



**CS4001NI Programming**

**30% Individual Coursework**

**2022-23 Autumn**

**Student Name:** Enter your Full Name Here

**London Met ID:** E.g. 18053646

**College ID:** E.g. NP01MM0474747

**Group: Enter your group here**

**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.*

# Introduction

|  |
| --- |
| Bankcard |
| - balance\_Amount: int  - card\_Id : int  - issuer\_Bank: String  - client\_Name : String  - bank\_Account : String |
| +<<constructor>>Bankcard(balance\_Amount:int, card\_Id : int, Issuer\_Bank : String, bank\_Account : String)  + getBalance\_Amount() : int  + getCard\_Id() : int  + getIssuer\_Bank() : String  + getClient\_Name(): String  + getBank\_Account() : String  + setBalance\_Amount(balance\_Amount:int): void  + setClient\_Name(client\_Name:String): void  + Display(): void |

# Class Diagram

|  |
| --- |
| Creditcard |
| - cvc\_number : int  - credit\_Limit : double  - interest\_Rate : double  - expiration\_Date : String  - grace\_Period : int  - is\_Granted : 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  + getExpiration\_Date() : String  + getGrace\_Period() : int  + getIs\_Granted() : Boolean  + setCredit\_Limit(new\_Creditlimit : double, new\_Gracelimit : int ) : void  + cancel\_Credit\_Card() : void  + Display() : void |

|  |
| --- |
| Debitcard |
| - pin\_Number : int  - date\_Of\_Withdrawal : String  - withdrawal\_Amount : int  - has\_Withdrawn : boolean |
| +<<constructor>>Bankcard(balance\_Amount : int , card\_Id : int, bank\_Account : String Issuer\_Bank : String, client\_Name:String , pin\_Number:int)  +getPin\_Number() : int  +getWithdrawal\_Amount() : int  + getDate\_Of\_Withdrawal() : String  +getHas\_Withdrawn(): boolean  +setWithdrawal\_Amount(withdrawal\_Amount: int) : void  +Withdraw(withdrawal\_Amount:int,pin\_Number:int): void  + Display (): void |

# 3) Pseudocode

## 3.1) Pseudocode for Bankcard class

CREATE a parent class Bankcard using public access modifier

DO

DECLARE instance variable balance\_Amount as int using private access modifier

DECLARE instance variable card\_Id as int using private access modifier

DECLARE instance variable issuer\_Bank as String using private access modifier

DECLARE instance variable client\_Name as String using private access modifier

DECLARE instance variable bank\_Account as String using private access modifier

CREATE constructor of Bank card which accept four parameters

DO

ASSIGN balance\_ Amount attribute to parameter balance\_Amount

ASSIGN card\_Id attribute to parameter card\_Id

ASSIGN client\_Name attribute to empty string

ASSIGN issuer\_Bank attribute to parameter issuer\_Bank

ASSIGN bank\_Account attribute to parameter bank\_Account

END DO

CREATE accessor method getBalance\_Amount() with return type int

DO

RETURN balance\_Amount

END DO

CREATE accessor method getCard\_Id() with return type int

DO

RETURN card\_Id

END DO

CREATE accessor method getIssuer\_Bank() with return type String

DO

RETURN issuer\_Bank

END DO

CREATE accessor method getClient\_Name() with return type String

DO

RETURN client\_Name

END DO

CREATE accessor method getBank\_Account() with return type String

DO

RETURN bank\_Account

END DO

CREATE mutator method setBalance\_Amount() with parameter int balance\_Amount

DO

ASSIGN balance\_Amount attribute to value of parameter balance\_Amount

END DO

CREATE mutator method setClient\_Name() with parameter String client\_Name

DO

ASSIGN client\_Name attribute to value of parameter client\_Name

END DO

CREATE a method Display() with no return type

DO

PRINT “CardID = card\_Id”

IF client\_Name is empty

PRINT “Please enter the client name”

ELSE

PRINT “ ClientName = client\_name”

END IF

PRINT “ IssuerBank = issuer\_Bank ”

PRINT “ BankAccount = bank\_Account ”

PRINT “ BalanceAmount = balance\_Amount “

END DO

## 3.2) Pseudocode for Debitcard class

CREATE a child class Debitcard using public access modifier

DO

DECLARE instance variable pin\_Number as int using private access modifier

DECLARE instance variable date\_Of\_Withdrawal as String using private access modifier

DECLARE instance variable withdrawal\_Amount as int using private access modifier

DECLARE instance variable has\_Withdrawn as boolean using private access modifier

CREATE constructor of Debitcard with a parameter “int balance\_Amount”, “int card\_Id”, “String bank\_Account”,”String issuer\_Bank”, “String client\_Name”,”int pin\_Number”

DO

CALL constructor of parent class with parameter “balance\_Amount”, “card\_Id”, “issuer\_Bank”, “bank\_Account”

CALL superclass mutator method setClient\_Name(client\_Name)

ASSIGN pin\_Number attribute to value of parameter pin\_Number

SET has\_Withdrawn attribute to false

END DO

CREATE an accessor method getPin\_Number with return type int

DO

RETURN pin\_Number

END DO

CREATE an accessor method getWithdrawal\_Amount() with return type int

DO

RETURN withdrawal\_Amount

END DO

CREATE an accessor method getDate\_Of\_Withdrawal() with return type String

DO

RETURN date\_Of\_Withdrawal

END DO

CREATE an accessor method getHas\_Withdrawn() with return type boolean

DO

RETURN has\_Withdrawn

END DO

CREATE a mutator method setWithdrawal\_Amount with parameter int withdrawal\_Amount

DO

ASSIGN withdrawal\_Amount attribute to value of parameter withdrawal\_Amount

END DO

CREATE a method Withdraw with parameter “int withdrawal\_Amount,String” “date\_Of\_Withdrawal”, “int pin\_Number”

DO

IF pin\_Number attribute is equal to the value of parameter pin\_Number THEN

IF withdrawal\_Amount is less or equal to getBalance\_Amount() THEN

SET balance\_Amount equal to ( balance\_Amount – withdrawal\_Amount)

SET has\_Withdrawn attribute to true

END IF

ELSE

PRINT “Not enough balance in your account “

END IF

ELSE

PRINT “Incorrect Pin number, please try again”

END DO

CREATE a method Display() with no return type

DO

IF has\_Withdrawn is set equal to true THEN

CALL Display method of superclass

PRINT “Pin number = pin\_Number”

PRINT “Withdrawal Amount = withdrawal\_Amount”

PRINT “Date of withdrawal = date\_Of\_Withdrawal”

END IF

` ELSE

PRINT “ Withdrawal transaction has not been carried out yet!”

PRINT “ Balance amount = getBalanceAmount()”

END DO

## 3.3) Pseudocode for Creditcard class

CREATE a child class Creditcard using public accessor modifier

DO

DECLARE instance variable cvc\_Number as int using private access modifier

DECLARE instance variable credit\_Limit as double using private access modifier

DECLARE instance variable interest\_Rate as double using private access modifier

DECLARE instance variable expiration\_Date as String using private access modifier

DECLARE instance variable grace\_Period as int using private access modifier

DECLARE instance variable is\_Granted as boolean using private access modifier

CREATE a constructor of Creditcard with parameter “int card\_Id”, “String client\_Name”, “String issuer\_Bank”, “String bank\_Account”, “int balance\_Amount”,” int cvc\_Number”,” double interest\_Rate”, “String expiration\_Date”

DO

CALL constructor of parent class with parameter “balance\_Amount”, “card\_Id”, “issuer\_Bank”, “bank\_Account”

ASSIGN cvc\_Number attribute to value of parameter cvc\_Number

ASSIGN interest\_Rate attribute to value of parameter interest\_Rate

ASSIGN expiration\_Date attribute to value of parameter expiration\_Date

SET is\_Granted attribute to false

CALL method setBalance\_Amount and SET the value of balance\_Amount attribute

CALL method setClient\_Name and SET the value of a client\_Name attribute

END DO

CREATE an accessor method getCvc\_Number() with return type int

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 tyoe double

DO

RETURN interest\_Rate

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 int

DO

RETURN grace\_Period

END DO

CREATE an accessor method getIs\_Granted() with return type boolean

DO

RETURN is\_Granted

END DO

CREATE a method setCredit\_Limit with parameter “doublenew\_Creditlimit”, “int new\_Gracelimit”

DO

IF credit\_Limit is less or equal to two times the balance\_Amount THEN

SET is\_Granted to true

ASSIGN credit\_Limit attribute to new\_Creditlimit

END IF

ELSE

PRINT “the credit amount can’t be issued, please try again later”

END DO

CREATE a method cancel\_Credit\_Card with no return type

DO

SET credit\_Limit attribute to zero

SET cvc\_Number attribute to zero

SET grace\_Period attribute to zero

SET is\_Granted attribute to false

END DO

CREATE a method Display with no return type

DO

IF is\_Granted attribute is equal to true THEN

# Method Description

## 4.1) Bankcard

getBalance\_Amount

It is a getter/accessor method with a return type integer which is used to return the value of balance\_Amount attribute when it is called.

getCard\_Id

It is a getter/accessor method with a return type integer which is used to return the value of card\_Id attribute when it is called.

getIssuer\_Bank

It is a getter/accessor method with a return type String which is used to return the value of issuer\_Bank attribute when it is called.

getClient\_Name

It is a getter/accessor method with a return type String which is used to return the value of client\_Name attribute when it is called.

getBank\_Account

It is a getter/accessor method with a return type String which is used to return the value of bank\_Account attribute when it is called.

setBalance\_Amount

It is a mutator/setter method with a parameter “balance\_Amount” which is used for assigning the value to balance\_Amount attribute.

setClient\_Name

It is a mutator/setter method with a parameter “client\_Name” which is used for assigning the value to client\_Name attribute.

Display

It is a method with no return type which is used for displaying the detail of a Bankcard class.

## 4.2) Debitcard

getPin\_Number

It is a getter/accessor method with a return type integer which is used to return the value of pin\_Number attribute when it is called.

getWithdrawal\_Amount

It is a getter/accessor method with a return type integer which is used to return the value of withdrawal\_Amount attribute when it is called.

getDate\_Of\_Withdrawal

It is a getter/accessor method with a return type String which is used to return the value of date\_Of\_Withdrawal attribute when it is called.

getHas\_Withdrawn

It is a getter/accessor method with a return type boolean which is used to return the value of has\_Withdrawn attribute when it is called.

setWithdrawal\_Amount

It is a mutator/setter method with a parameter “withdrawal\_Amount” which is used for assigning the value to withdrawal\_Amount attribute.

Withdraw

It is method with no return type which is used for withdrawing the amount from balance amount.

Display

It is a method with no return type which is used for displaying the detail of a debit card.

## 4.3) Creditcard

getCvc\_Number

It is a getter/accessor method with a return type integer which is used to return the value of cvc\_Number attribute when it is called.

getCredit\_Limit

It is a getter/accessor method with a return type double which is used to return the value of credit\_Limit attribute when it is called.

getInterest\_Rate

It is a getter/accessor method with a return type double which is used to return the value of interest\_Rate attribute when it is called.

getExpiration\_Date

It is a getter/accessor method with a return type String which is used to return the value of expiration\_Date attribute when it is called.

getGrace\_Period

It is a getter/accessor method with a return type integer which is used to return the value of grace\_Period attribute when it is called.

getIs\_Granted  
It is a getter/accessor method with a return type boolean which is used to return the value of is\_Granted attribute when it is called.

setCredit\_Limit

It is a method with no return type and two parameter “double new\_Creditlimit” , “int new\_Grace” which is used to set the credit limit.

cancel\_Credit\_Card

It is a method with no return type which is used to cancel the credit card of a client.

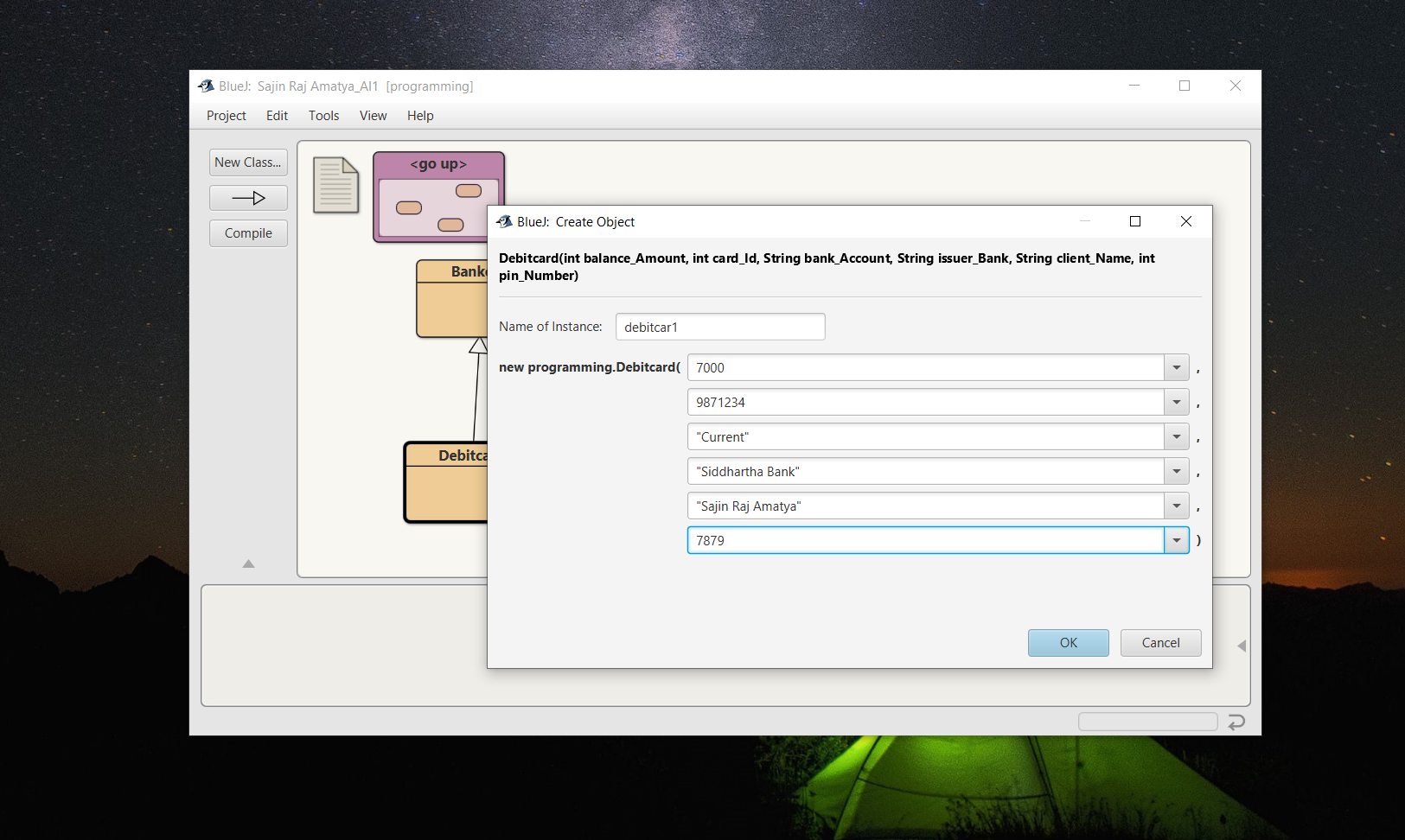
Display

It is a method with not return type which is used to display the detail of credit card.

# Testing

## 5.1) Test 1: Inspect the Debit Card class, withdraw the amount, and re- inspect the Debit Card Class

|  |  |
| --- | --- |
| Test No. | 1 |
| Objective: | To inspect the Debit card class, withdraw the amount, and re-inspect the Debit Debit Card Class |
| Action: | * The Debit card is called with the following arguments:   balance\_Amount = 7000  card\_Id = 9871234  bank\_Account = “Current”  issuer\_Bank = “Siddhartha Bank”  client\_Name = “Sajin Raj Amatya”  pin\_Number = 7879   * Inspection of the Debit card class. * void Withdraw is called with the following argument.   withdrawal\_Amount = 5000  date\_Of\_Withdrawal = “2023-01-21”  pin\_Number = 7879   * Re-inspection of the Debi tcard class |
| Expected Result: | The required amount would be withdrawn from the Debit card |
| Actual Result: | The required amount was withdrawn from the Debit card |
| Conclusion: | The test is successful. |

Output Result:

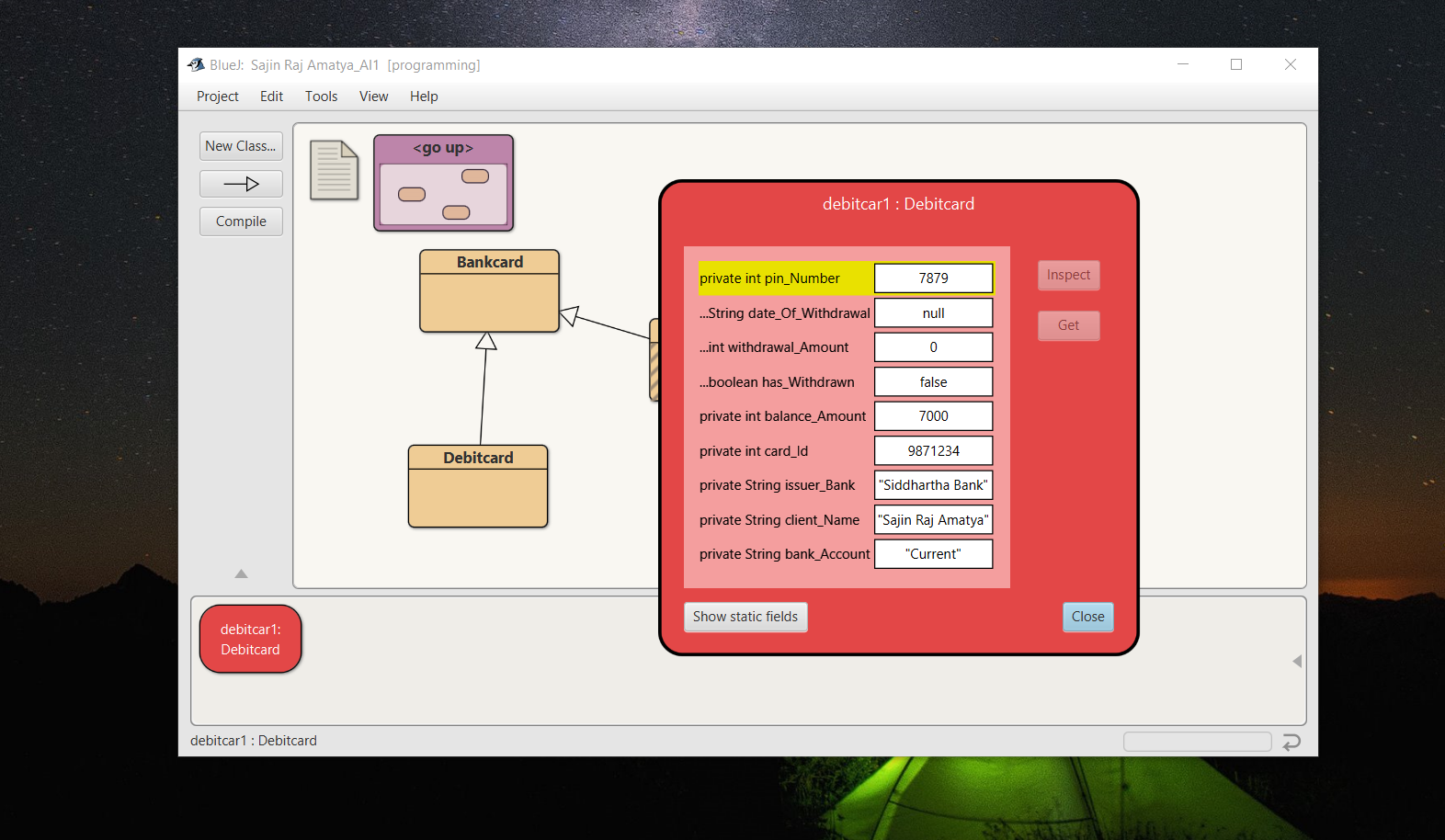


Figure 1 : Screenshot of creating object and assigning the data in Debit card class

Figure 2 Screenshot for inspection of Debit card class

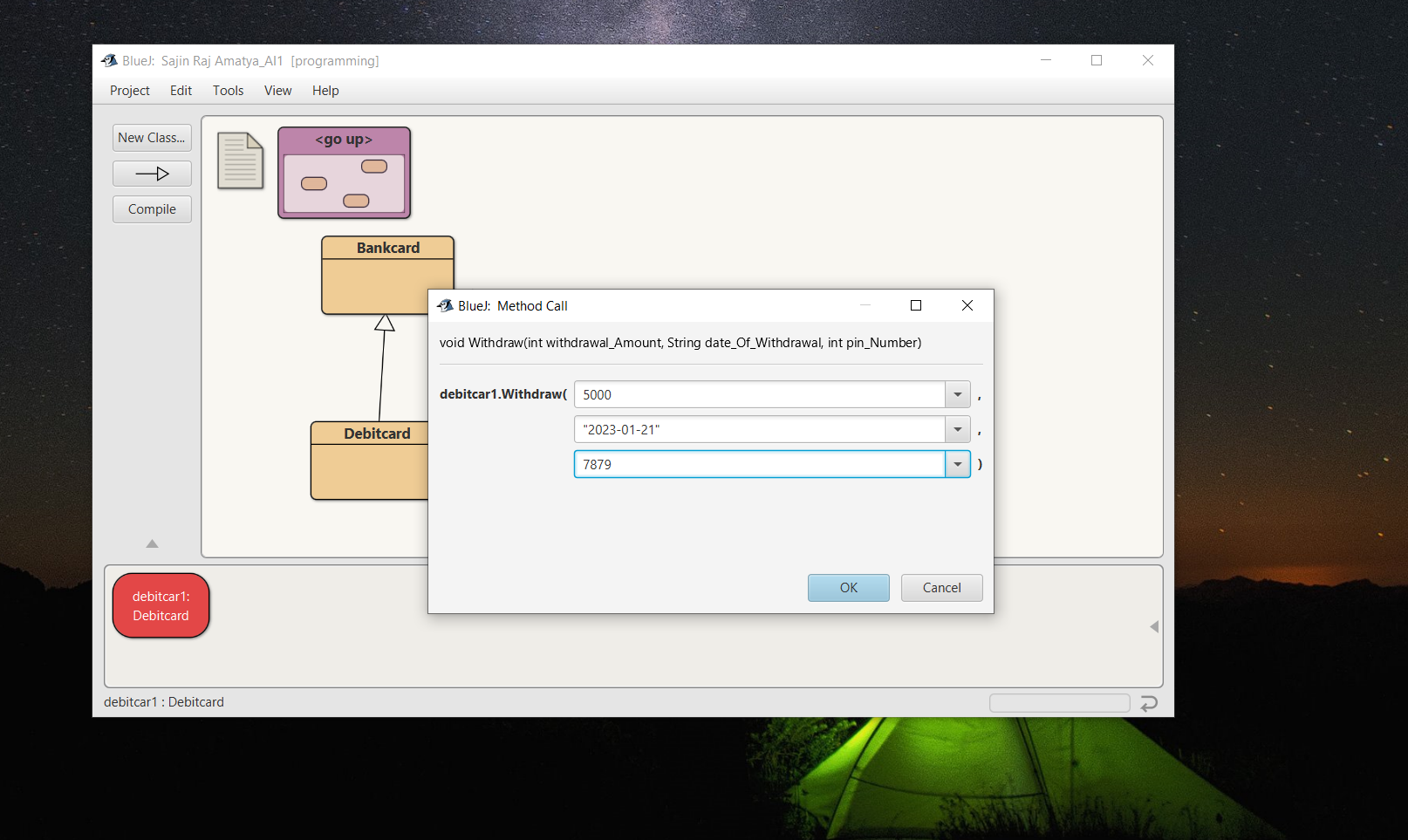
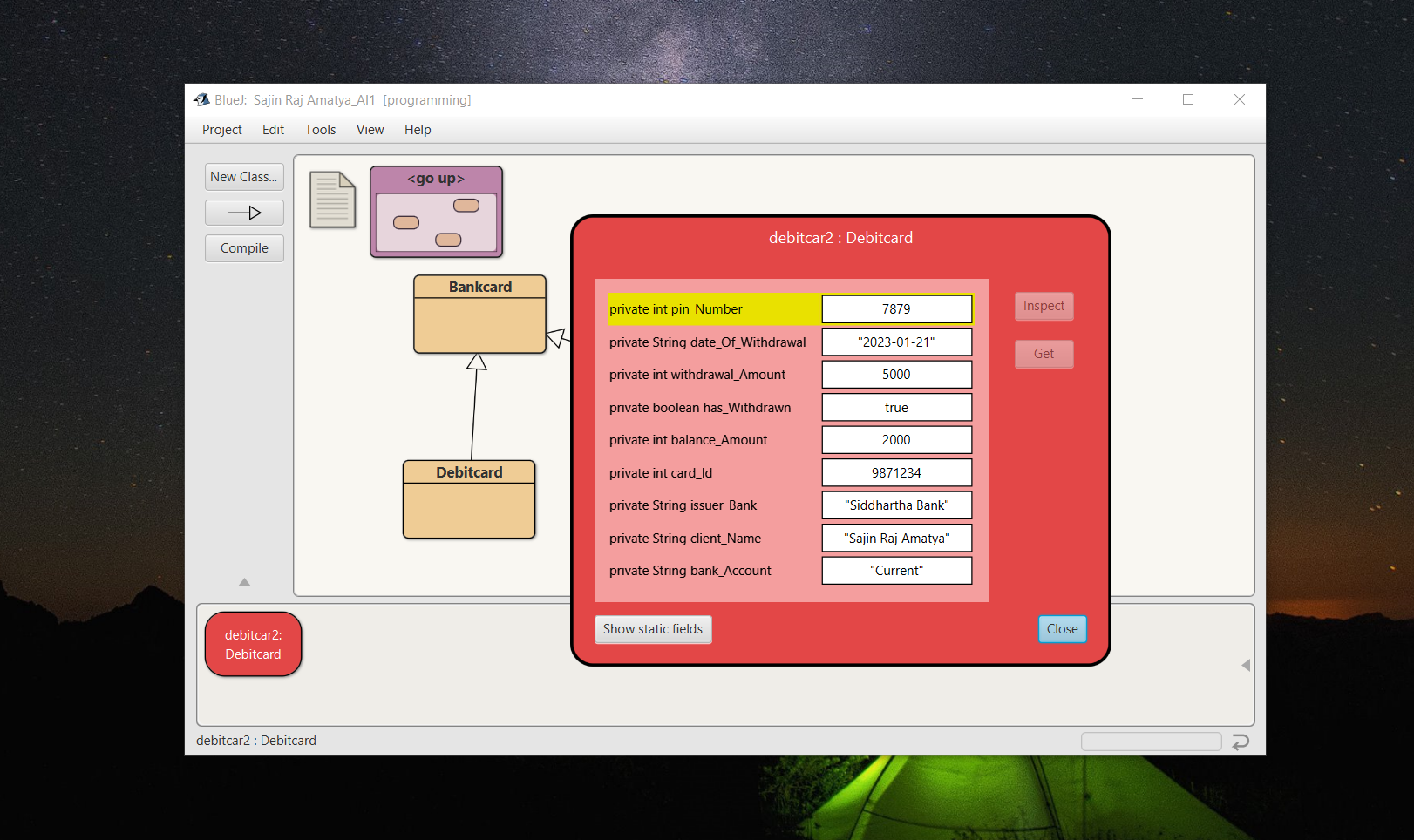
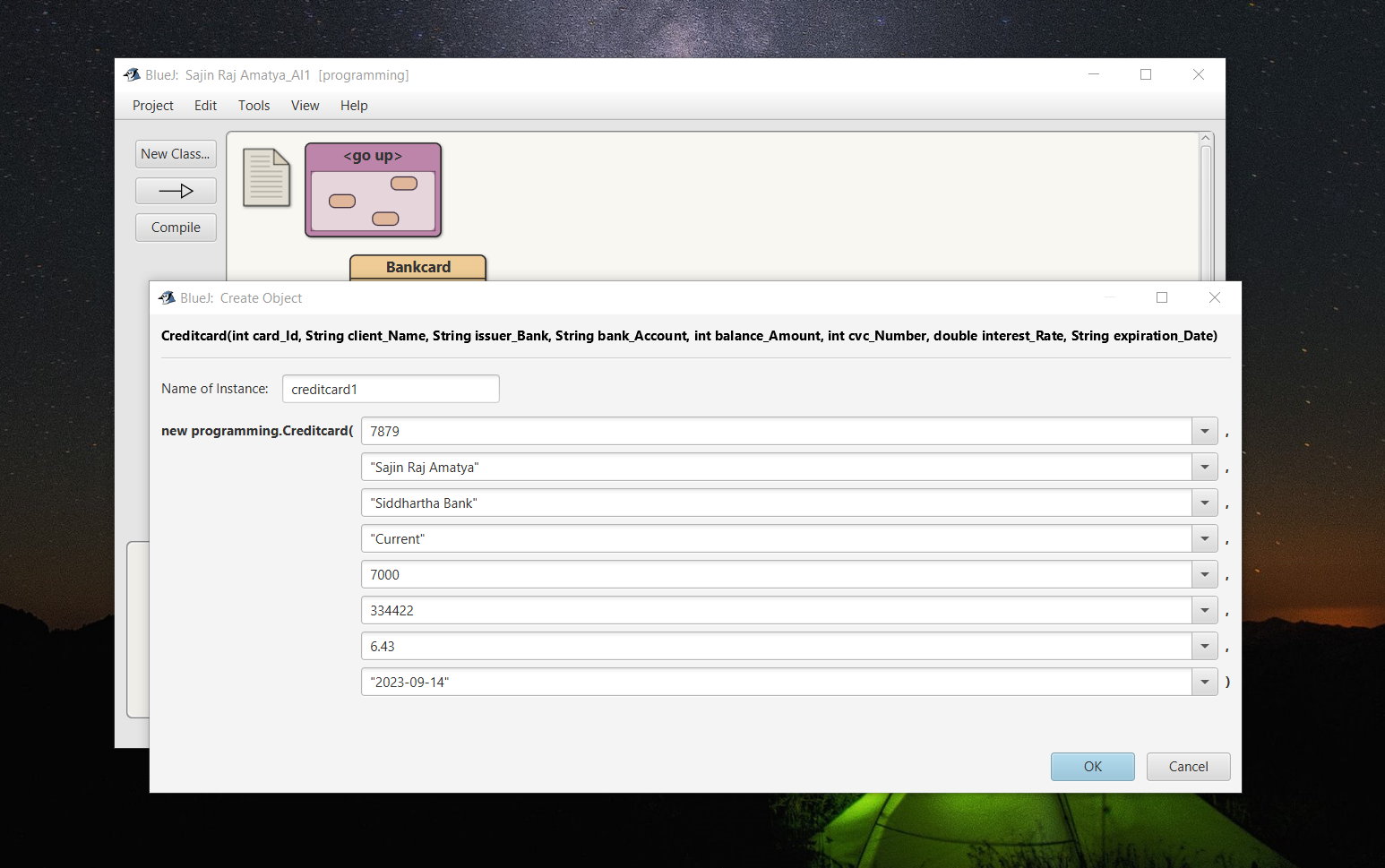


Figure 3 Screenshot for Inserting the data in void Withdraw method

Figure 4 Screenshot for Re-inspection after withdrawing the amount

|  |  |
| --- | --- |
| Test No. | 2 |
| Objective: | To inspect the Credit Card class, set the credit limit and re-inspect the  Credit Card class |
| Action: | * The Credit card is called with the following arguments:   card\_Id= 7879  client\_Name = “Sajin Raj Amatya”  issuer\_Bank = “Siddhartha Bank”  bank\_Account = “Current”  balance\_Amount = 7000  cvc\_Number = 334422  interest\_Rate = 6.43  expiration\_Date = “2023-09-14”   * Inspection of the Credit Card class. * void setCredit\_Limit is called with the following argument.   new\_Creditlimit = 20000  new\_Graceperiod = 11   * Re-inspection of the Credit card class |
| Expected Result: | The credit limit would be set on the Credit card |
| Actual Result: | The credit limit was set on the Credit card. |
| Conclusion: | The test is successful. |

## 5.2) Test 2: Inspect Credit Card class, set the credit limit and reinspect the Credit Card class



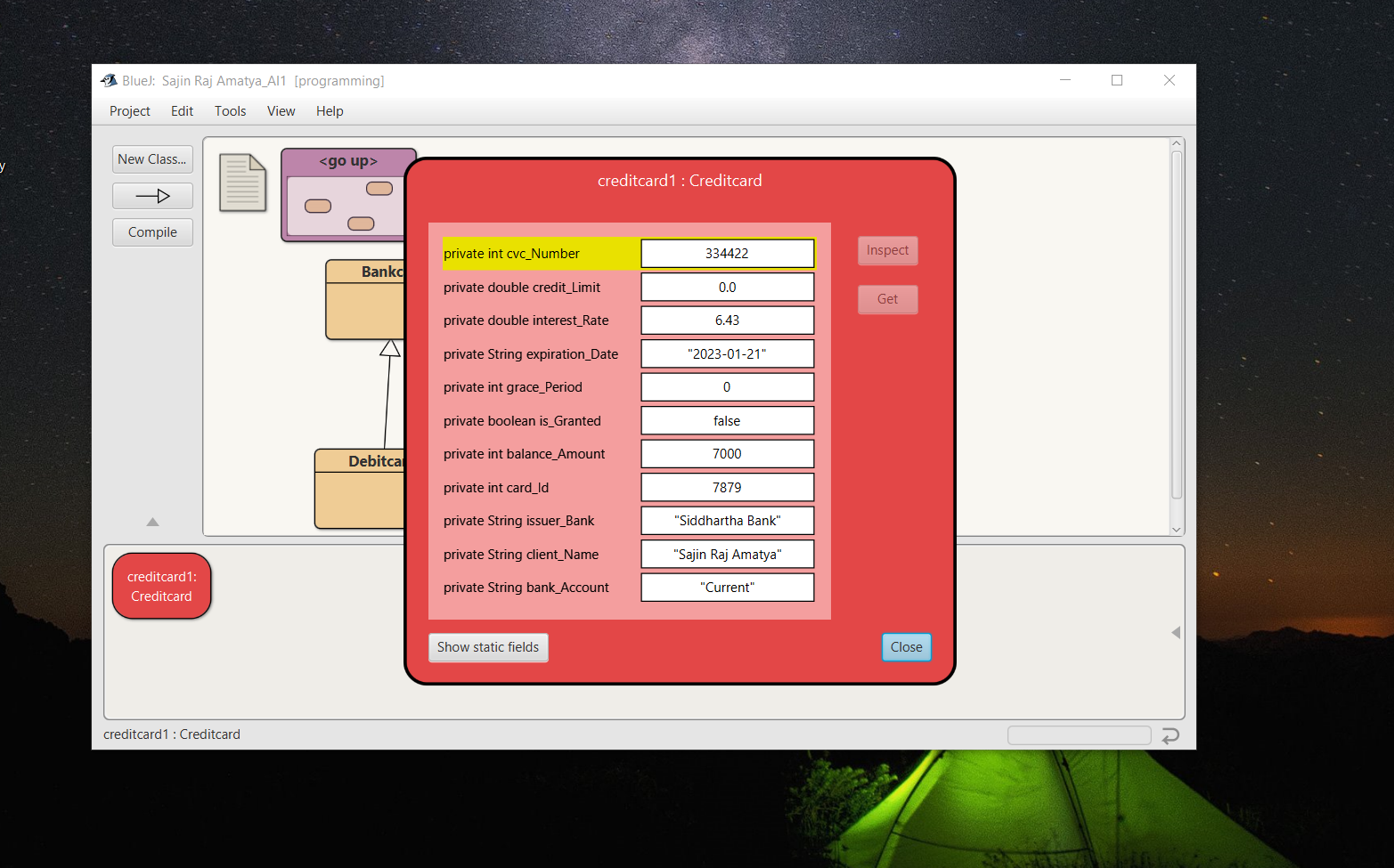


Figure 5 Screenshot of creating object and assigning the data in Credit card class

Figure 6 Screenshot for inspection of Credit card class

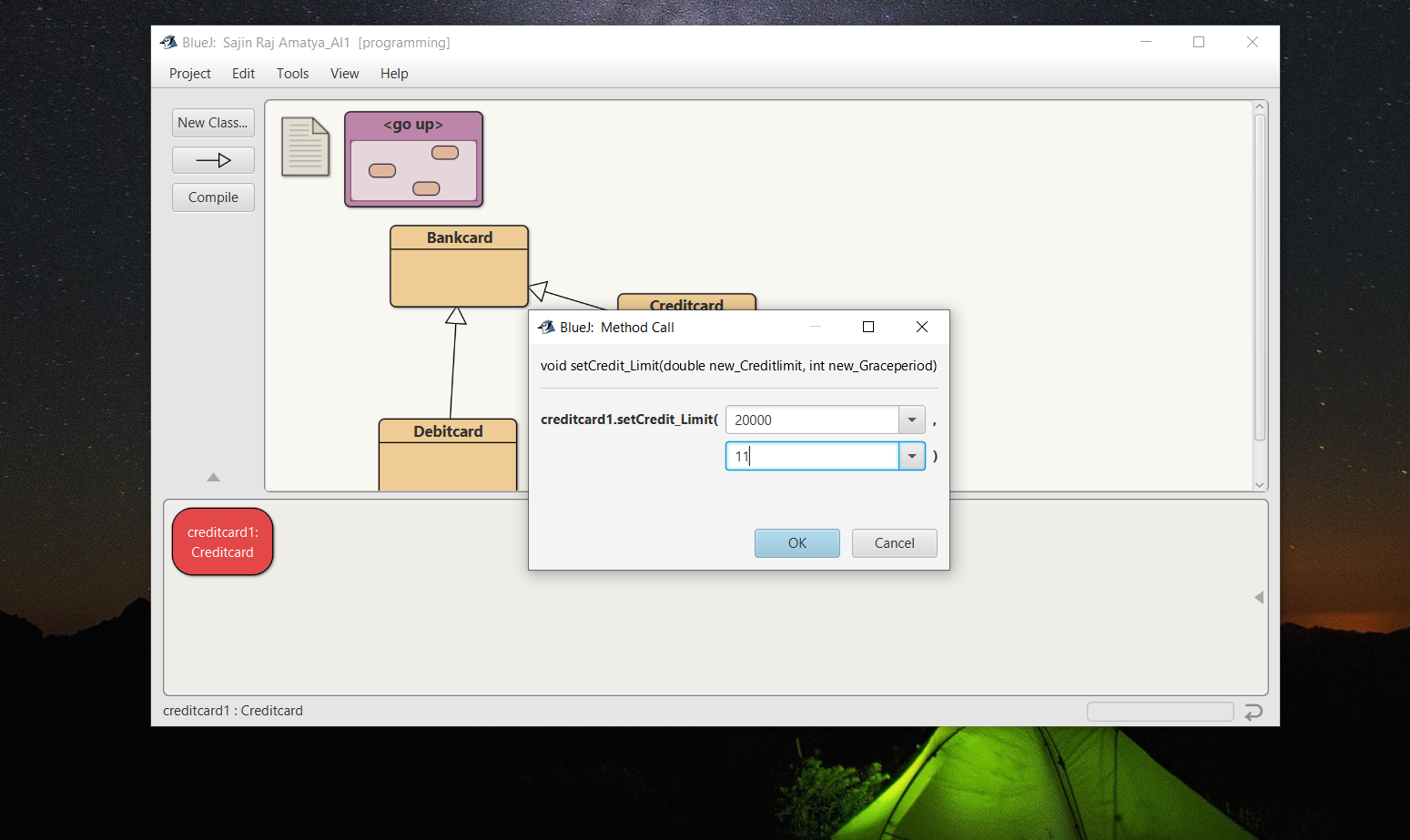


Figure 7 Screenshot of inserting data in void setCredit\_Limit

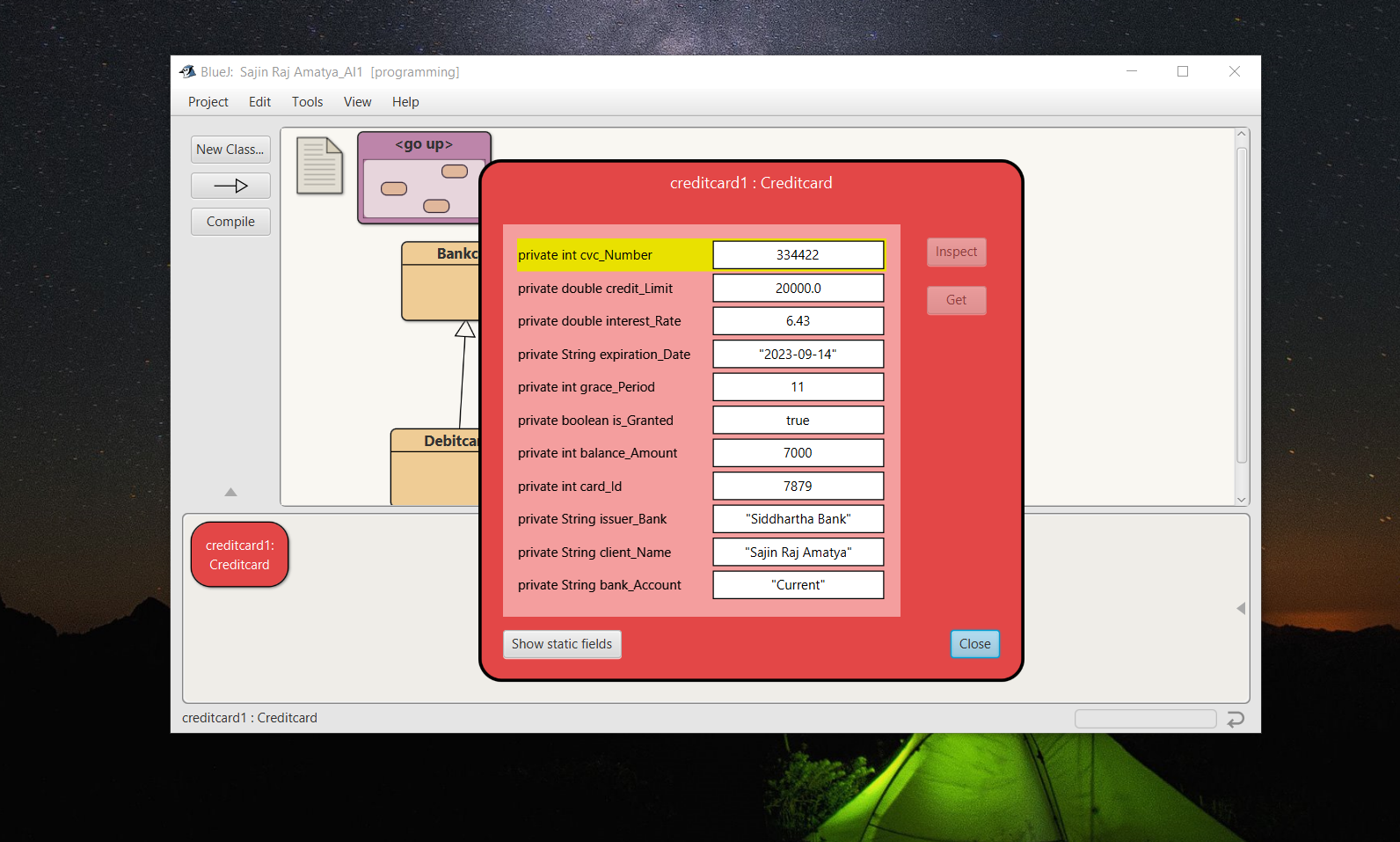


Figure 8 Screenshot of Re-inspection after setting credit limit

## 5.3) Test 3: Inspect Credit Card class again after cancelling the credit card.

|  |  |
| --- | --- |
| Test No. | 3 |
| Objective: | To Inspect the Credit Card class again after canceling the credit card. |
| Action: | * The void cancelCredit\_Card method is called where,   Cvc\_Number is set to zero.  Credit\_Limit is set to zero.  Grace\_Period is set to zero.  Is\_Granted is set to false.   * Inspection of a Credit card after canceling. |
| Expected Result: | The detail of the Credit card would be removed |
| Actual Result: | The detail of the Credit card was removed |
| Conclusion: | The test is successful. |

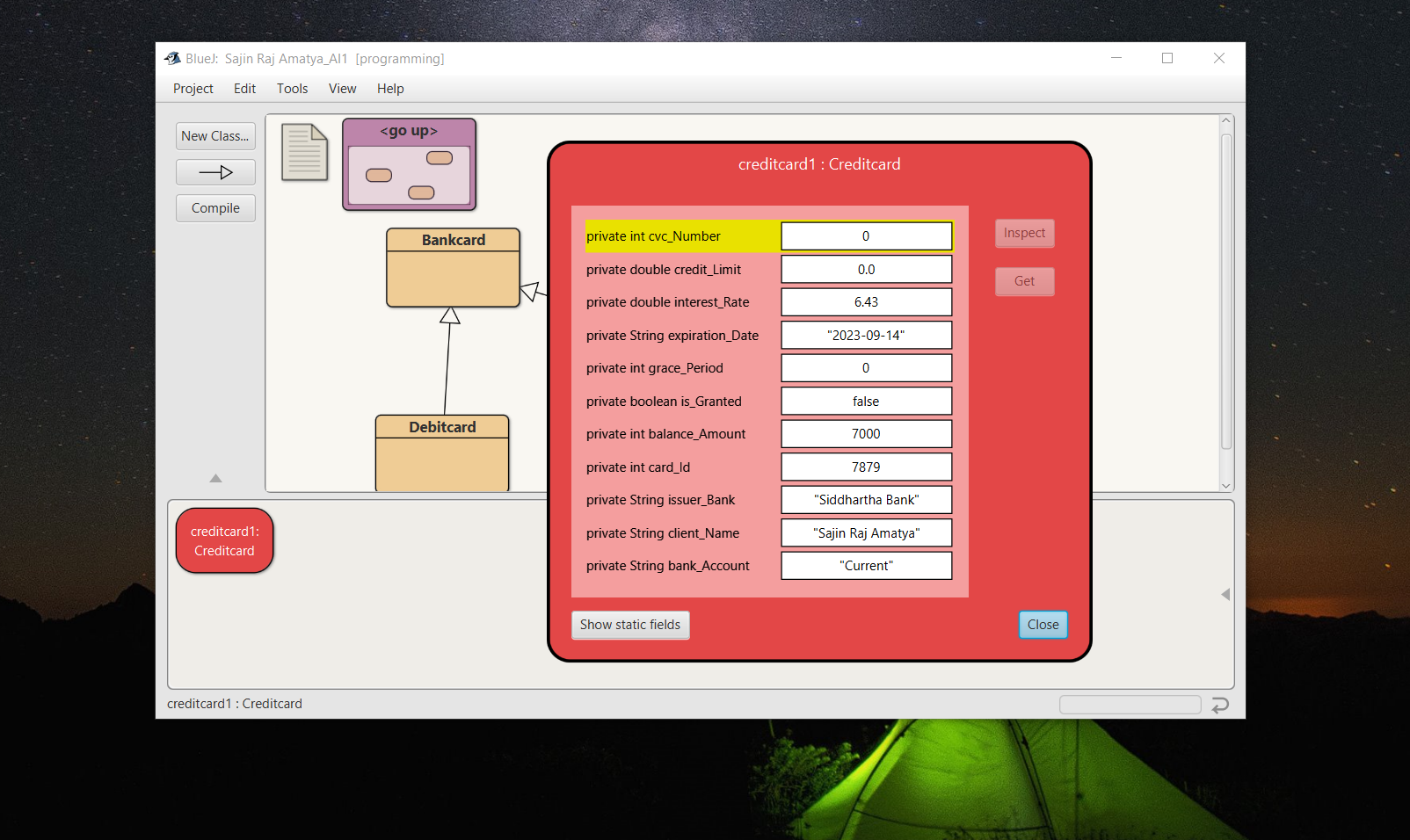


Figure 9 Screenshot of inspection after cancelling of Credit card

## 5.4) Test 4: 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: | * The void Display method is called from the Debit card class. * Display the output of the Debit card class. * The void Display method is called from the Credit card class. * Display the output of the Credit card class. |
| Expected Result: | The detail of the Credit card and Debit card would be displayed |
| Actual Result: | The detail of the Credit card and Debit card was displayed. |
| Conclusion: | The test is successful. |



Figure 10 Screenshot for displaying the detail of Credit card class

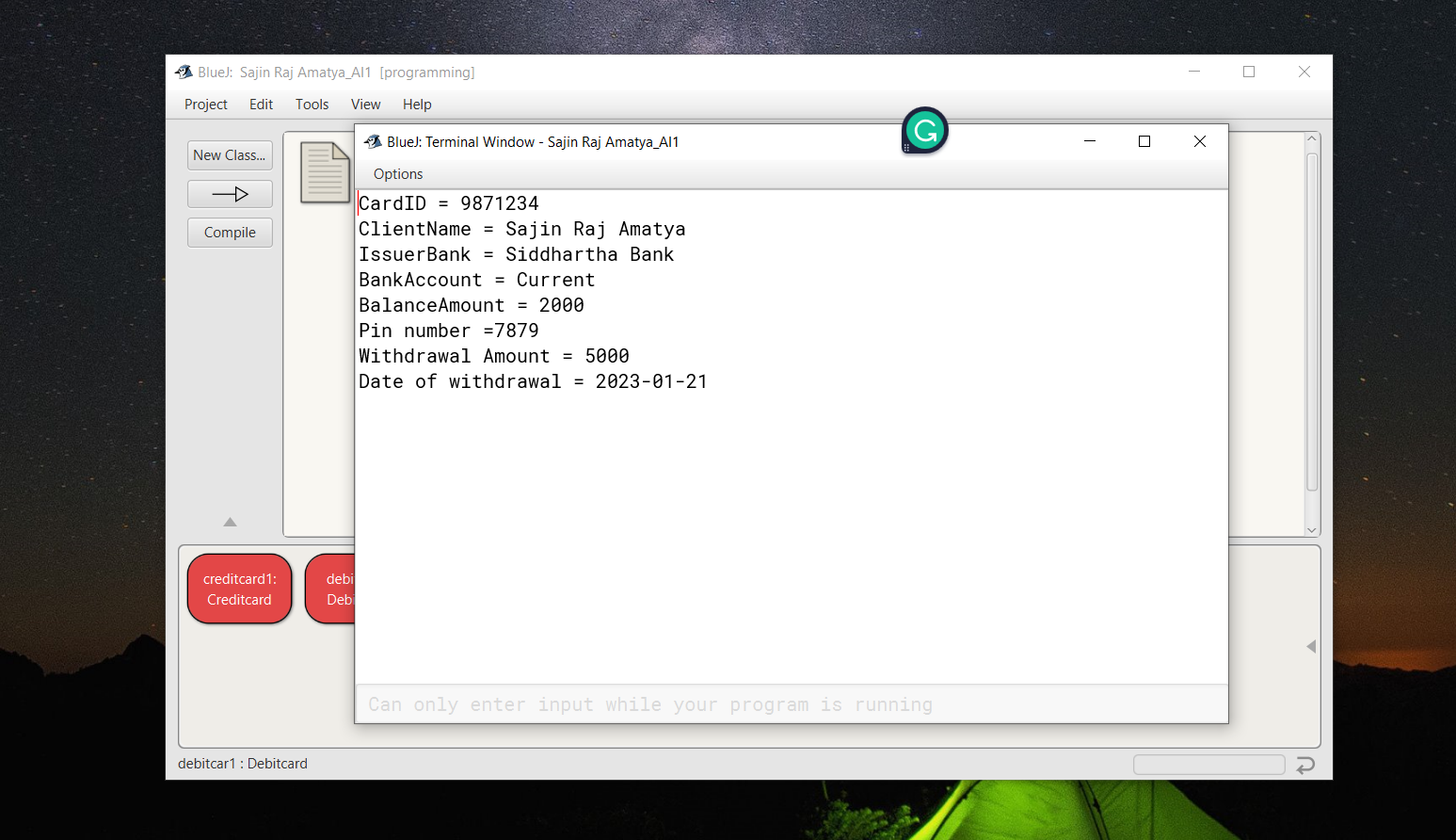


Figure 11 Screenshot for displaying the detail of a Debit card class