

Islington College



Islington college

Programming

CS4001NI

Coursework 2

Submitted By:

Prayash Bikram Shah

16033180

Group: C7

Date: 14th April, 2017

Submitted To:

Mr. Rabin Regmi

Mr. Weenit Maharjan

RTE Department

Word Count: 2661 (Whole Report)

Semester: Autumn

Table of Contents

Class Diagram.....	1
Pseudocode	4
Add Car to Buy.....	4
Add Car to Rent	4
Buy Car	5
Rent Car.....	5
Return Car	6
Display All	6
Method Description	8
Testing	10
Test 1: 1.1: Adding a car to buy	10
1.2: Adding a car to rent	11
1.3: Selling a car	12
1.4: Renting a car	13
1.5: Returning a car	14
1.6: Displaying all of the cars.....	15
Test 2: Dialog Boxes.....	16
Error Detection:	17
Error 1:.....	17
Error 2:.....	17
Error 3:.....	19
Conclusion	21
Appendix:	22

Table of Figures:

Figure 1 Class diagram of the project shown in Blue J.....	3
Figure 2 Inserting value for adding a car to buy	10
Figure 3 Displaying the Car added.....	10
Figure 4 Inserting values to add a car to rent.....	11
Figure 5 Displaying the car added for renting.....	11
Figure 6 Inserting values to buy a car	12
Figure 7 Display of after car bought	12
Figure 8 Inserting values to rent a car	13
Figure 9 Display after renting a car	13
Figure 10 Inserting value to return the rented car	14
Figure 11 Display after returning the rented car	14
Figure 12 Displaying all the cars added	15
Figure 13 Dialog Box showing error message.....	16
Figure 14 Dialog Box Showing Success Message	16
Figure 15 Syntax Error in while Coding.....	17
Figure 16 Syntax error solved	17
Figure 17 Runtime error showing a button not working	17
Figure 18 Code for the button that was not working.....	18
Figure 19 Solved Code.....	18
Figure 20 Result After solving the problem of the button code.....	18
Figure 21 Runtime Error of not taking the input that was on the list	19
Figure 22 Part of coding where the problem was present	19
Figure 23 Solved the problem in the code.....	20
Figure 24 Solved the runtime error.....	20
Figure 25 Source Code 1	22
Figure 26 Source Code 2	22
Figure 27 Source Code 3	23
Figure 28 Source Code 4	23

Figure 29 Source Code 5	24
Figure 30 Source Code 6	24
Figure 31 Source Code 7	25
Figure 32 Source Code 8	25
Figure 33 Source Code 9	26
Figure 34 Source Code 10	26
Figure 35 Source Code 11	27

Table of Tables:

Table 1 Class Diagram of Car Company.....	2
Table 2 Short Description of all methods.....	9
Table 3 Test Table 1	10
Table 6 Test Table 2	11
Table 7 Test Table 3	12
Table 8 Test Table 4	13
Table 9 Test Table 5	14
Table 10 Test Table 6	15
Table 11 Test Table 7	16

Class Diagram

CarCompany
<ul style="list-style-type: none">- description: String- Price: Integer- Year: Integer- mileage: Integer- fee: Integer- rate: Integer- customername: String- dateofrent: String- dateofreturn: String- days: Integer- number: Integer- size: Integer- textfieldPrice: JTextField- textfieldYear: JTextField- textfieldmileage: JTextField- textfielddescription: JTextField- textfieldfee: JTextField- textfieldrate: JTextField- textfieldname: JTextField- textfielddate: JTextField- textfieldrdate: JTextField- textfieldday: JTextField- textfieldcar: JTextField- labelPrice: JLabel- labelYear: JLabel- labelmileage: JLabel- labeldescription: JLabel- labelfee: JLabel

<ul style="list-style-type: none">- labelrate: JLabel- labelname: JLabel- labeldate: JLabel- labelrdate: JLabel- labelday: JLabel- labelCar: JLabel- buttoncartobuy: JButton- buttoncartorent: JButton- buttonbuycar: JButton- buttonrentcar: JButton- buttonreturncar: JButton- buttondisplay: JButton- buttonclear: JButton
<ul style="list-style-type: none">+ CarCompany(): void+ main(): static void+ actionPerformed(): void+ cartobuy(): void+ cartorent(): void+ carTobuy(): void+ rentcar(): void+ returncar(): void+ display(): void+ clear(): void+ getSource(): string+ getText(): string+ setText(): string+ getCarNumber(): integer

Table 1 Class Diagram of Car Company

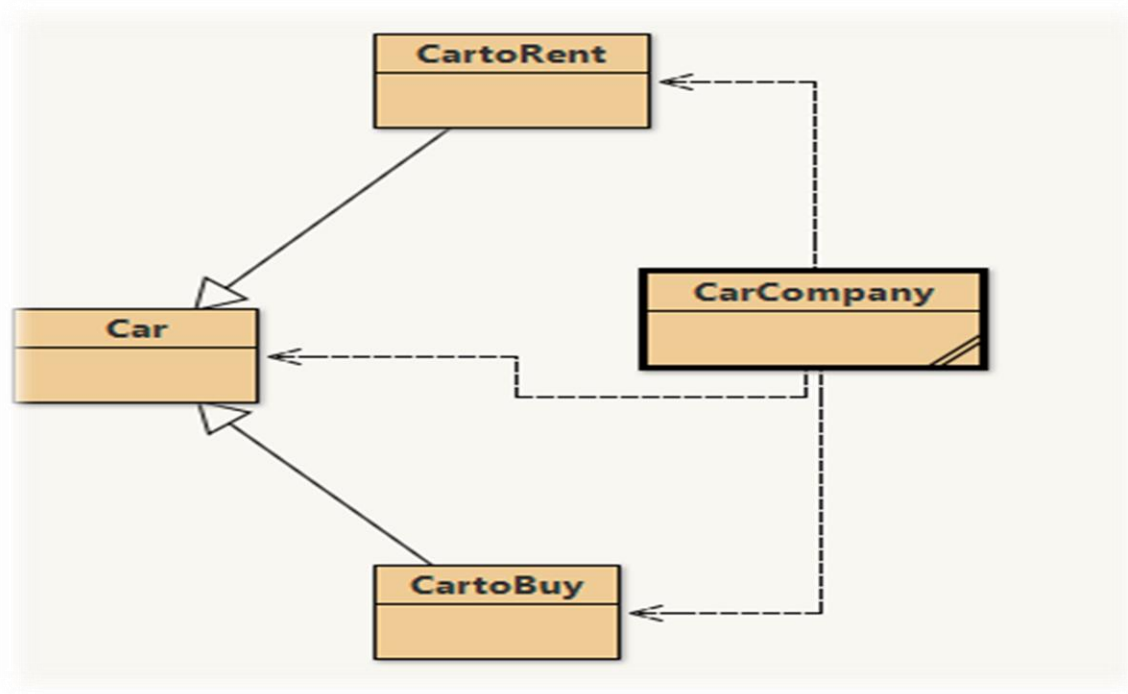


Figure 1 Class diagram of the project shown in Blue J

Pseudocode

Add Car to Buy

cartobuy()

TRY

SET description **TO** value inserted by user in description textfield

SET Price **TO** value inserted by user in description textfield

SET Year **TO** value inserted by user in description textfield

SET mileage **TO** value inserted by user in description textfield

IF (Price **is not equal to** 0 **AND** Year **is not equal to** 0 **AND** mileage **is not equal to** 0 **AND** description **is not equal to** empty)

SET VechileList **TO** ArrayList (new CartoBuy (description,Price,Year,mileage))

SHOW pop message "Values Inserted."

ELSE

SHOW pop message "Check entered values."

END IF

CATCH error (NumberFormatException nfe)

SHOW pop message "Invalid Entry"

Add Car to Rent

cartorent()

TRY

SET description **TO** value inserted by user in description textfield

SET fee **TO** value inserted by user in fee textfield

SET rate **TO** value inserted by user in rate textfield

IF (fee **is not equal to** 0 **AND** rate **is not equal** 0 **AND** description **is not equal to** empty)

SET VechileList **TO** ArrayList (new CartoRent(description,fee,rate))

SHOW pop message "Values Inserted."

ELSE

SHOW pop message "Invalid Input"

END IF

CATCH error (NumberFormatException nfe)

SHOW pop message "Check Your Entry"

Buy Car

carTobuy()

SET customername **TO** value inserted by user in customername textfield

IF (CarNumber **is part of** CartoBuy)

SET CartoBuy **TO** carTobuy **equals to** CartoBuy **AND** CarNumber

SET carTobuy **TO** element of CartoBuy's customername

END IF

Rent Car

rentcar()

SET customername **TO** value inserted by user in customername textfield

SET dateofrent **TO** value inserted by user in dateofrent textfield

SET dateofreturn **TO** value inserted by user in dateofreturn textfield

SET days **TO** value inserted by user in days textfield

IF (CarNumber is of CartoRent)

CartoRent carTorent **equals to** CartoRent's CarNumber

carTorent **TO** element of CartoRent (customername,dateofrent,dateofreturn,days)

SHOW pop message "You Successfully rented the car."

ELSE

SHOW pop message "Check entered values."

END IF

CATCH error (NumberFormatException nfe)

SHOW pop message "Check Your Entry"

Return Car

returncar()

IF(CarNumber is part of CartoRent)

SET CartoRent **TO** carTorent **equals to** CartoRent **AND** CarNumber();

SET carTorent **TO** element of returnCar

END IF

Display All

display()

FOR(Car cars : VechileList)

PRINT(Car Number **AND** elements of VechileList);

IF(cars instanceof CartoRent)

SET CartoRent **TO** cartorent **equal to** CartoRent cars

cartorent display()

IF (cars instanceof CartoBuy)

CartoBuy cartobuy **equal to** (CartoBuy)cars

```
cartobuy.display()
```

Method Description

Name of the Method	Description of Method
main	This method is used to compile and run the class in command prompt.
actionperformed	This method invokes the button when it is called upon and certain works are carried with respective methods related to the buttons.
cartobuy	This method is for adding a car to buy for the user with some values and store it as per the arrangement of values in the car to buy class.
cartorent	This method is for adding a car to rent for the user with some values and store it as per the arrangement of values in the car to rent class.
carTobuy	This method is for buying or selling a car from the cars added to the car to buy list for the user.
rentcar	This method is for renting a car from the cars added to the car to rent list for the user.
returncar	This method is for returning the car rented by the user.
display	This method displays all of the print statement in it with some correct conditions.
clear	In this method, all the text field are set to null when the clear button is clicked.
CarCompany	This is the method where a frame is created with some label and textfield.
getSource	This method is for getting the source and which returns J button for the particular button mentioned.
getText	This method is for getting the value in the particular text field and which returns a string value.
setText	This method is for setting the text field to desired value and in this program, it is for setting text field to null.

getCarNumber	This method is for getting car number and which returns a integer value.
--------------	--

Table 2 Short Description of all methods

Testing

Test 1: 1.1: Adding a car to buy

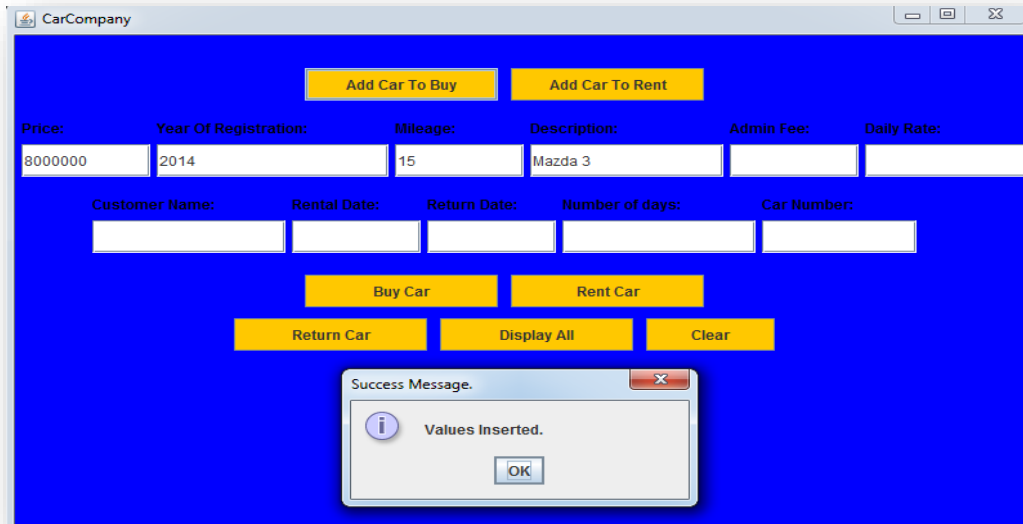


Figure 2 Inserting value for adding a car to buy

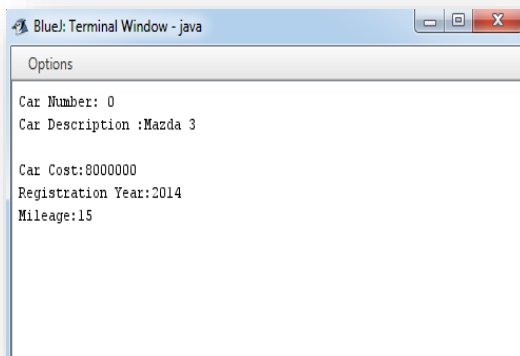


Figure 3 Displaying the Car added

Test Name:	Adding a car to buy.
Action:	Inserting Values in price, year of registration, mileage and description for adding a car for selling
Result:	Values are accepted and the inserted values are being displayed in Terminal.

Table 3 Test Table 1

1.2: Adding a car to rent

Figure 4 Inserting values to add a car to rent

```

Options
Car Number: 0
Car Description :Mazda 3

Car Cost:8000000
Registration Year:2014
Mileage:15
Car Number: 1
Car Description :Toyota Yaris

Daily Rate: Rs.5000
Admin Fee: Rs.1500

```

Figure 5 Displaying the car added for renting

Test Name:	Adding a Car to Return
Action:	Inserting values in text field of description, admin fee, Daily rate for adding a car for rent.
Result:	Values are accepted and inserted values are displayed in Terminal.

Table 4 Test Table 2

1.3: Selling a car

The screenshot shows a Java Swing window titled "CarCompany". It has a blue background. At the top, there are two yellow buttons: "Add Car To Buy" and "Add Car To Rent". Below these, there are six input fields with labels: "Price:", "Year Of Registration:", "Mileage:", "Description:", "Admin Fee:", and "Daily Rate:". The "Customer Name:" field contains the text "Prayash Bikram Shah". Below the input fields, there are three yellow buttons: "Buy Car", "Rent Car", and "Return Car". At the bottom, there are three more yellow buttons: "Display All" and "Clear".

Figure 6 Inserting values to buy a car

The screenshot shows a BlueJ Terminal Window titled "BlueJ: Terminal Window - java". It displays the following output:

```
Options
Car Number: 0
Car Description :Mazda 3
Client's Name: Prayash Bikram Shah
Car Cost:8000000
Registration Year:2014
Mileage:15
Car Number: 1
Car Description :Toyota Yaris

Daily Rate: Rs.5000
Admin Fee: Rs.1500
```

Figure 7 Display of after car bought

Test Name:	Selling a car.
Action:	Inserting values of the customer buying the car and the car number
Result:	Values are accepted and the is bought by the inserted customer name as shown in terminal.

Table 5 Test Table 3

1.4: Renting a car

The screenshot shows a Java Swing window titled "CarCompany" with a blue background. At the top, there are two yellow buttons: "Add Car To Buy" and "Add Car To Rent". Below these are six input fields for car details: "Price:", "Year Of Registration:", "Mileage:", "Description:", "Admin Fee:", and "Daily Rate:". Underneath these are five input fields for rental details: "Customer Name:" (containing "Pratik Bikram Shah"), "Rental Date:" (containing "14th feb 2917"), "Return Date:" (containing "18th feb 2017"), "Number of days:" (containing "4"), and "Car Number:" (containing "1"). At the bottom, there are three yellow buttons: "Buy Car", "Rent Car", and "Return Car", followed by "Display All" and "Clear".

Figure 8 Inserting values to rent a car

The screenshot shows a BlueJ Terminal Window titled "BlueJ: Terminal Window - java". It displays the following output:

```
Options
Car Number: 0
Car Description :Mazda 3
Client's Name: Prayash Bikram Shah
Car Cost:8000000
Registration Year:2014
Mileage:15
Car Number: 1
Car Description :Toyota Yaris
Client's Name: Pratik Bikram Shah
Daily Rate: Rs.5000
Admin Fee: Rs.1500
Rental Date:14th feb 2917
Return Date:18th feb 2017
Number of Days:4
```

Figure 9 Display after renting a car

Test Name:	Renting a Car.
Action:	Inserting the values of Customer name, rental date, return date, number of days and car number for renting a car.
Result:	Values are being accepted and the car number inserted is being rented by the customer.

Table 6 Test Table 4

1.5: Returning a car

Figure 10 Inserting value to return the rented car

```
Options
Car Description :Mazda 3
Client's Name: Prayash Bikram Shah
Car Cost:8000000
Registration Year:2014
Mileage:15
Car Number: 1
Car Description :Toyota Yaris
Client's Name: Pratik Bikram Shah
Daily Rate: Rs.5000
Admin Fee: Rs.1500
Rental Date:14th feb 2017
Return Date:18th feb 2017
Number of Days:4
Car Number: 0
Car Description :Mazda 3
Client's Name: Prayash Bikram Shah
Car Cost:8000000
Registration Year:2014
Mileage:15
Car Number: 1
Car Description :Toyota Yaris
Daily Rate: Rs.5000
Admin Fee: Rs.1500
```

Figure 11 Display after returning the rented car

Test Name:	Returning a Car.
Action:	Inserting the car number that is being rented for returning the car.
Result:	Inserted car number has been returned and is again available for renting.

Table 7 Test Table 5

1.6: Displaying all of the cars

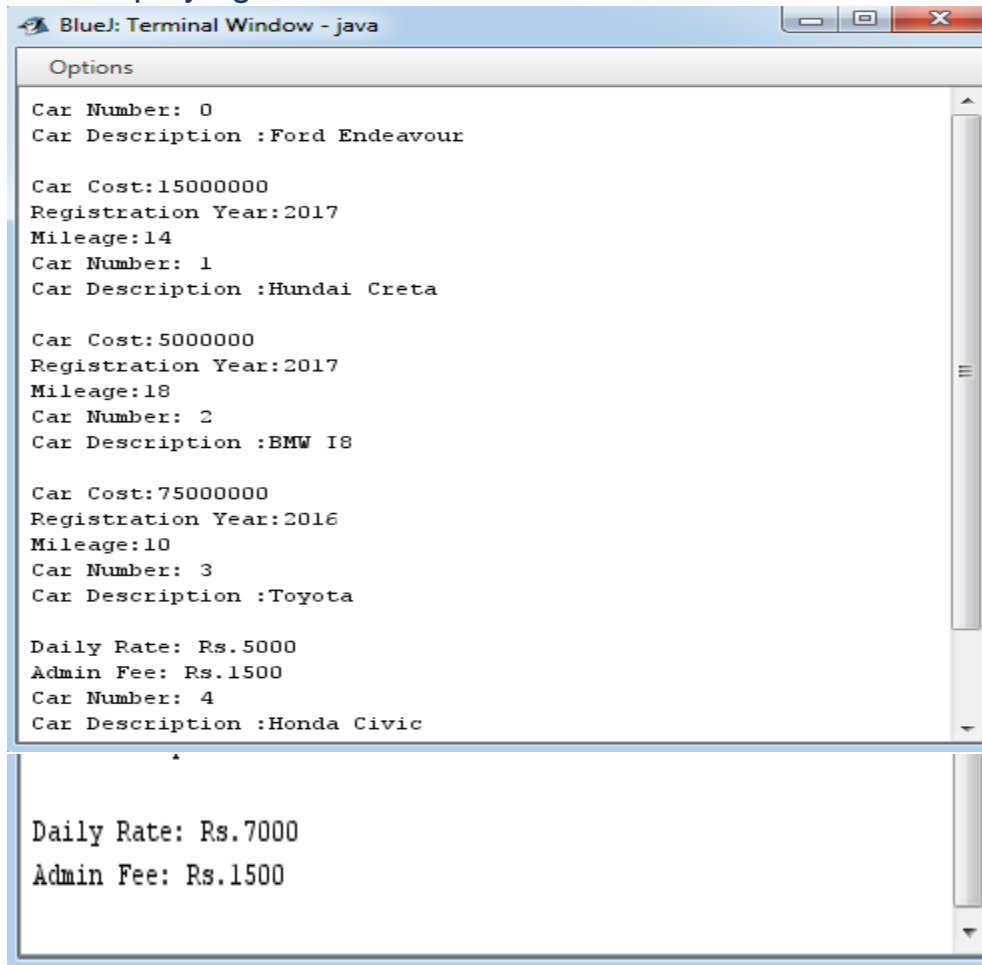


Figure 12 Displaying all the cars added

Test Name:	Displaying all of cars.
Action:	Inserting values for adding cars for buying and renting
Result:	All inserted cars are being displayed on the terminal after clicking display button.

Table 8 Test Table 6

Test 2: Dialog Boxes

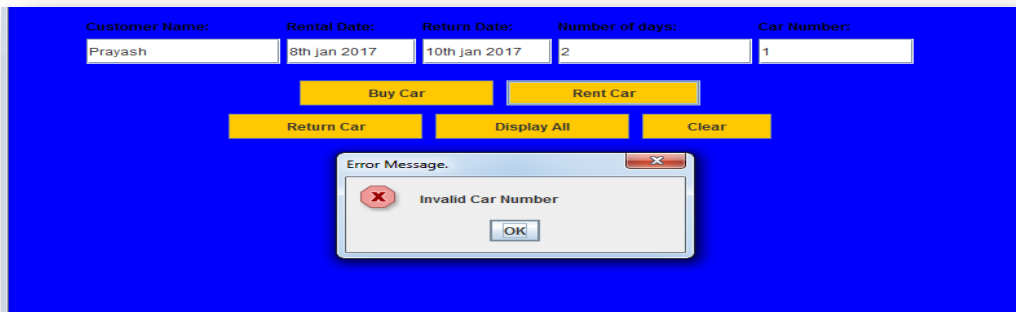


Figure 13 Dialog Box showing error message

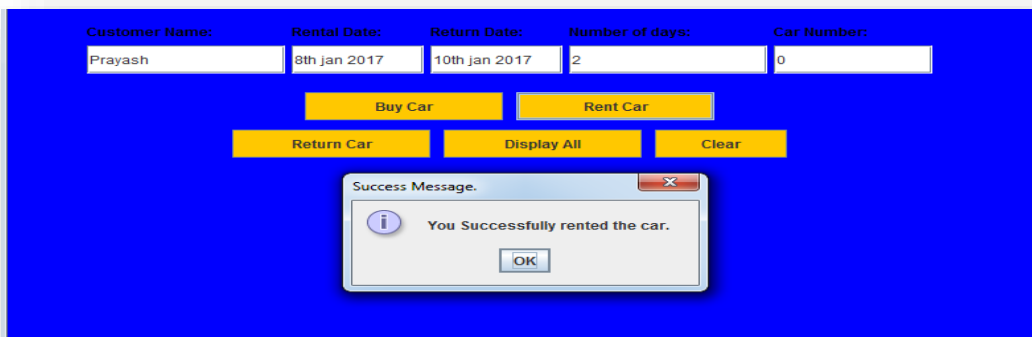


Figure 14 Dialog Box Showing Success Message

Test Name:	Dialog Boxes.
Action:	Inserting Invalid car number so that the error message pops up and again inserting the right car number so that success message pops up.
Result:	Error message pops up when inserting car number that does not exist and success message pops up when right car number is being inserted.

Table 9 Test Table 7

Error Detection:

Error 1:

This error is a Syntax error which occurred when compiling the codes in action performed method in in which there are expected identifier and was not mentioned in it as shown in figure 15.

```
public static void main(String[] args) {
    new CarCompany();
}
@Override
public void actionPerformed(ActionEvent ) {
```

Figure 15 Syntax Error in while Coding

As there was and expected identifier after Action Event “e” was added as shown in the figure 16 and after doing this the program was compiled successfully.

```
@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == buttoncartobuy) {
```

Figure 16 Syntax error solved

Error 2:

Figure 17 Runtime error showing a button not working

The above figure 17 shows that values were being inserted in the four of the field to add to car to buy. And after values were being submitted when I clicked clear button the text

fields did not clear. This was due to missing a code in a part which is shown in below figure 18.

```
buttonclear = new JButton("Clear");
buttonclear.setBounds(490,260,100,30);
buttonclear.setBackground(Color.ORANGE);
FRAME.add(buttonclear);
```

Figure 18 Code for the button that was not working

As there was a line of code missing in the figure 18 due to which clear button was not working, a line was added on that part of the code as shown in the figure 19.

```
buttonclear = new JButton("Clear");
buttonclear.setBounds(490,260,100,30);
buttonclear.addActionListener(this);
buttonclear.setBackground(Color.ORANGE);
FRAME.add(buttonclear);
```

Figure 19 Solved Code



Figure 20 Result After solving the problem of the button code

After adding the line of the code when the clear button was pressed all the values that were present in the various text field was cleared and all of the text field was set to null.

Error 3:

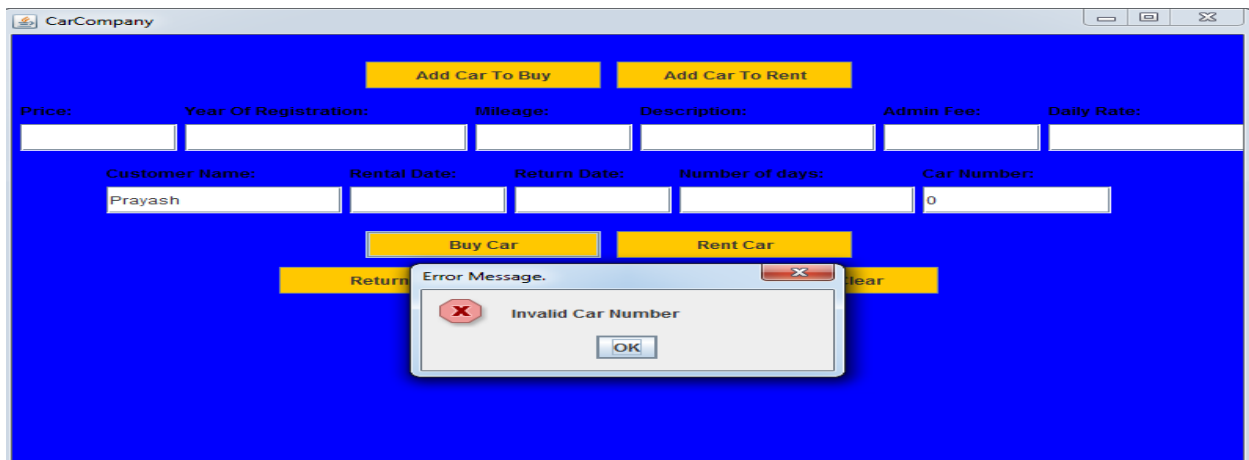


Figure 21 Runtime Error of not taking the input that was on the list

This error was occurred when I tried to buy a car from the list of cars added to buy car. The car that I tried to buy was number 0 and when I inserted name and car number and clicked on “Buy Car” button a error message popped up saying invalid car number. There were the cars inserted before but the car number 0 was not allowed to access as shown in figure 21.

The above error was a runtime error which occurred while I was checking the buttons work. This runtime error was caused due to an error in an if statement which was used in Car Number method as shown in figure 22 below. I had coded number is greater or equal to 1 which was causing the error.

```
public int getCarNumber(){
    int number = 0;
    int size = VechileList.size();
    try{
        number = Integer.parseInt(textfieldcar.getText());
        if (number >=1 && number < size){
            return number;
        } else{
            JOptionPane.showMessageDialog(FRAME,
                "Invalid Car Number",
                "Error Message.",
                JOptionPane.ERROR_MESSAGE);
        }
    }
    catch (NumberFormatException nfe) {
        JOptionPane.showMessageDialog(FRAME,
            "Invalid Entry",
            "Error Message.",
            JOptionPane.ERROR_MESSAGE);
    }
    return 0;
}
```

Figure 22 Part of coding where the problem was present


```

public int getCarNumber(){
    int number = 0;
    int size = VechileList.size();
    try{
        number = Integer.parseInt(textfieldcar.getText());
        if (number >=0 && number < size){
            return number;
        } else{
            JOptionPane.showMessageDialog(FRAME,
                "Invalid Car Number",
                "Error Message.",
                JOptionPane.ERROR_MESSAGE);
        }
    }
    catch (NumberFormatException nfe) {
        JOptionPane.showMessageDialog(FRAME,
            "Invalid Entry",
            "Error Message.",
            JOptionPane.ERROR_MESSAGE);
    }
    return 0;
}

```

Figure 23 Solved the problem in the code

For Solving this error, I altered my code and corrected the if statement that I had used in the car number method. All I did was changed that case to number is greater or equal to 0 instead of 1 as shown in figure 23 and the problem was solved. After compiling the class with alter of the code there was no problem accessing the car number 0 anymore as shown in the figure 24.

Figure 24 Solved the runtime error

Conclusion

This Coursework was about creating a class to the previous assignments classes making a Graphical user Interface which takes input and stores cars for selling and renting a car. This assessment was to create a GUI for a car company using Java Programming language. Not just only creating GUI was being done, I had to use array list for storing the cars for sale and rent to this class. All the necessary topics for completing this assignment was taught, I learned a lot from lecture classes of Mr. Rabin Regmi sir and all those topics were implemented and faced in the tutorial and lab classes that were taught by Mr. Weenit Maharjan sir. Creating a frame, button, label wasn't easy as it was the first time that I was doing a programming that made a Graphical User Interface. Mr. Weenit Maharjan sir gave some ideas on lab classes and made the class to practice some question related to coursework which helped a lot to complete this coursework. As I read the question we could modify the layout I researched about changing colors of the frame, button and fonts which then I used on my layout and made a different layout from that was given to us. While doing this project, I went through all the slides tutorial question and every example that I learned or went across while learning about array list, JFrame, JLabel, JButton and other topics which helped me to do this project in a proper manner and easily. Creating this project tested my capabilities to research and my idea of creativity while changing the layout and it was like revising what I had learned in my classes and even helped me learning the things that I had missed out not attending some of the classes. My assignment may have some flaws and some drawbacks but overall I was able to create the GUI and the requirements that was mentioned in question. It was a great opportunity to learn many new things and go through and implement what I had been learning through this Programming module taught in Islington College.

Appendix:

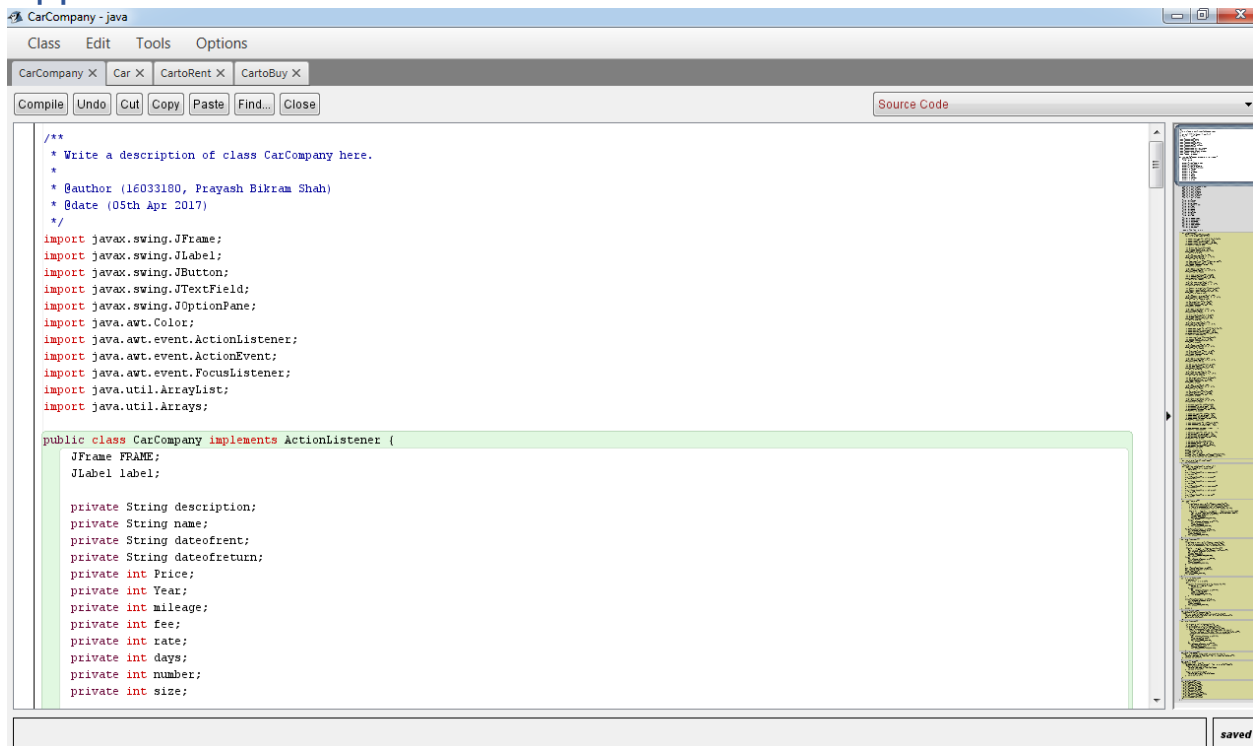


Figure 25 Source Code 1

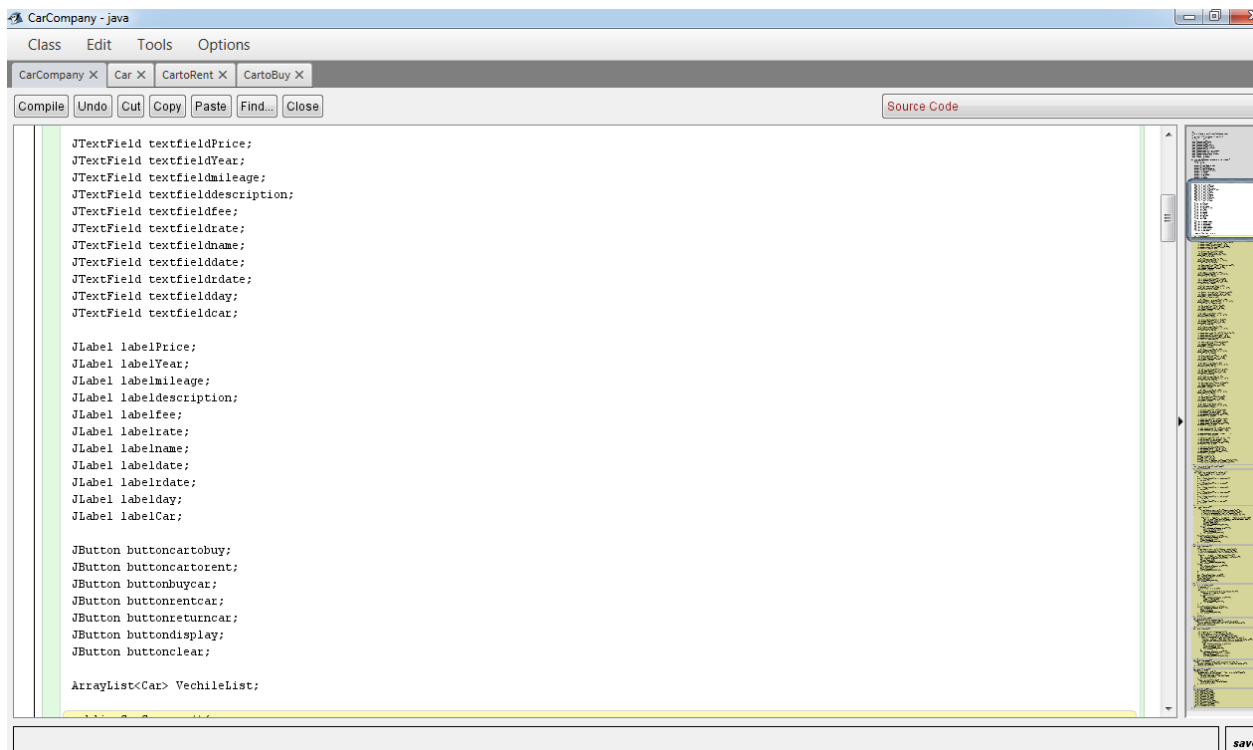


Figure 26 Source Code 2

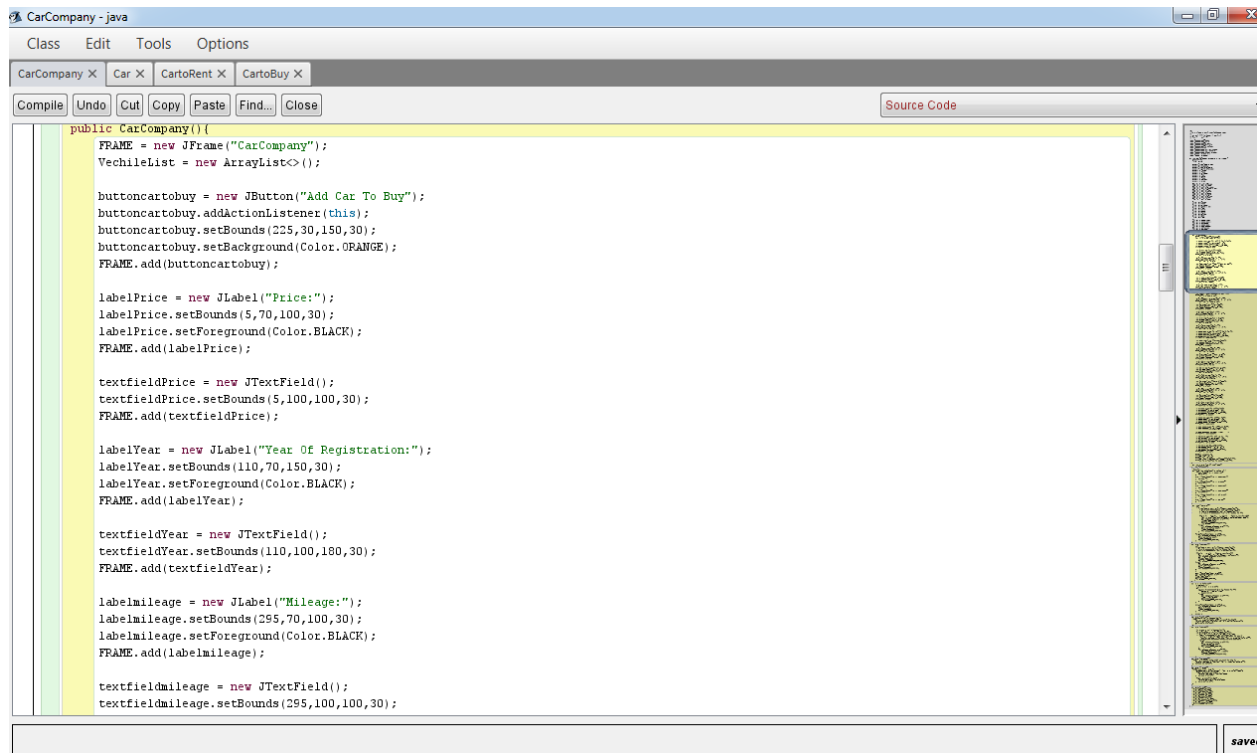


Figure 27 Source Code 3

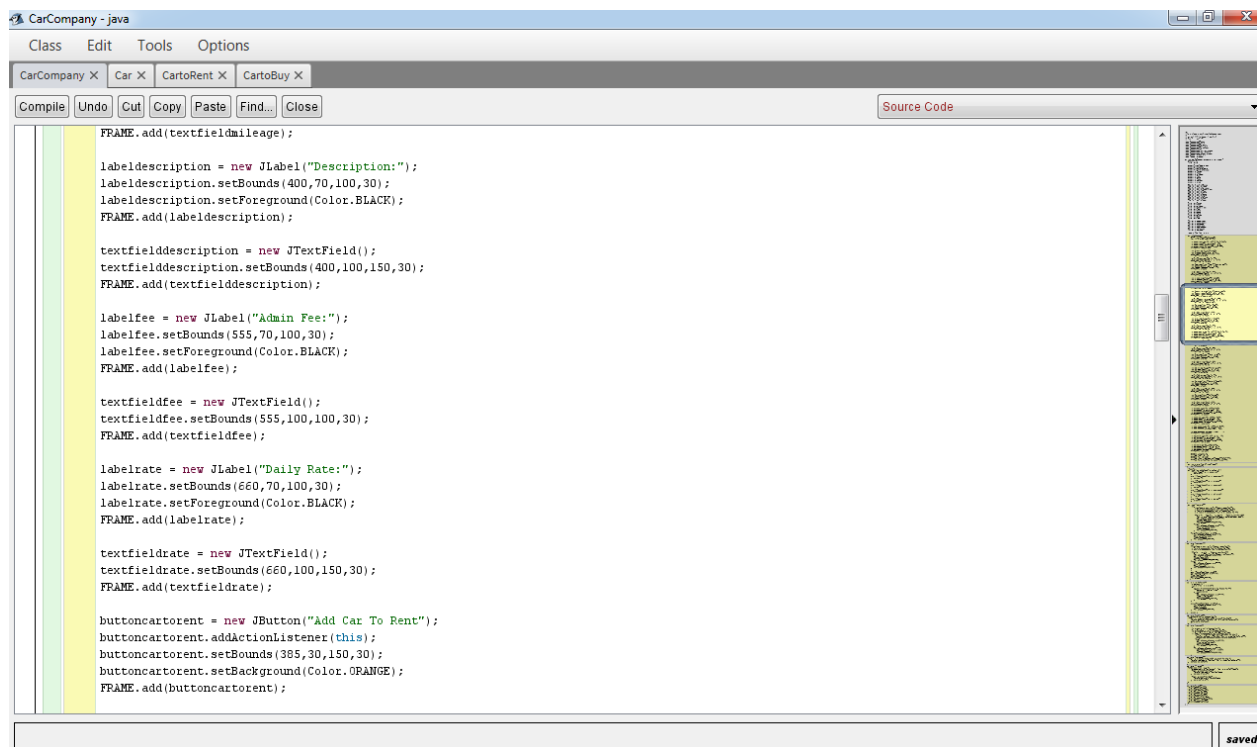


Figure 28 Source Code 4

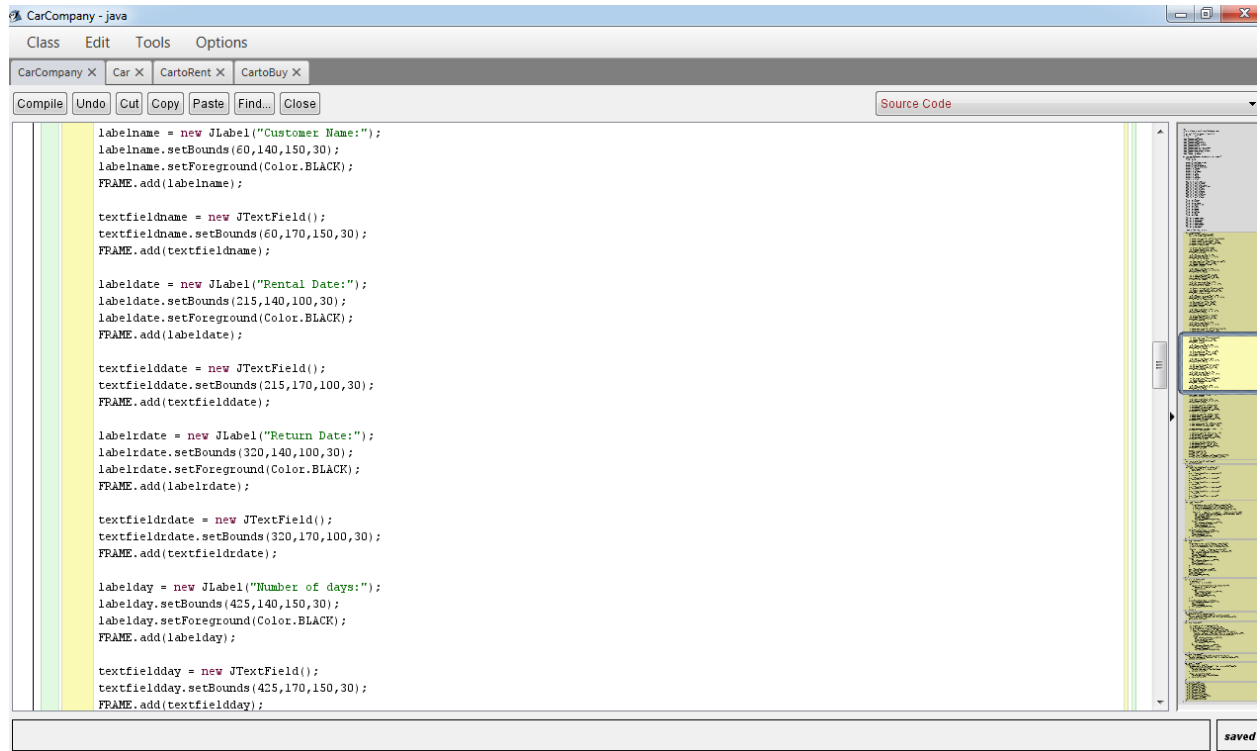


Figure 29 Source Code 5

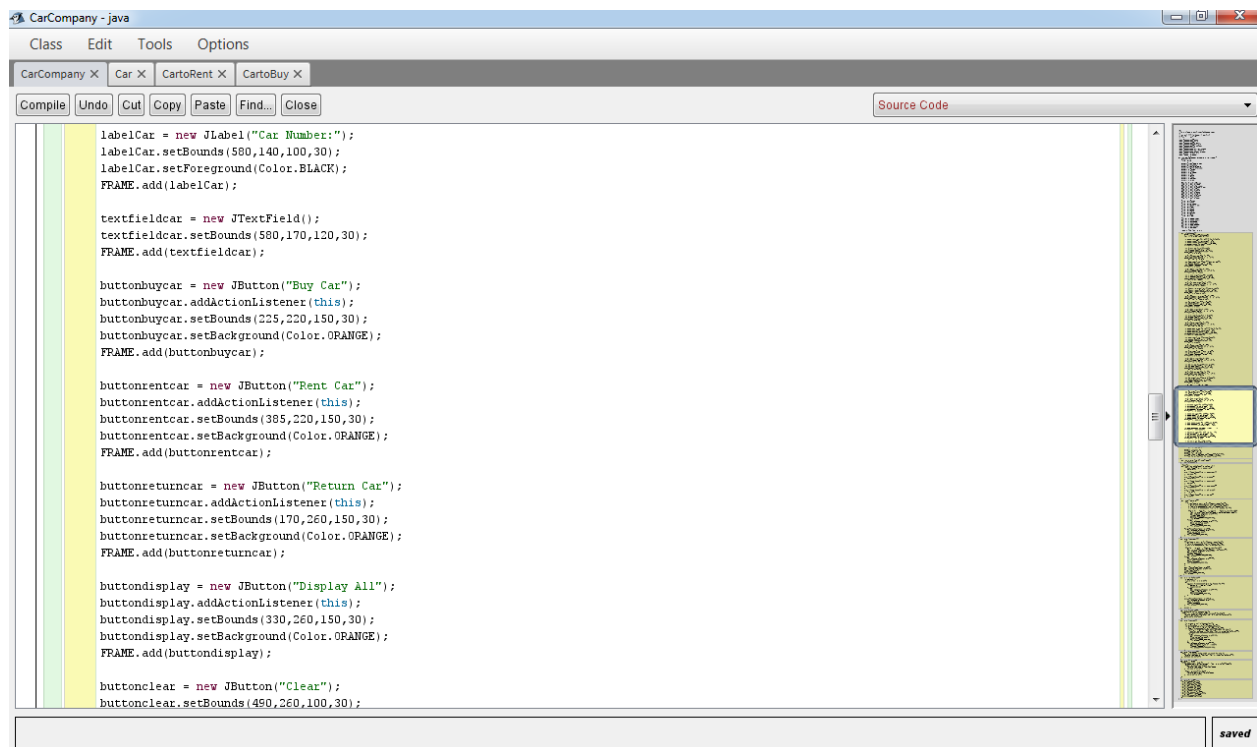


Figure 30 Source Code 6

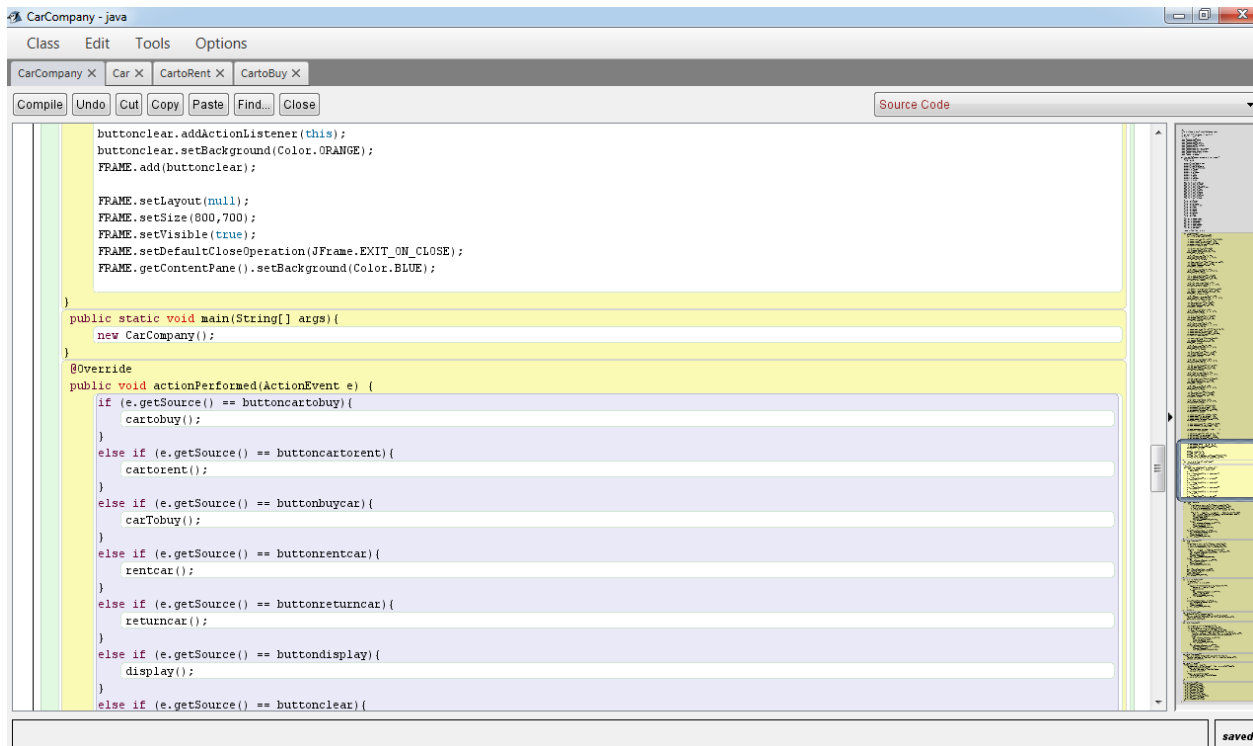


Figure 31 Source Code 7

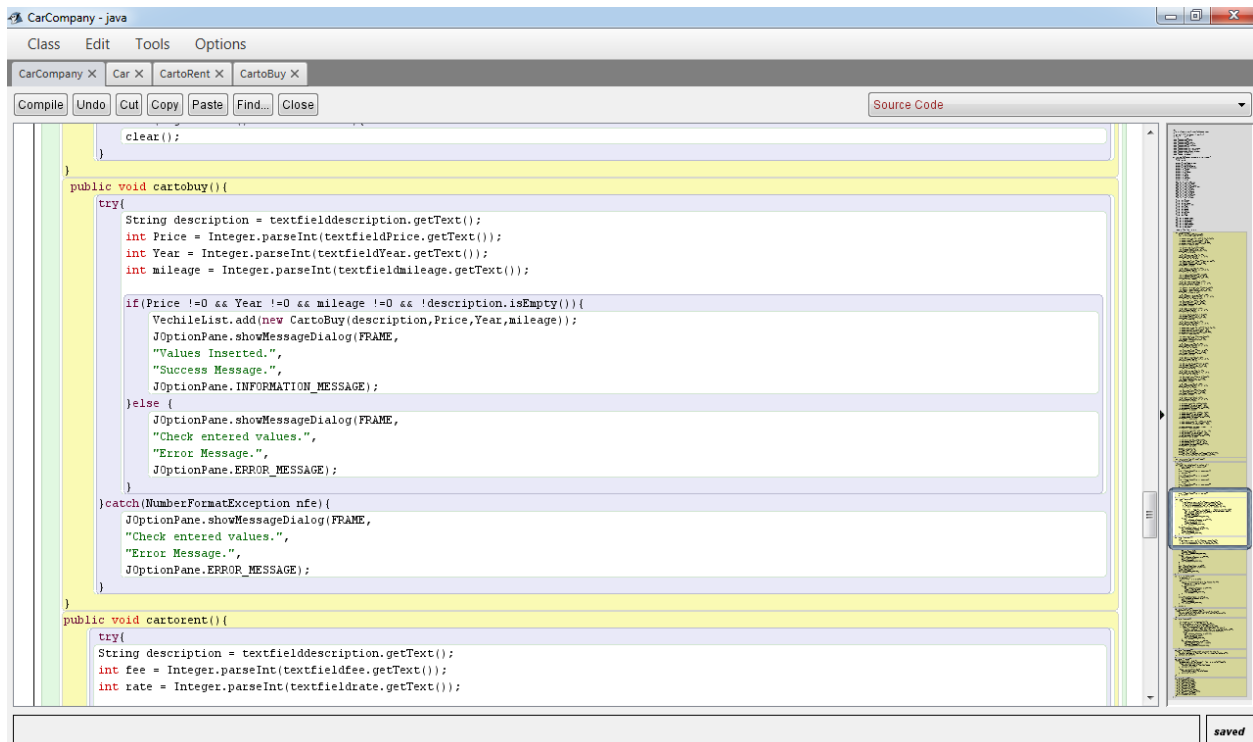


Figure 32 Source Code 8

```

CarCompany - java
Class Edit Tools Options
CarCompany X Car X CartoRent X CartoBuy X
Compile Undo Cut Copy Paste Find... Close Source Code

if (fee != 0 && rate != 0 && !description.isEmpty()) {
    VchileList.add(new CartoRent(description, fee, rate));
    JOptionPane.showMessageDialog(FRAME,
        "Values Inserted.",
        "Success Message.",
        JOptionPane.INFORMATION_MESSAGE);
} else {
    JOptionPane.showMessageDialog(FRAME,
        "Check Your Entry",
        "Error Message",
        JOptionPane.ERROR_MESSAGE);
}
} catch (NumberFormatException nfe) {
    JOptionPane.showMessageDialog(FRAME,
        "Check Your Entry",
        "Error Message",
        JOptionPane.ERROR_MESSAGE);
}
}

public int getCarNumber() {
    int number = 0;
    int size = VchileList.size();
    try {
        number = Integer.parseInt(textfieldcar.getText());
        if (number >= 0 && number < size) {
            return number;
        } else {
            JOptionPane.showMessageDialog(FRAME,
                "Invalid Car Number",
                "Error Message",
                JOptionPane.ERROR_MESSAGE);
        }
    }
}

```

Figure 33 Source Code 9

```

CarCompany - java
Class Edit Tools Options
CarCompany X Car X CartoRent X CartoBuy X
Compile Undo Cut Copy Paste Find... Close Source Code

} catch (NumberFormatException nfe) {
    JOptionPane.showMessageDialog(FRAME,
        "Invalid Entry",
        "Error Message",
        JOptionPane.ERROR_MESSAGE);
}
return 0;
}

public void cartobuy() {
    String name = textfieldname.getText();
    if (VchileList.get(getCarNumber()) instanceof CartoBuy) {
        CartoBuy cartobuy = (CartoBuy) VchileList.get(getCarNumber());
        cartobuy.cartobuy(name);
    }
}

public void rentcar() {
    String name = textfieldname.getText();
    String dateofrent = textfielddate.getText();
    String dateofreturn = textfieldrdate.getText();
    int days = Integer.parseInt(textfieldday.getText());
    if (VchileList.get(getCarNumber()) instanceof CartoRent) {
        CartoRent cartorent = (CartoRent) VchileList.get(getCarNumber());
        cartorent.rentcar(name, dateofrent, dateofreturn, days);
    } else {
        JOptionPane.showMessageDialog(FRAME,
            "Check entered values.",
            "Error Message",
            JOptionPane.ERROR_MESSAGE);
    }
    JOptionPane.showMessageDialog(FRAME,
        "You Successfully rented the car.",
        "Success Message",
        JOptionPane.INFORMATION_MESSAGE);
}

```

Figure 34 Source Code 10

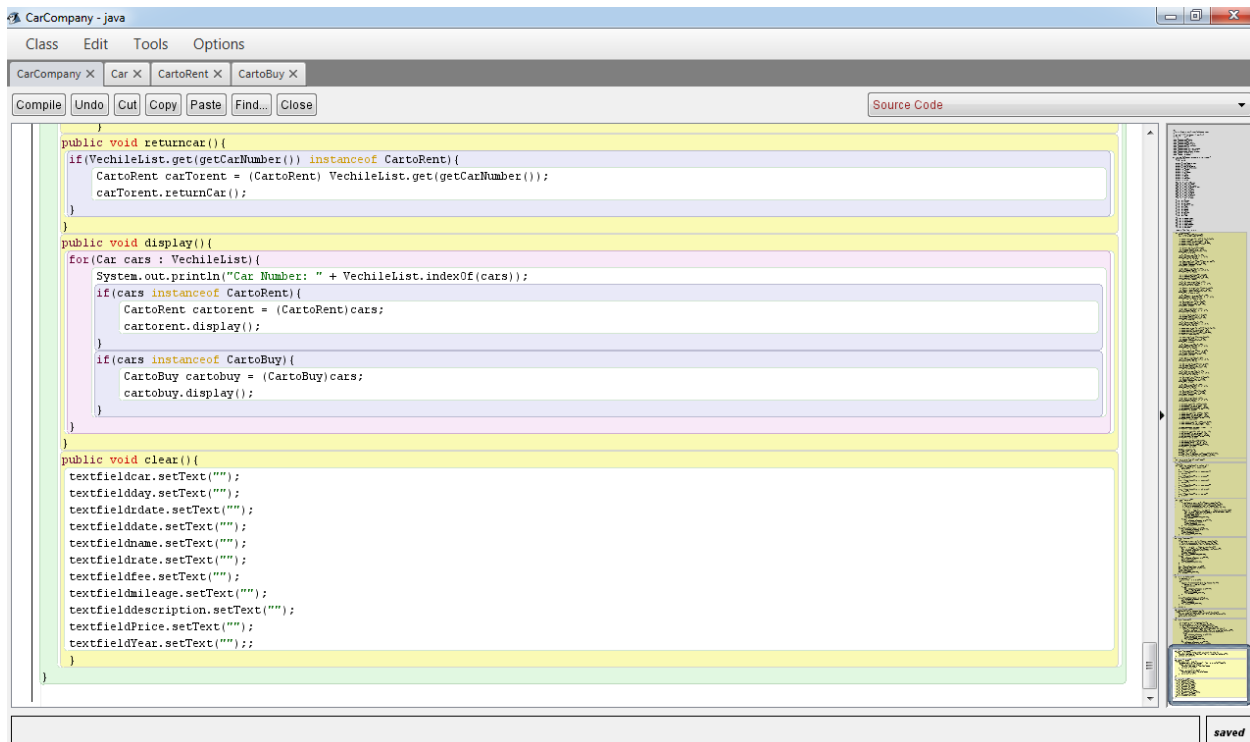


Figure 35 Source Code 11