000 Cvc_number:211 Interest_rate:2.3 Expiration_date:”2023/12/03” ➡ Inspection of the creditcard class. Void setcreditlimit is called with the following arguments: Credit_limit:21 Grace_period:12 ➡ Re-inspection of credit_card class

Expected result:	Credit_limit is set
Actual result:	Credit _limit is set correctly.
Conclusion:	The test is successful.

Output result:

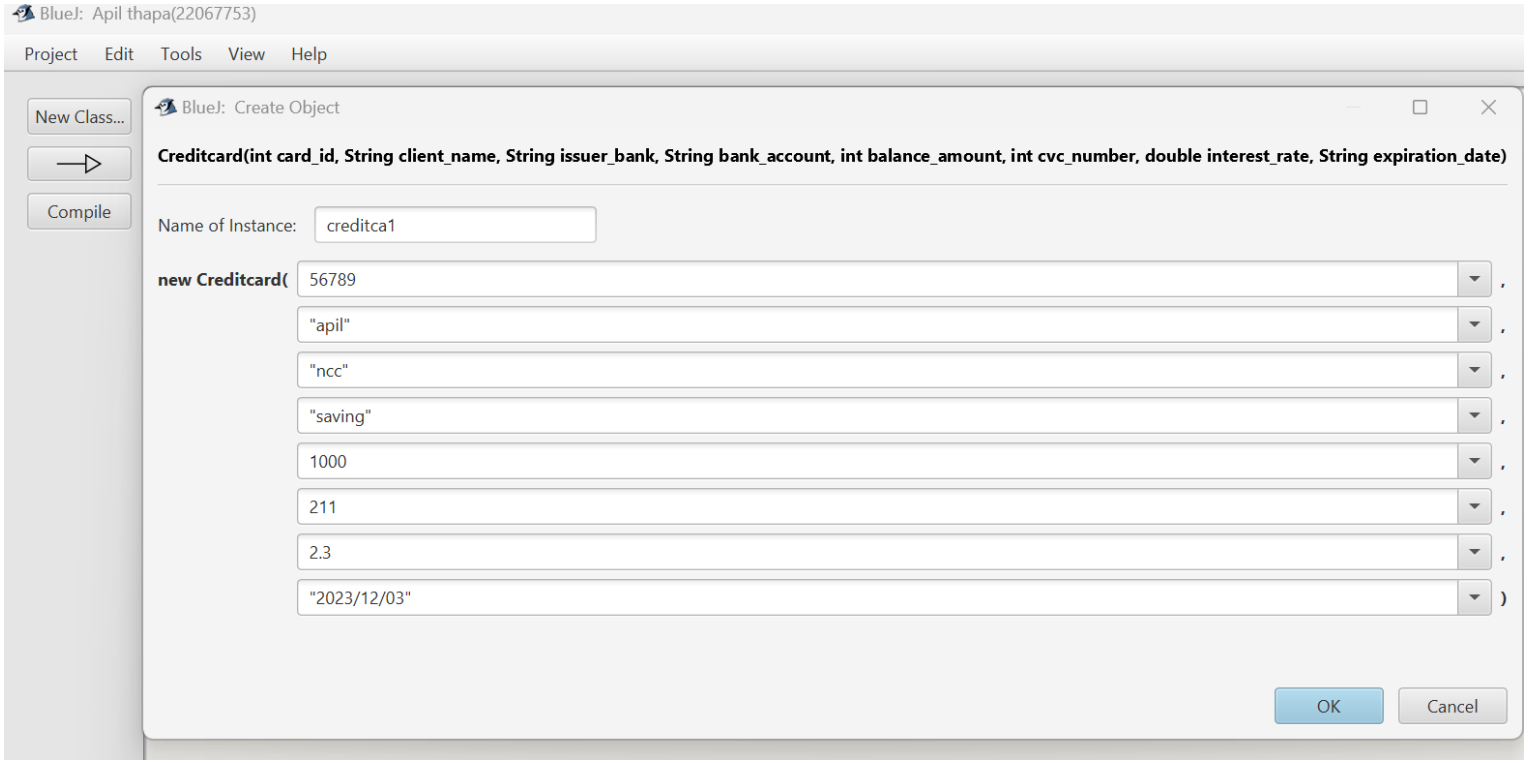


Figure 9 screenshot of assigning all value in credit card class

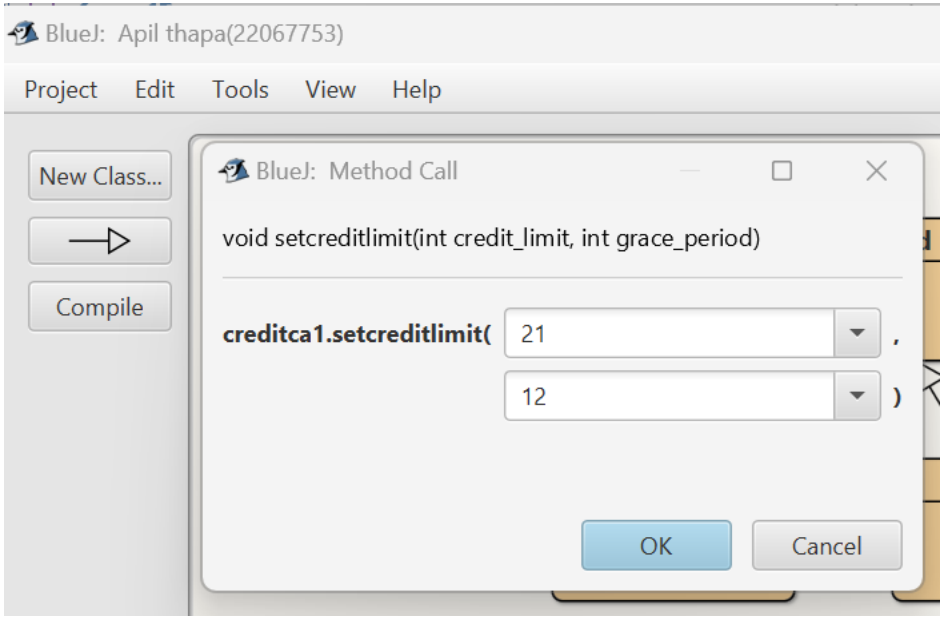


Figure 10 screenshot of assigning credit_limit and grace_period

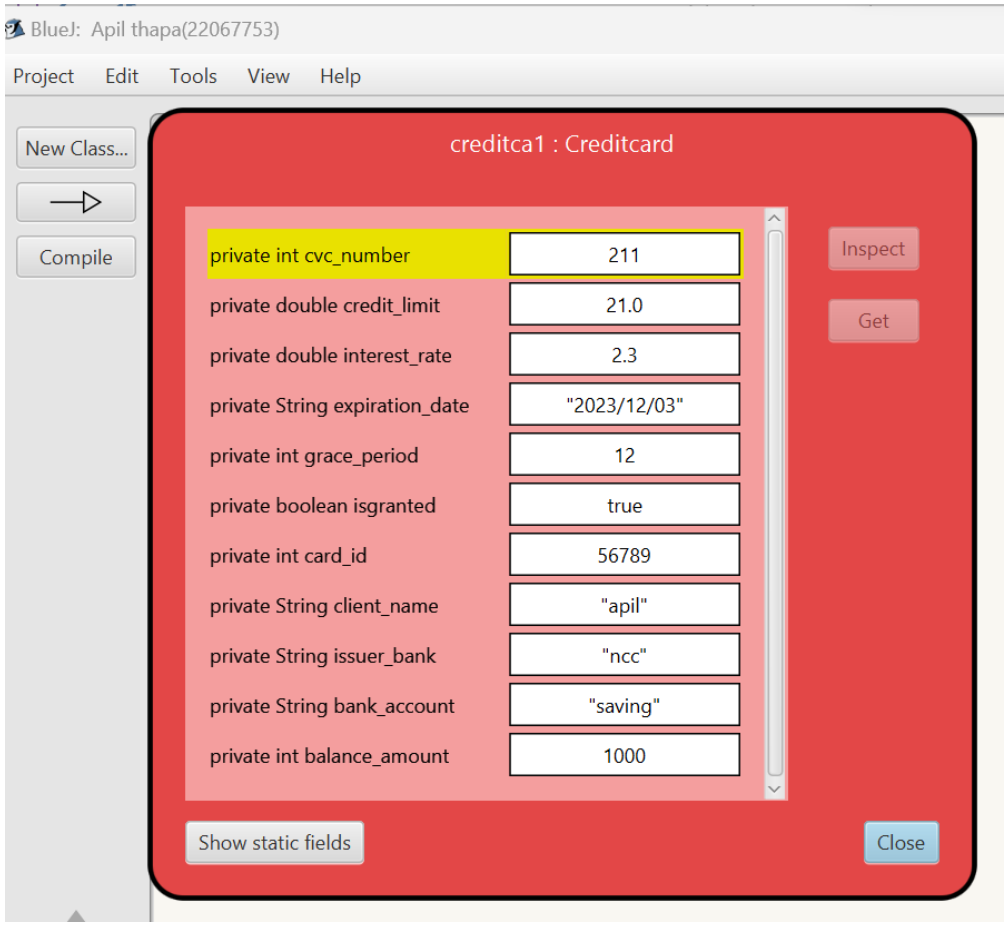


Figure 11 screenshot of inspection of credit card class after setting of credit limit.

Table 3-Test 3: To Inspect Credit Card class again after cancelling the credit card.

Test No:	3
Objective:	To Inspect Credit Card class again after cancelling the credit card.
Action:	➡ Credit card class is called with the following arguments: Card_id:56789 Client_name:”apil” Issuer_bank:”ncc” Bank_account:”saving” Balance_amount:1000 Cvc_number:211 Interest_rate:2.3 Expiration_date:”2023/12/03” ➡ Inspection of the creditcard class. Void setcreditlimit is called with the following arguments: Credit_limit:21 Grace_period:12 ➡ Inspection of the creditcard class again. Void cancelcreditcard is clicked which cancel the credit card.

Expected result:	Credit card is cancelled out.
Actual result:	Credit card is cancelled.
Conclusion:	Thes test is successful.

Output result:

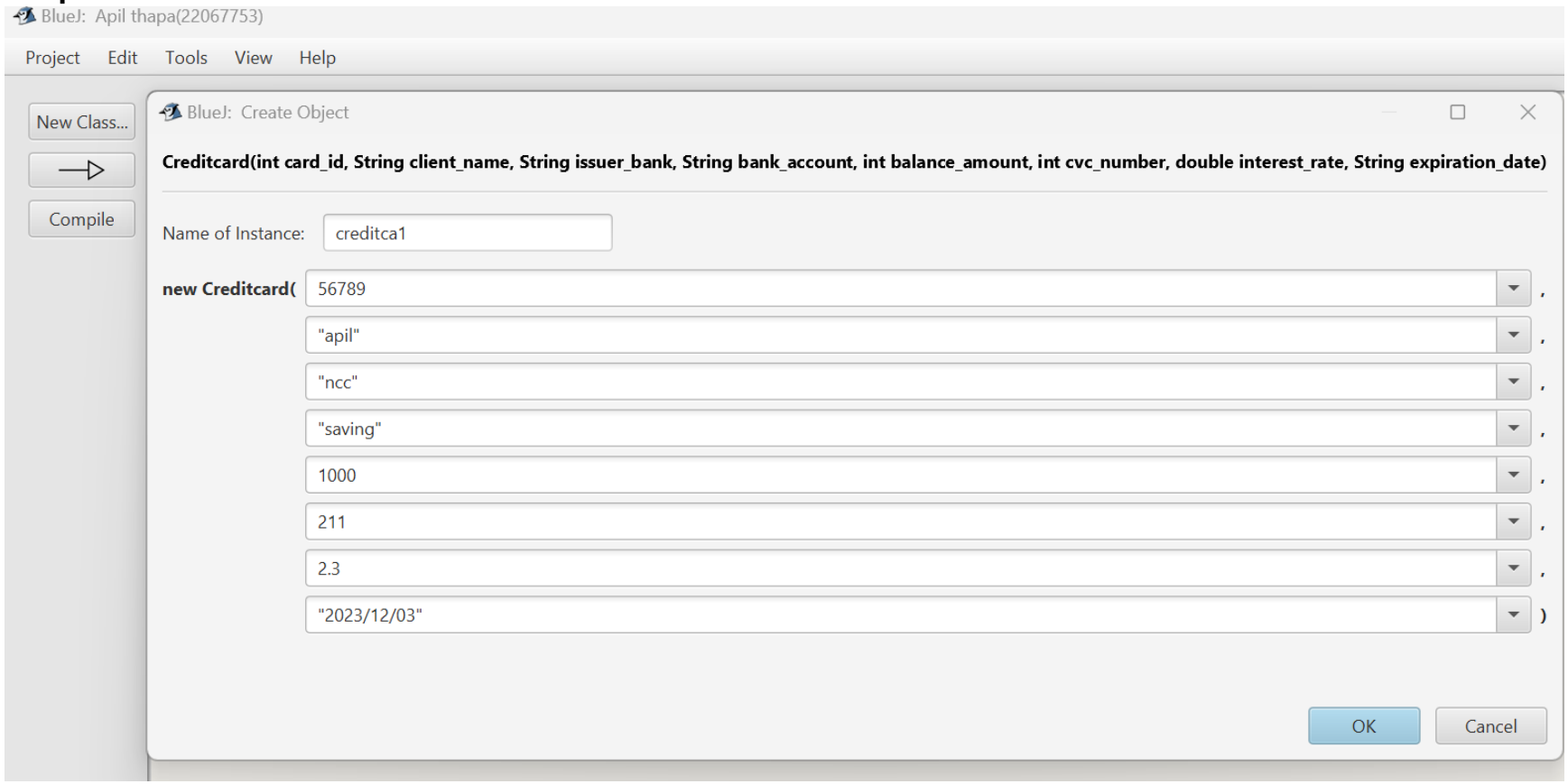


Figure 12screenshot of the value/data assigning in credit card class

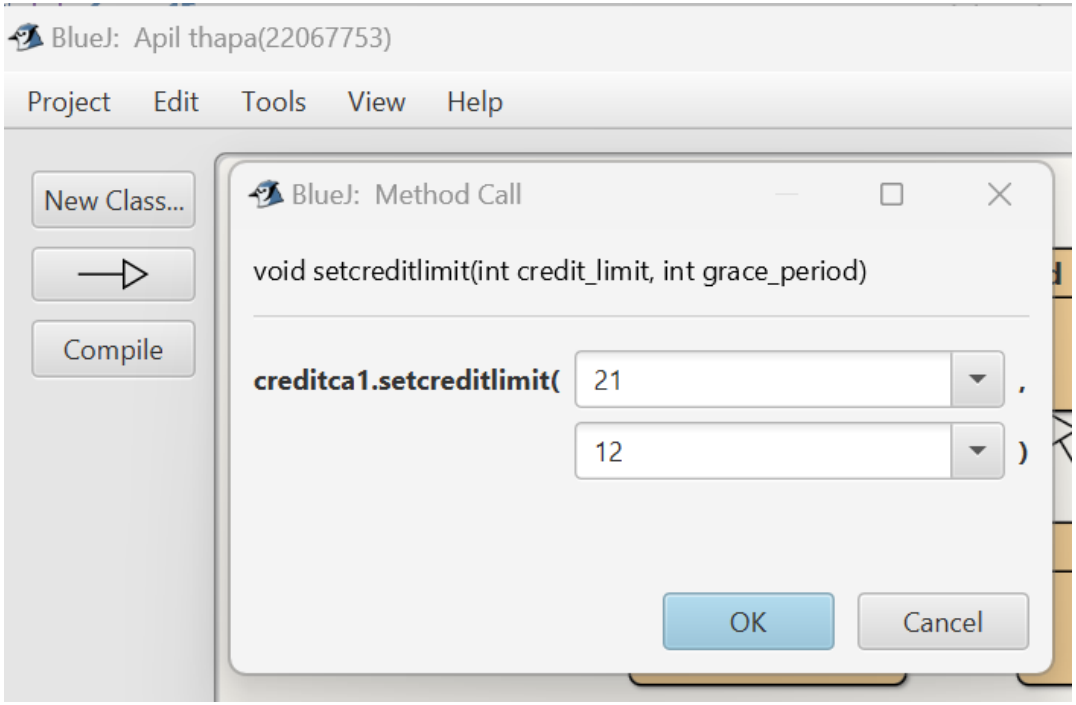


Figure 13screenshot of the set value of credit_limit and grace_period in creditcard class

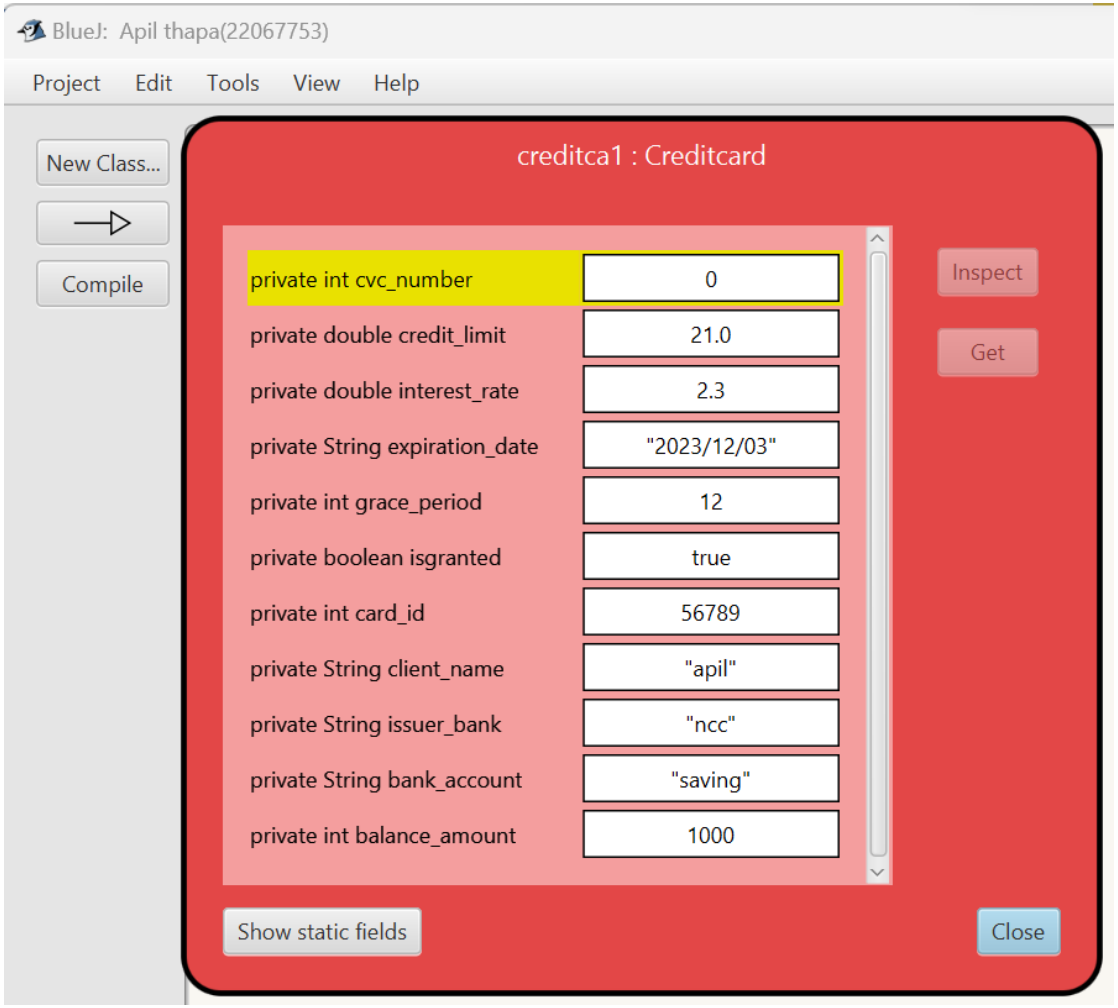


Figure 14screenshot of inspection of credit card class after setting all the value

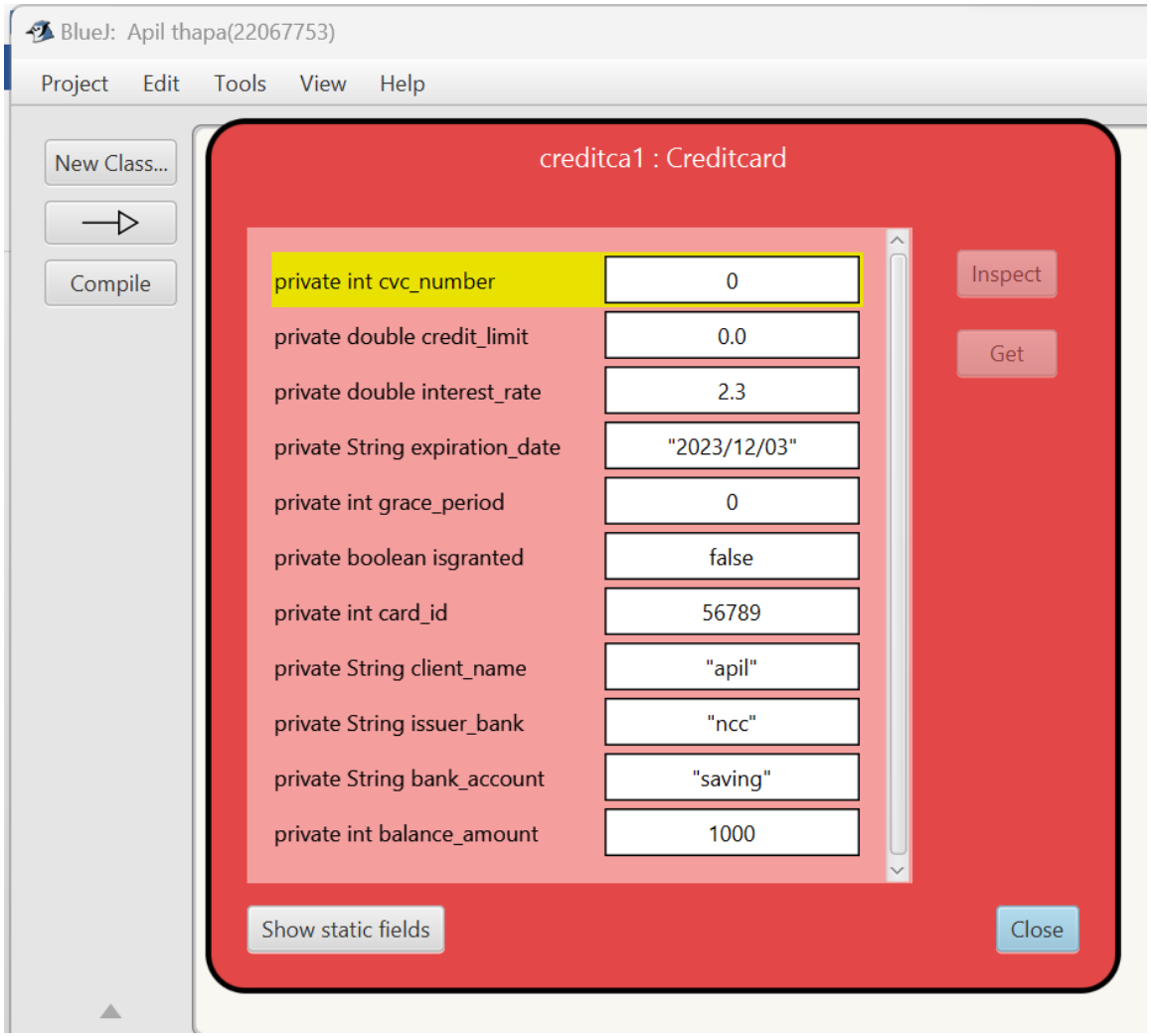


Figure 15screenshot of the value after cancelling credit card.

Table 4-Test 4: To Display the details of Debit Card and Credit Card classes.

Test No:	4
Objective:	To Display the details of Debit Card and Credit Card classes.
Action:	➡ Debit card class is called with following arguments: Balance_amount:5000 Card_id:12345 Bank_account:”current” Issuer bank:”nic asia” Client_name:”apil” Pin_number:9090 ➡ Inspection of debit card class. Void display is called. Certain suitable message is displayed saying :”please set client name first;transaction has not been carried out” ➡ Credit card class is called with the following arguments: Card_id:56789 Client_name:”apil” Issuer_bank:”ncc” Bank_account:”saving”

	Balance_amount:1000 Cvc_number:211 Interest_rate:2.3 Expiration_date:”2023/12/03” ➡ Inspection of the creditcard class. Void display is called. All message are displayed with suitable annotation.
Expected result:	Display method displayed in both classes.
Actual result:	Display method displayed in both classes.
Conclusion:	The test is successful.

Output results:

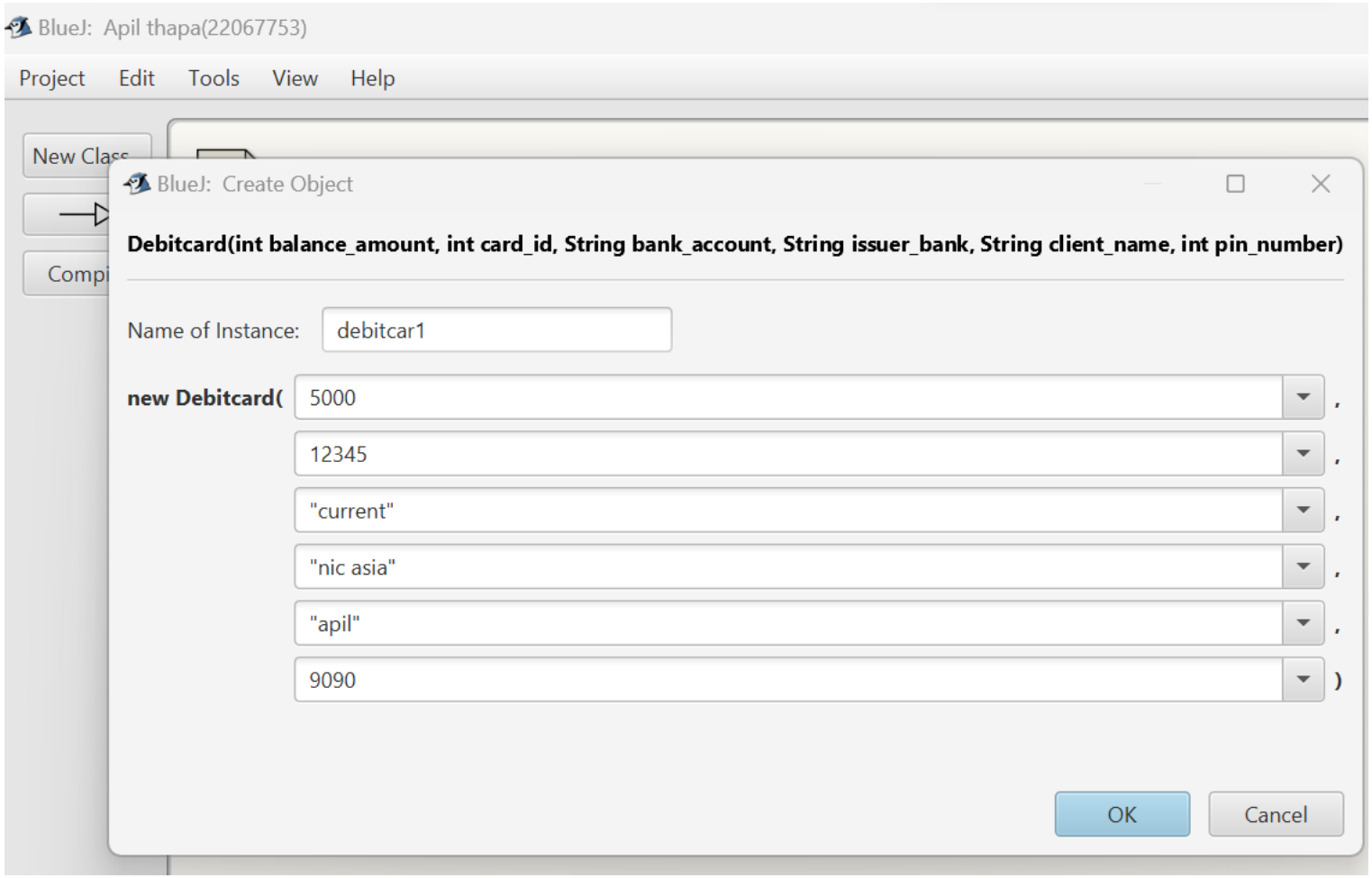


Figure 16screenshot for assigning data in debitcard class

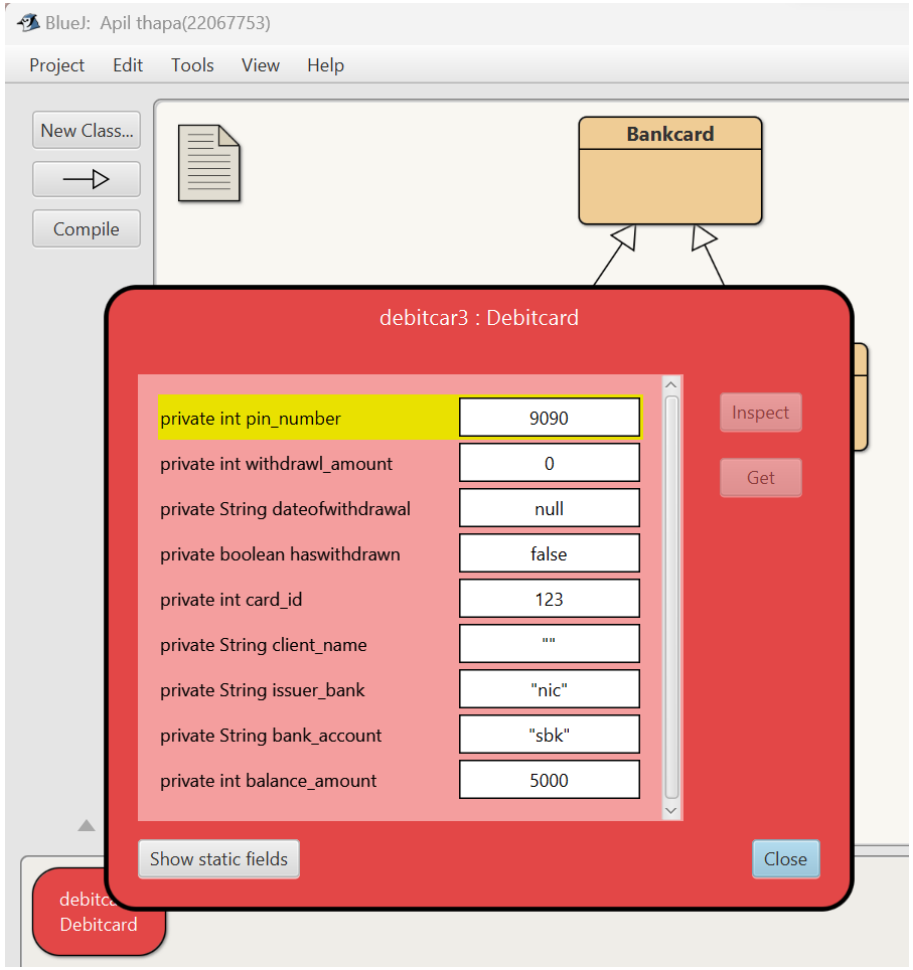
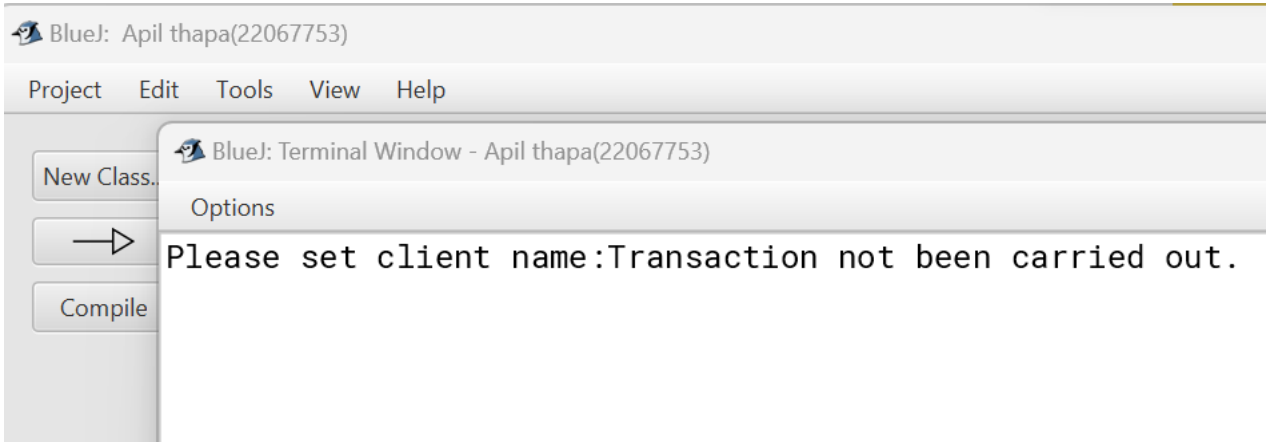


Figure 17screenshot for inspection of debit card class

Figure 18screenshot of method display of debit card class



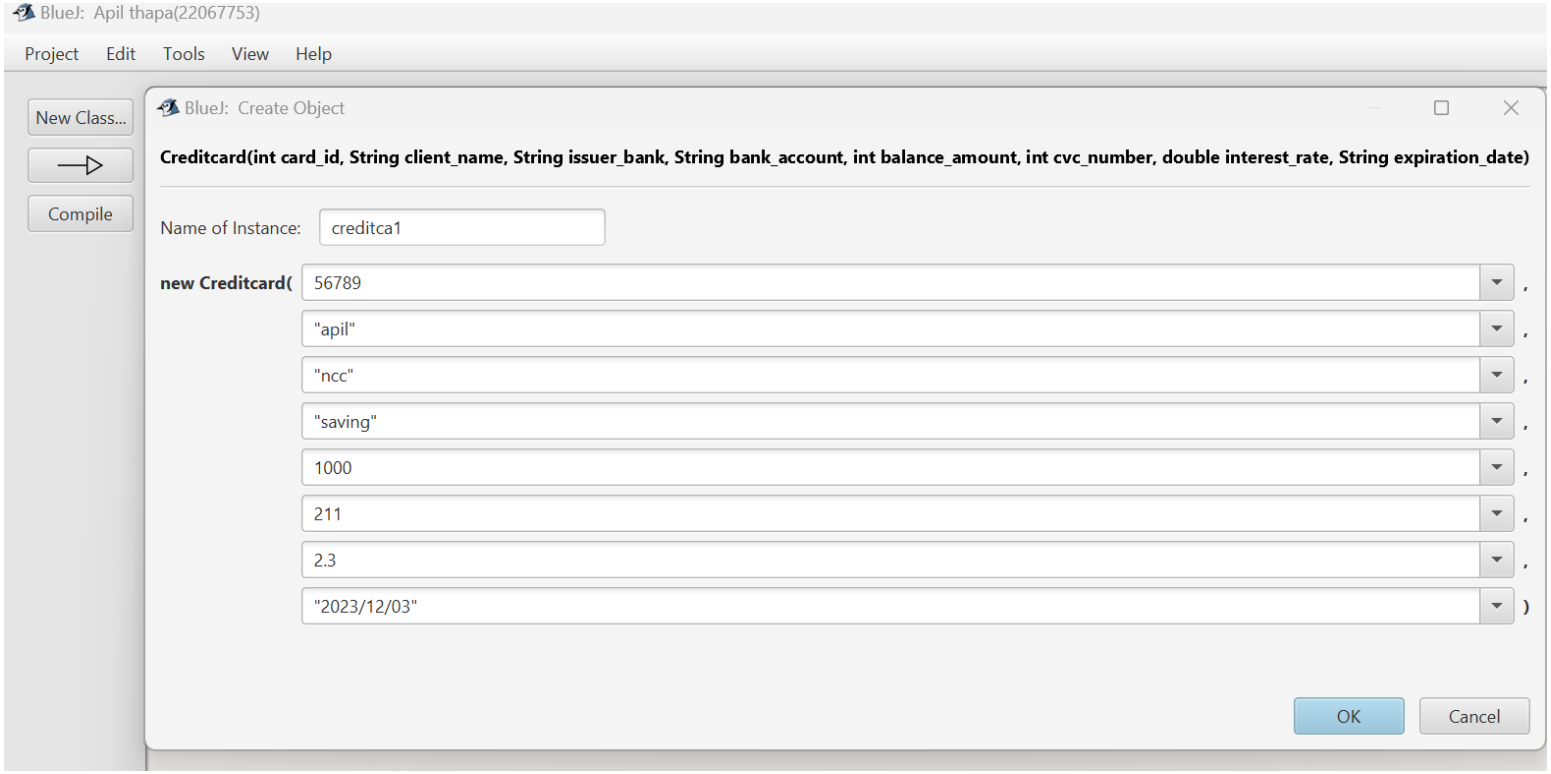


Figure 19screenshot of assigning value of creditcard class

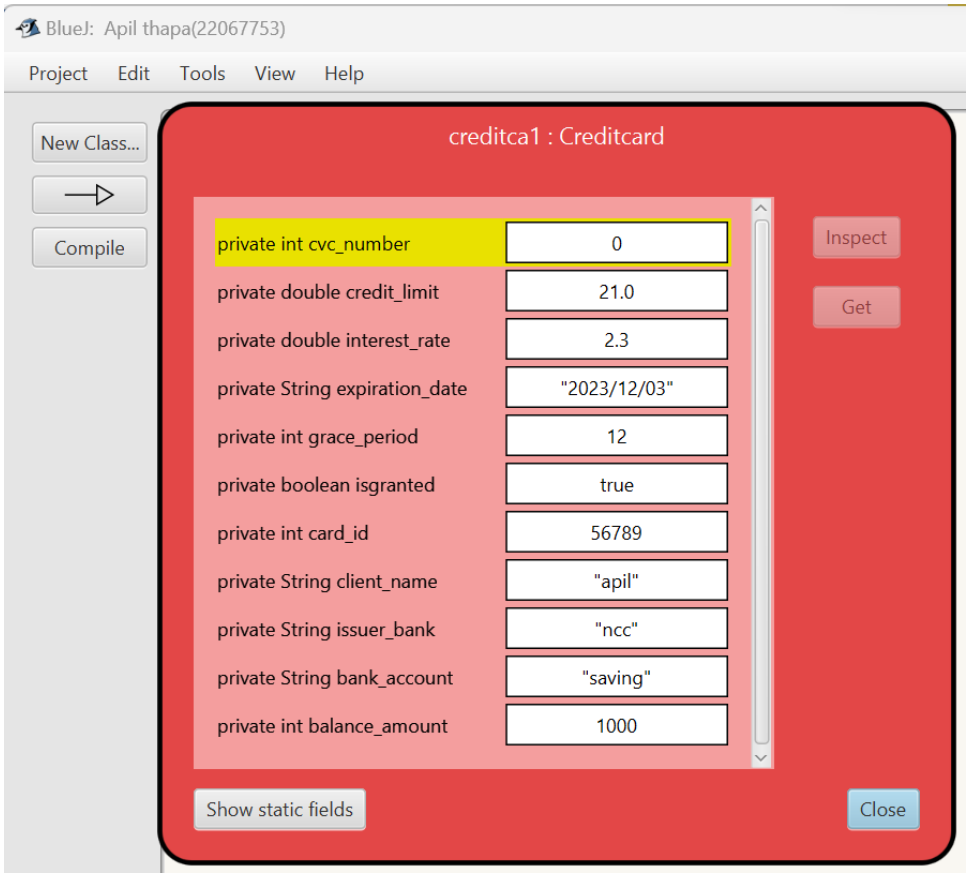
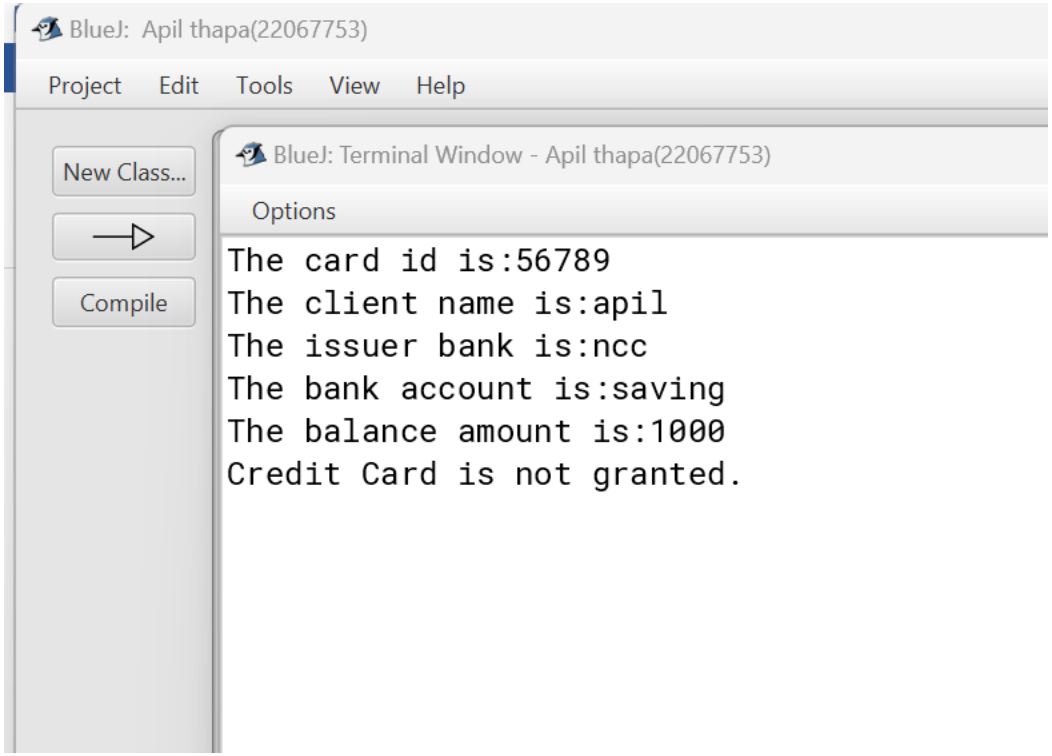

BlueJ: Apil thapa(22067753)
Project
Edit
Tools
View
Help
New Class...
Compile
creditca1 : Creditcard
private int cvc_number
0
private double credit_limit
21.0
private double interest_rate
2.3
private String expiration_date
"2023/12/03"
private int grace_period
12
private boolean isgranted
true
private int card_id
56789
private String client_name
"apil"
private String issuer_bank
"ncc"
private String bank_account
"saving"
private int balance_amount
1000
Inspect
Get
Show static fields
Close

Figure 20screenshot of inspection of credit card class after setting all value

Figure 21screenshot of displaying method called display in crtedit card class



6 Error analysis:

Syntax error detection

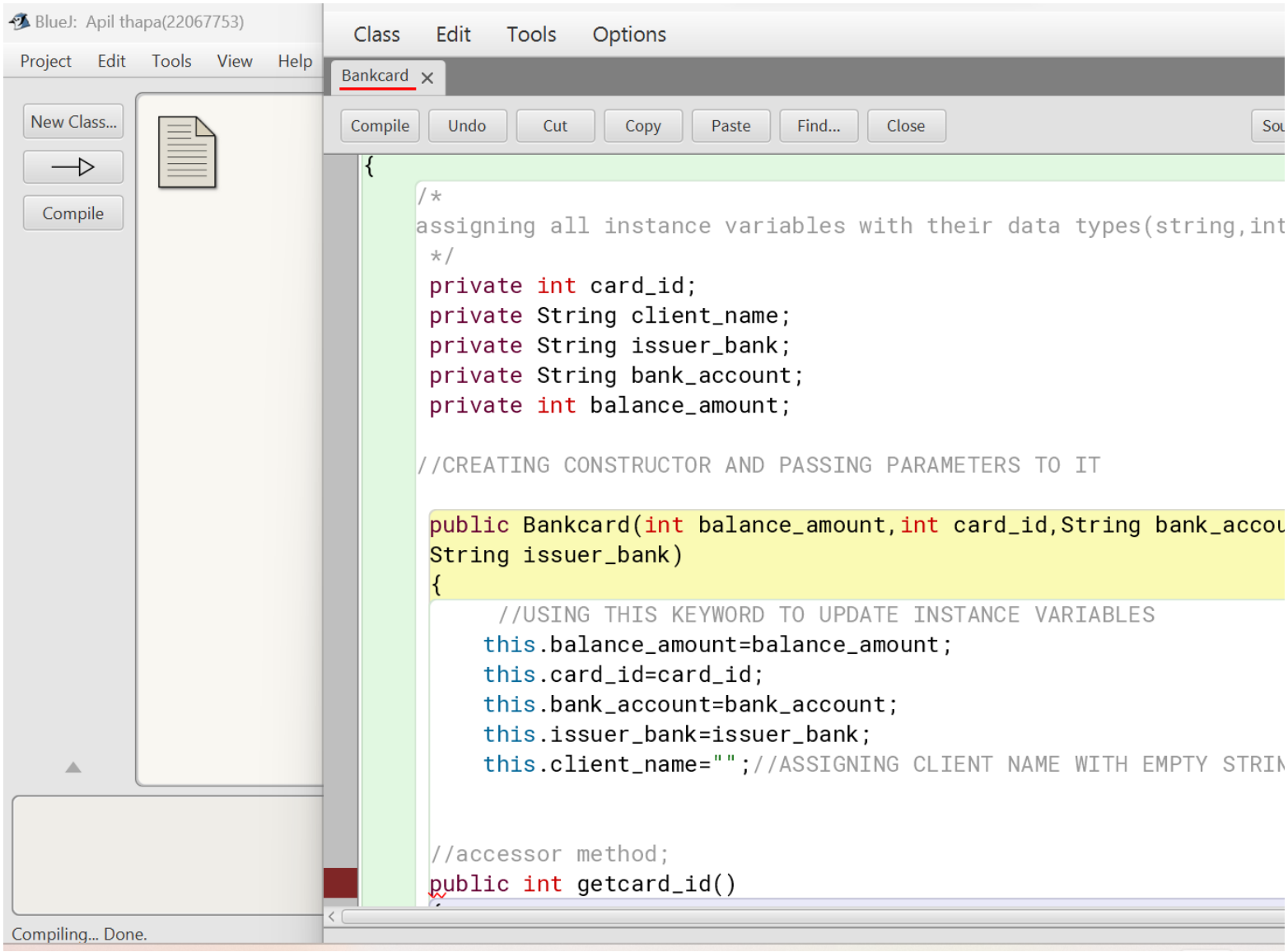


Figure 22 syntax error

In the constructor bankcard curly braces is opened and not closed at the end which is the example of syntax error.

Syntax error correction:

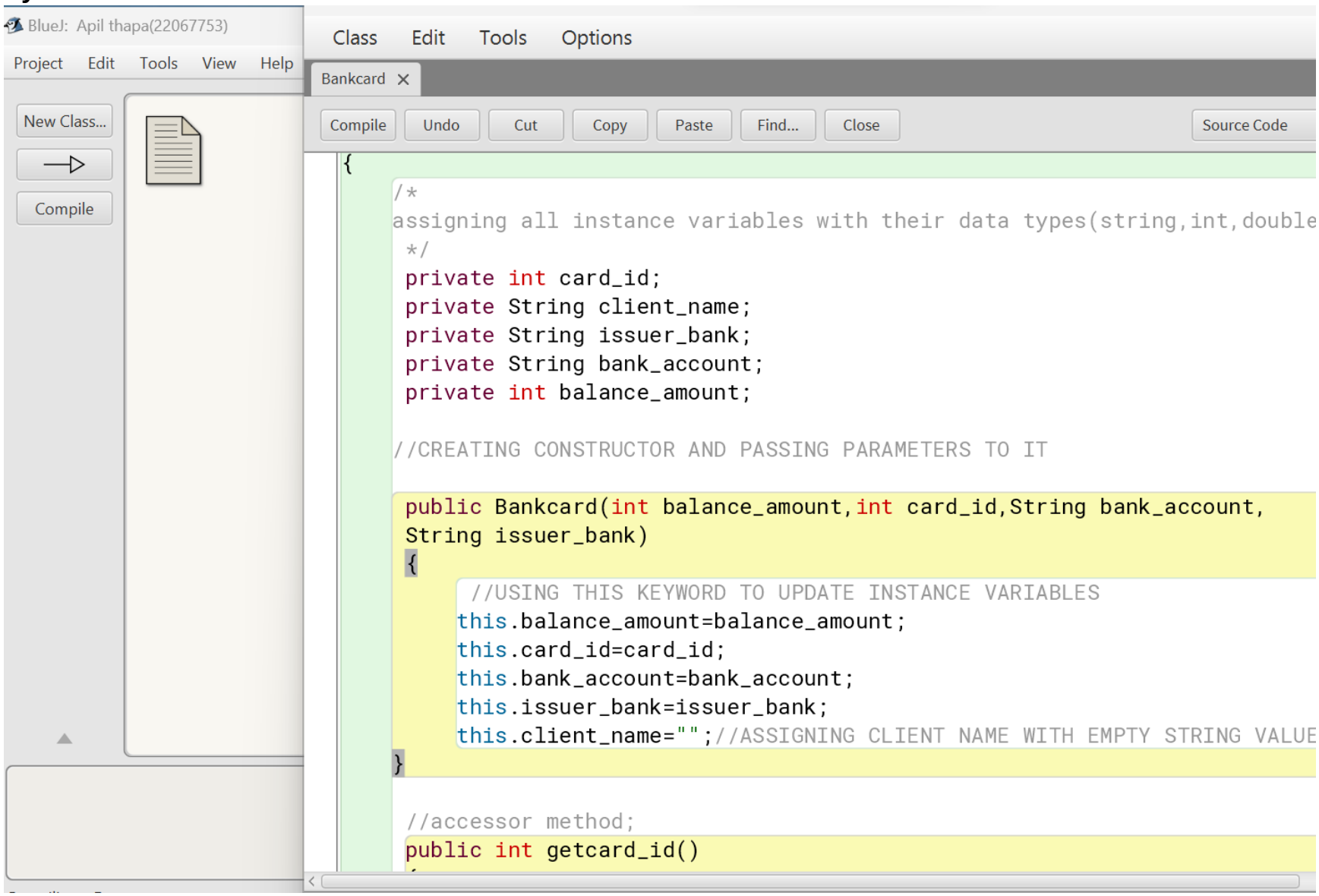


Figure 23syntax error correction

Syntax error is corrected by closing curly braces at constructor bankcard.it is identified by looking each line precisely

semantic error detection

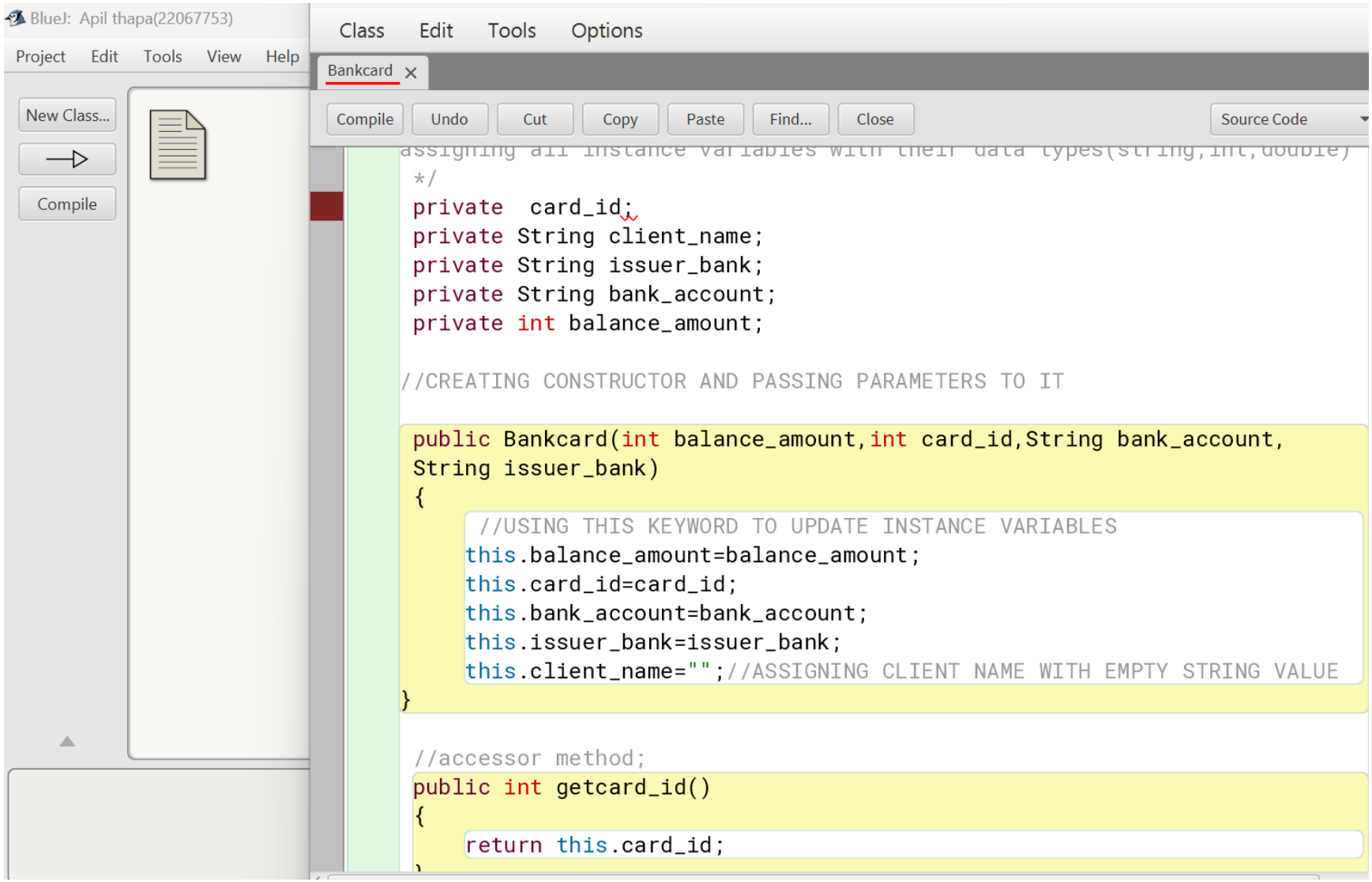


Figure 24symantic error

In this figure value of card id is not assigned which is the example of semantic error

Semantic error correction:

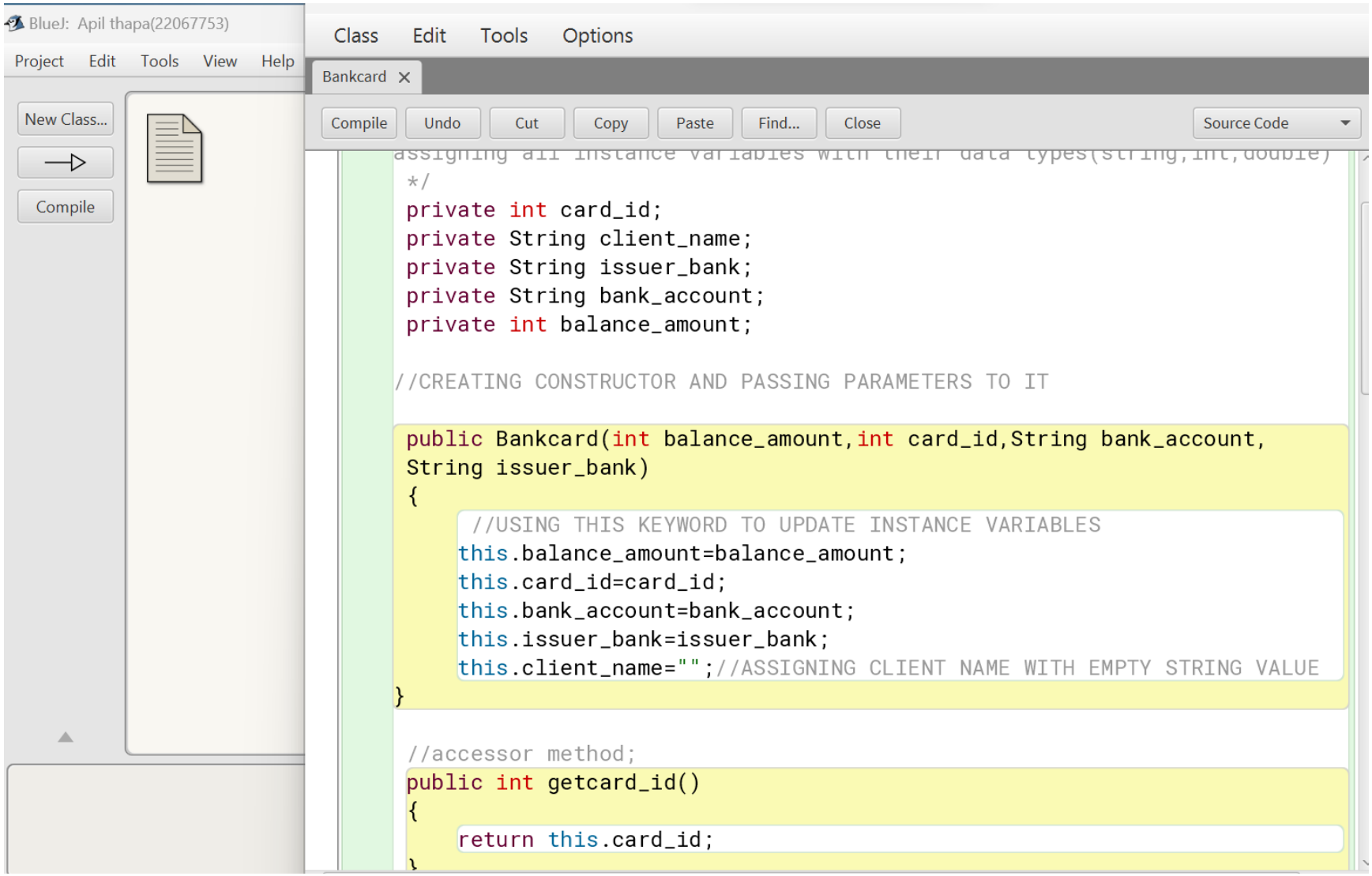


Figure 25symantic error detection

It is corrected when value of card id is assigned to integer which is identified precisely by looking each line.

Logical error detection

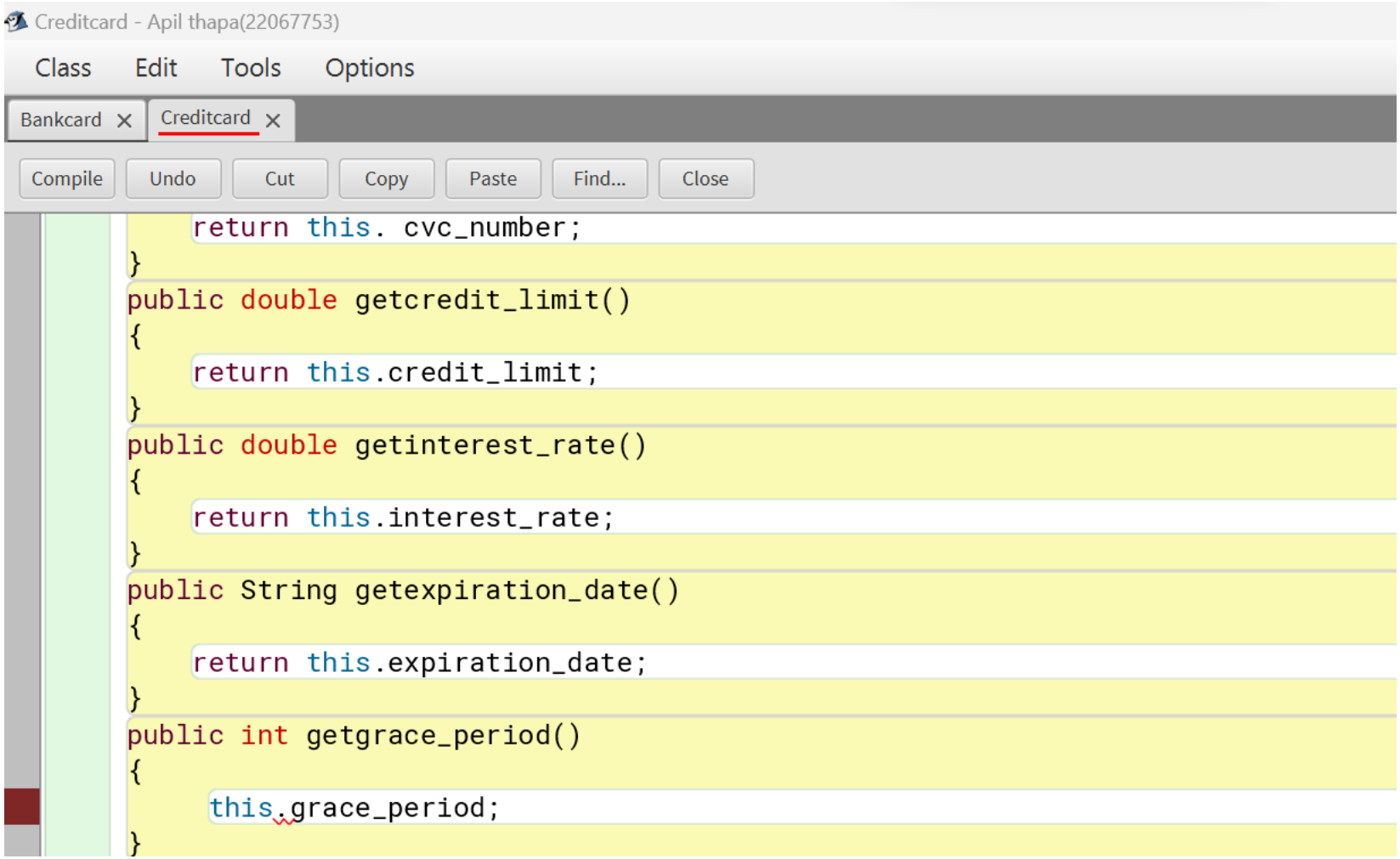


Figure 26logical error

In this getter method there is an error which shows logical error

Logical error correction

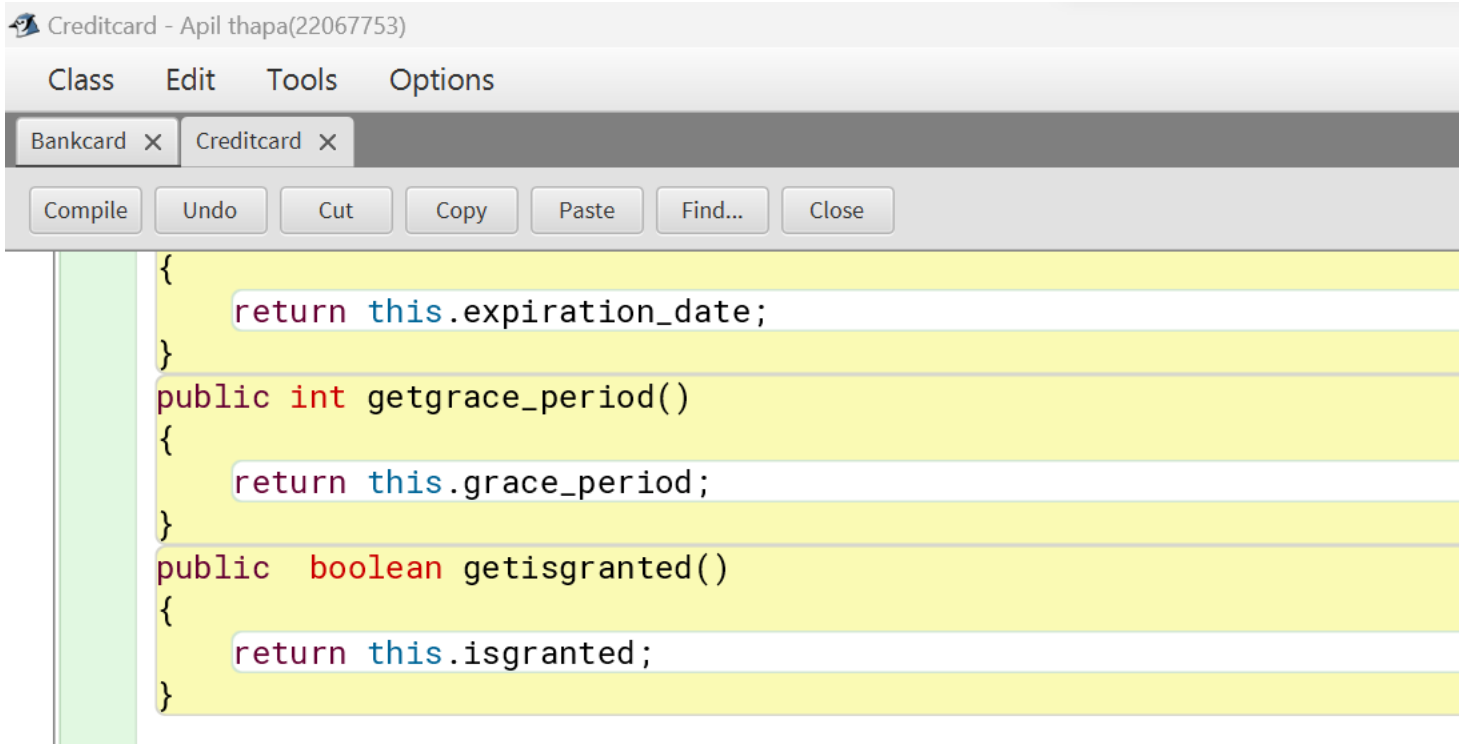


Figure 27logical error detection

It is corrected by returning updated value of grace period which we can clearly see logical error being corrected.

7 Conclusion

My coursework is generally decent. I learn a lot while working on this project. This training is specifically designed to help you create a Java oop software to create debit, credit, and bank cards. After coding, the overall effect was as anticipated. This project demonstrates the operation of a bank card and a pair of financial transaction cards as well as how the code functions. According to my assessment, this idea may be applied to a variety of future projects. My proposal is that, for greater improvements, we should concentrate more on giving those financial transactions security.

I first understood exactly how the code operates.

I'm learning more about financial transactions and how to use my resources thanks to this assignment.

References

Ballew, j., 2021. *ms-word.* [Online]
Available at: https://www.lifewire.com/microsoft-word-4159373
[Accessed 18 january 2023].

booch, G., jacobson, i. & rumbaugh, j., 1994-95. *javatpoint.* [Online]
Available at: https://www.javatpoint.com/uml
[Accessed 18 january 2023].

gosling, j., 20 september 2022. *blue-j.* [Online]
Available at: https://www.bluej.org/
[Accessed 18 january 2023].

ubah, k., july 26 2021. *freecodecamp.* [Online]
Available at: https://www.freecodecamp.org/news/what-is-pseudocode-in-programming/
[Accessed 19 january 2023].

9 Appendix :

Code for Bankcard class

```
public class Bankcard
{

    private int card_id;
    private String client_name;
    private String issuer_bank;
    private String bank_account;
    private int balance_amount;
```

```
 public Bankcard(int balance_amount,int card_id,String bank_account,
 String issuer_bank)
 {
     this.balance_amount=balance_amount;
     this.card_id=card_id;
     this.bank_account=bank_account;
     this.issuer_bank=issuer_bank;
     this.client_name.equals("");
}

 //accessor method;
 public int getcard_id()
 {
     return this.card_id;
 }
```

```
public String getclient_name()
{
    return this.client_name;
}
public String getissuer_bank()
{
    return this.issuer_bank;
}
public String getbank_account()
{
    return this.bank_account;
}
 public int getbalance_amount()
{
    return this.balance_amount;
}
```

```
public void setclient_name(String client_name)
{
    this.client_name=client_name;
}
public void setbalance_amount(int balance_amount)
{
    this.balance_amount=balance_amount;
}

    public void display()
{
     if(client_name.equals(""))
     {
        System.out.print("Please set client name:");
```

```
        }
        else
       {
        System.out.println("The card id is:"+card_id);
        System.out.println("The client name is:"+client_name);
        System.out.println("The issuer bank is:"+issuer_bank);
        System.out.println("The bank account is:"+bank_account);
        System.out.println("The balance amount is:"+balance_amount);
       }



   }
```

Code for Debitcard class

```
public class Debitcard extends Bankcard
```

```
{
    private int pin_number;
    private int withdrawal_amount;
    private String dateofwithdrawal;
    private boolean haswithdrawn;


    public Debitcard(int balance_amount,int card_id,String bank_account,
    String issuer_bank,String client_name,int pin_number)
     {


        super(balance_amount,card_id,bank_account,issuer_bank);
        this.pin_number=pin_number;
        this.haswithdrawn=false;
```

```
  }

public int getpin_number()
{
    return this.pin_number;
}
public int getwithdrawal_amount()
{
    return this.withdrawal_amount;
}
public String getdateofwithdrawal()
{
    return this.dateofwithdrawal;
}
public  boolean gethaswithdrawn()
{
```

```
    return this.haswithdrawn;
}
 public void setwithdrawal_amount(int withdrawal_amount)
{
    this.withdrawal_amount=withdrawal_amount;
}

public void withdraw(int withdrawal_amount,String dateofwithdrawal,int pin_number )
 {
   if(pin_number==this.pin_number)
   {
       if(getbalance_amount()-withdrawal_amount>=0)
           {
             setbalance_amount(getbalance_amount()- withdrawal_amount);
             this.withdrawal_amount=withdrawal_amount;
             this.dateofwithdrawal=dateofwithdrawal;
```

```
            this.haswithdrawn=haswithdrawn;
            System.out.println("Withdrawal successful. New balance: " +this.getbalance_amount());


        }
        else
        {
            System.out.println("Insufficient balance.");
        }
}


  else
  {
      System.out.print("please enter valid information:invalid pin");
  }
```

```
}
  @Override
 public void display()
 {
    super.display();
    if(haswithdrawn==true)
    {
        System.out.print("pin number is:"+pin_number);
        System.out.print("withdrawal_amount is:"+withdrawl_amount);
        System.out.print("dateofwithdrawal is:"+dateofwithdrawal);


    }


    else
     {
```

```
            System.out.print("Transaction not been carried out.");
        }


    }
}
```

```
}
```

Code for Creditcard class

```
public class Creditcard extends Bankcard
```

```
{
    //assigning variables with respective data types.
     private int cvc_number;
     private double credit_limit;
     private double interest_rate;
     private String expiration_date;
     private int grace_period;
     private boolean isgranted;

     public Creditcard(int card_id,String client_name,String issuer_bank,String bank_account,int balance_amount,
     int cvc_number,double interest_rate,String expiration_date)
     {
         super(balance_amount,card_id,bank_account,issuer_bank);
         this.cvc_number=cvc_number;
         setclient_name(client_name);
         this.interest_rate=interest_rate;
```

```
    this.expiration_date=expiration_date;
    this.isgranted=false;

}

public int getcvc_number()
{
    return this. cvc_number;
}
public double getcredit_limit()
{
    return this.credit_limit;
}
public double getinterest_rate()
{
    return this.interest_rate;
```

```
}
public String getexpiration_date()
{
    return this.expiration_date;
}
public int getgrace_period()
{
    return this.grace_period;
}
public  boolean getisgranted()
{
    return this.isgranted;
}

public void setcreditlimit(int credit_limit,int grace_period)
{
```

```
if(credit_limit<=(2.5*getbalance_amount()))
{

    System.out.print("credit granted");
    this.isgranted=true;
    this.credit_limit=credit_limit;
    this.grace_period=grace_period;



}
else
{

    System.out.print(" your credit can't be granted/issued");
```

```
    }



}
public void cancelcreditcard()
{
    cvc_number=0;
    grace_period=0;
    credit_limit=0;
    isgranted=false;



}

public void display()
    super.display();
```

```
        if(isgranted==true)
        {
           System.out.println("CVC: " + cvc_number);
           System.out.println("Credit Limit: " + credit_limit);
           System.out.println("Grace Period: " + grace_period);
        }
       else
       {
           System.out.println("Credit Card is not granted.");
       }
    }
}
```