**Used Car Warehouse Database System**

**Test Strategy**

Miao Wang

27647986

[mwan0011@student.monash.edu](mailto:mwan0011@student.monash.edu)

Date: 20/10/2017

Contents

[Test Plan 4](#_Toc496263278)

[Test steps 4](#_Toc496263279)

[Actual test 5](#_Toc496263280)

[Test constructors 5](#_Toc496263281)

[Default constructor 5](#_Toc496263282)

[Constructor with one parameter (carReg) 5](#_Toc496263283)

[Constructor with eight parameters 6](#_Toc496263284)

[Test accessors 7](#_Toc496263285)

[getCarMake() 7](#_Toc496263286)

[GetCarModel() 8](#_Toc496263287)

[GetCarReg() 8](#_Toc496263288)

[GetColour1() 8](#_Toc496263289)

[GetColour2() 9](#_Toc496263290)

[GetColour3() 9](#_Toc496263291)

[GetPrice() 10](#_Toc496263292)

[GetYearMade() 10](#_Toc496263293)

[Test mutator 11](#_Toc496263294)

[SetCarMake() 11](#_Toc496263295)

[SetCarModel() 12](#_Toc496263296)

[SetCarReg() 13](#_Toc496263297)

[SetColour1() 14](#_Toc496263298)

[SetColour2() 15](#_Toc496263299)

[SetColour3() 16](#_Toc496263300)

[SetPrice() 17](#_Toc496263301)

[SetYearMade() 18](#_Toc496263302)

[Test other methods 19](#_Toc496263303)

[ToString() 19](#_Toc496263304)

[Hashcode() 20](#_Toc496263306)

[Equals() 21](#_Toc496263307)

# Test Plan

The car class contains three constructers and 19 methods, which including 8 mutators and 8 accessors. I design positive test and nagetive test for every mutators in this test strategy. For every test unit, the test data, expected result and actual results have been recorded.

## Test steps

Create a car object with the default constructor.

Create car objects with all non-default constructors.

Test all the get methods.

Test all the set methods.

Test the other method.

# Actual test

## Test constructors

### Default constructor

Test data: no input

Expected result:

carReg = "";

yearMade = 2017;

carMake = "";

carModel = "";

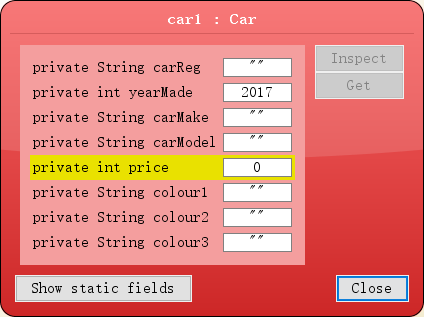
price = 0;

colour1 = "";

colour2 = "";

colour3 = "";

actual result:



### Constructor with one parameter (carReg)

Test data:

CarReg: "mater"

Expected result:

carReg = " mater ";

yearMade = 2017;

carMake = "";

carModel = "";

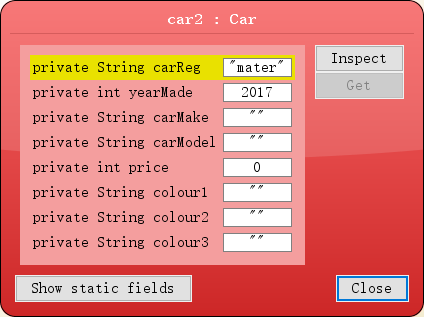
price = 0;

colour1 = "";

colour2 = "";

colour3 = "";

actual result:



### Constructor with eight parameters

Test data:

carReg = " McQ ";

yearMade = 2017;

carMake = "bmw";

carModel = "116d";

price = 50000;

colour1 = "red";

colour2 = "white";

colour3 = "yellow";

Expected result:

carReg = " McQ ";

yearMade = 2017;

carMake = "bmw";

carModel = "116d";

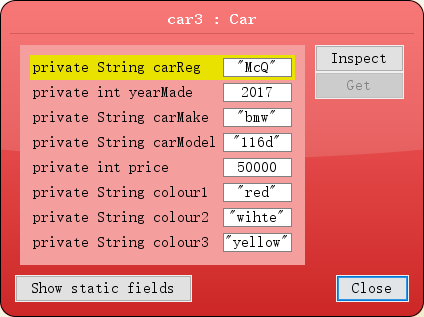
price = 50000;

colour1 = "red";

colour2 = "wihte";

colour3 = "yellow";

actual result:

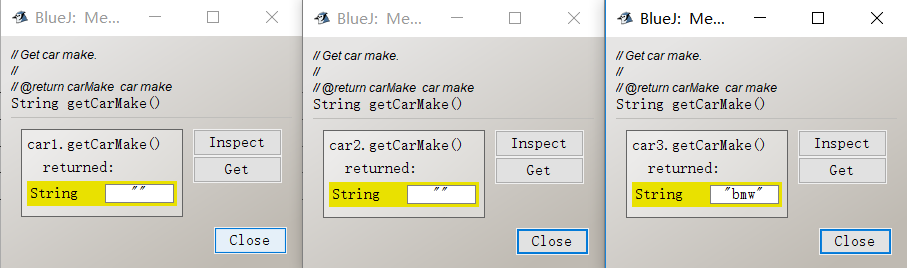


## Test accessors

### getCarMake()

|  |  |
| --- | --- |
| Test data | Expected result |
| car1 | “” |
| car2 | “” |
| car3 | “bwm” |

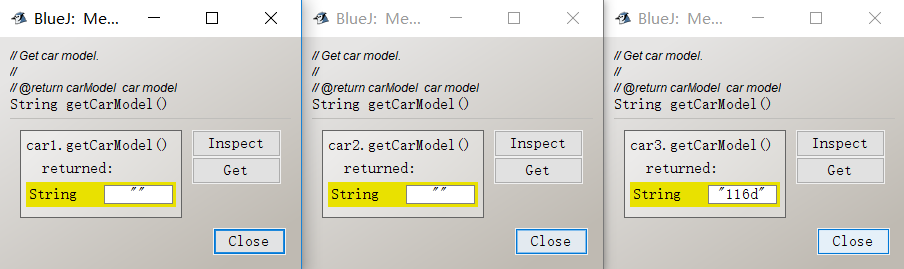
Actual result



### GetCarModel()

|  |  |
| --- | --- |
| Test data | Expected result |
| car1 | “” |
| car2 | “” |
| car3 | “116d” |

Actual result:

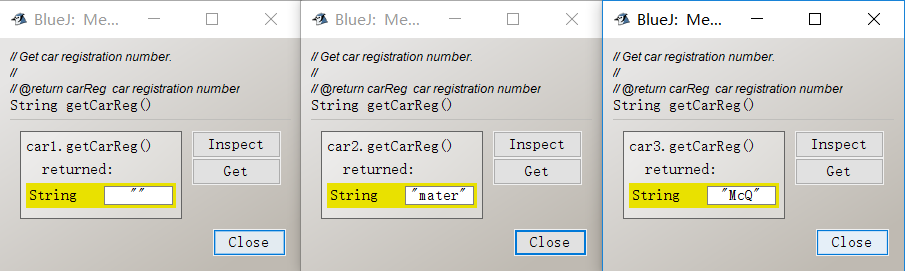




### GetCarReg()

|  |  |
| --- | --- |
| Test data | Expected result |
| car1 | “” |
| car2 | “mater” |
| car3 | “McQ” |

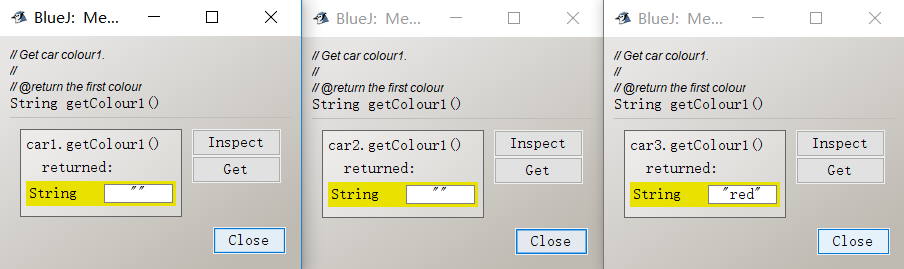
Actual result:



### GetColour1()

|  |  |
| --- | --- |
| Test data | Expected result |
| car1 | “” |
| car2 | “” |
| car3 | “red” |

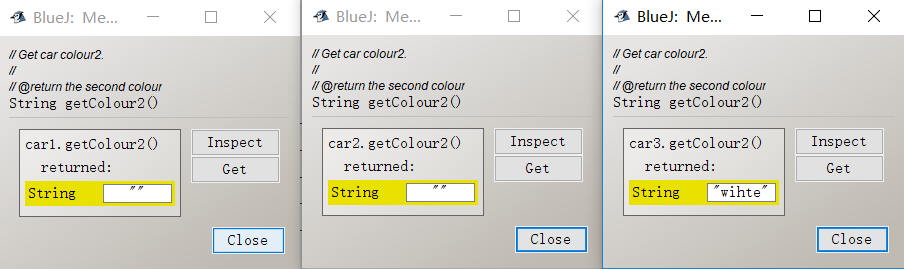
Actual result:



### GetColour2()

|  |  |
| --- | --- |
| Test data | Expected result |
| car1 | “” |
| car2 | “” |
| car3 | “wihte” |

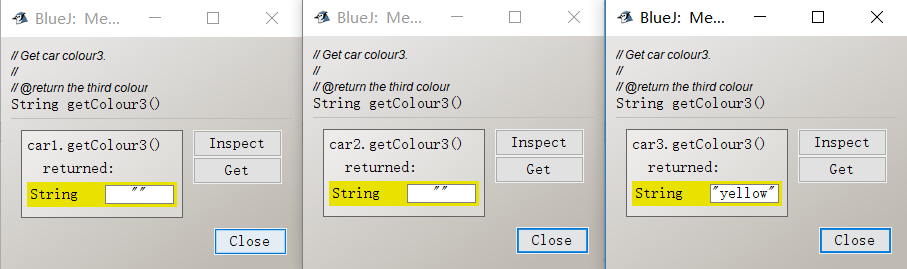
Actual result:



### GetColour3()

|  |  |
| --- | --- |
| Test data | Expected result |
| car1 | “” |
| car2 | “” |
| car3 | “yellow” |

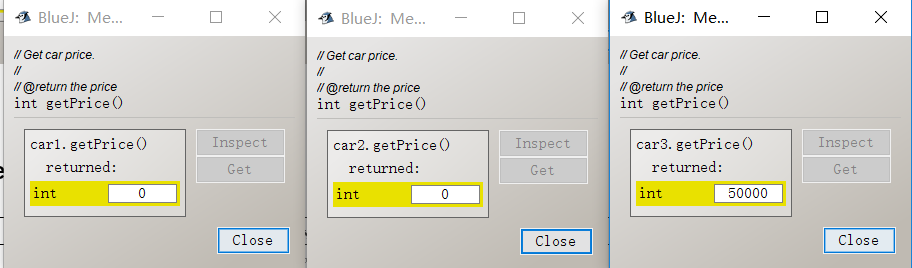
Actual result:



### GetPrice()

|  |  |
| --- | --- |
| Test data | Expected result |
| car1 | 0 |
| car2 | 0 |
| car3 | 50000 |

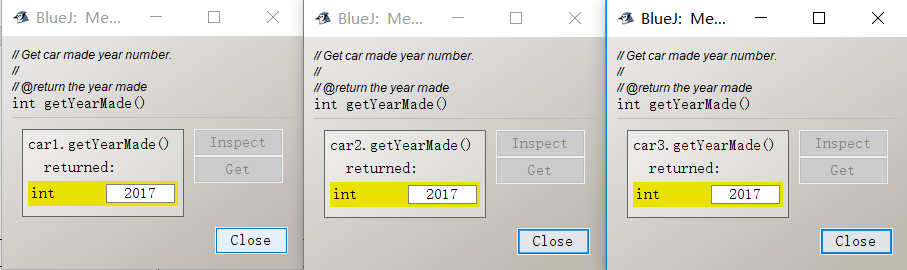
Actual result:



### GetYearMade()

|  |  |
| --- | --- |
| Test data | Expected result |
| car1 | 2017 |
| car2 | 2017 |
| car3 | 2017 |

Actual result:



## Test mutator

### SetCarMake()

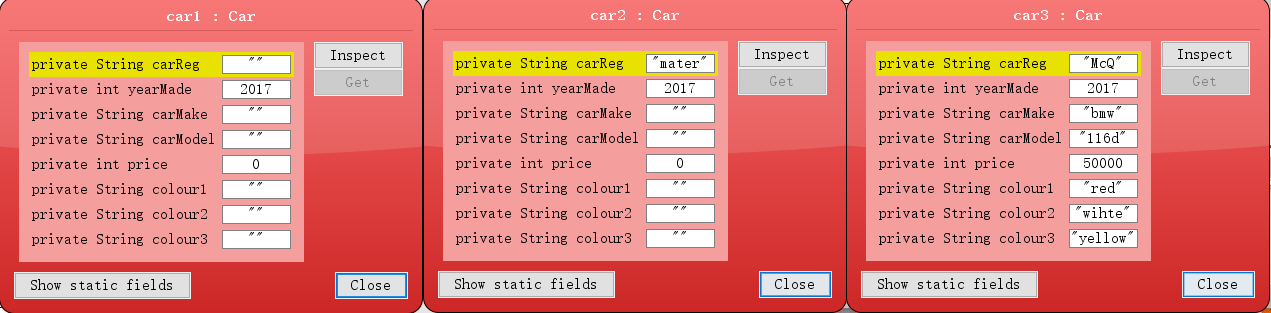
#### negative test

Test data: “+-\*/”

Expected result:

|  |  |
| --- | --- |
| car1 | “” |
| car2 | “” |
| car3 | “bmw” |

Actual result:



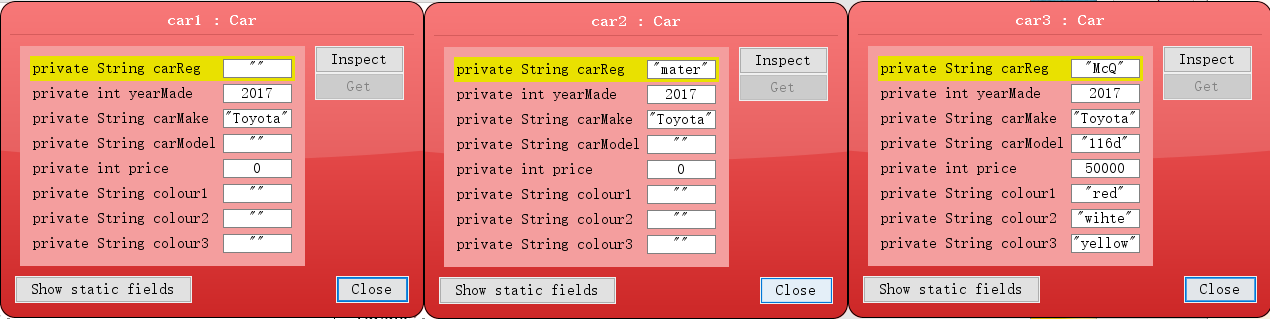
#### Positive test

Test data: “Toyota”

Expected result:

|  |  |
| --- | --- |
| car1 | “Toyota” |
| car2 | “Toyota” |
| car3 | “Toyota” |

Actual result:



### SetCarModel()

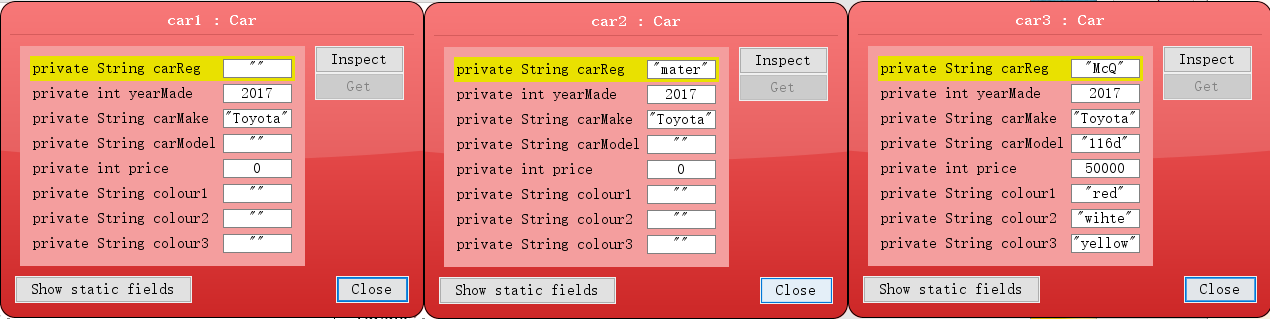
#### negative test

Test data: “+-\*/”

Expected result:

|  |  |
| --- | --- |
| car1 | “” |
| car2 | “” |
| car3 | “116d” |

Actual result:



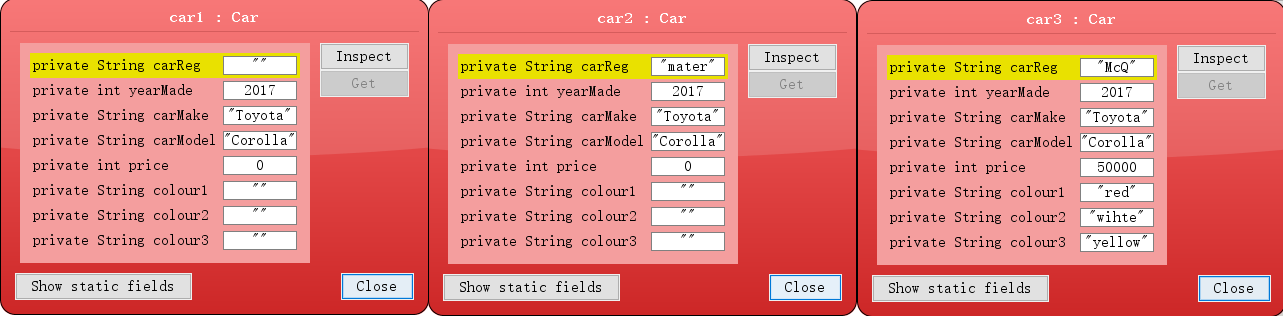
#### positive test

Test data: “Corolla”

Expected result:

|  |  |
| --- | --- |
| car1 | “Corolla” |
| car2 | “Corolla” |
| car3 | “Corolla” |

Actual result:



### SetCarReg()

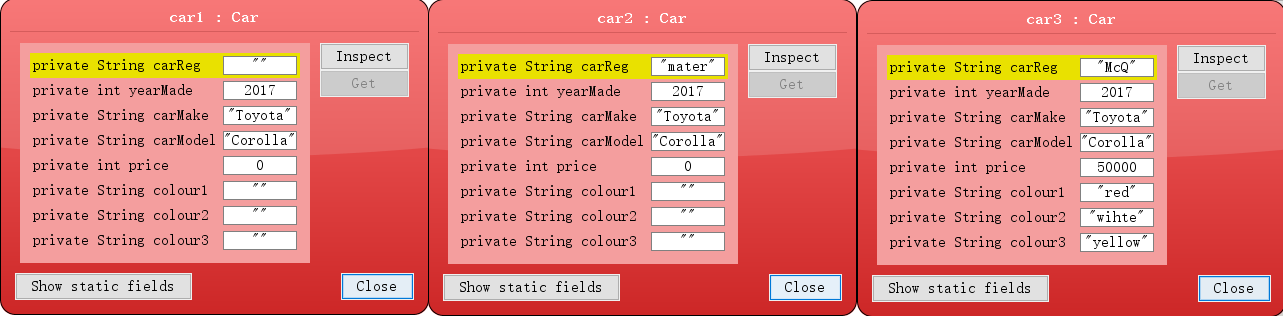
#### Negative test

Test data: “zxc 123”

Expected result:

|  |  |
| --- | --- |
| car1 | “” |
| car2 | “Mcqueen” |
| car3 | “Corolla” |

Actual result:



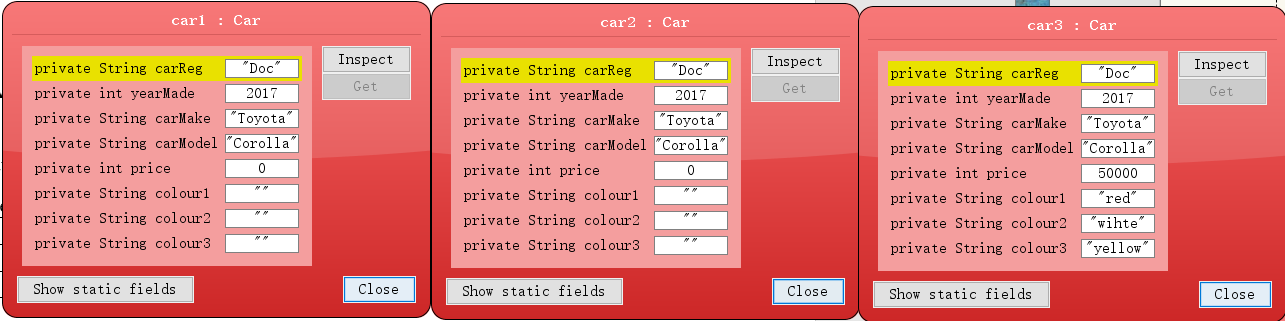
#### Positive test

Test data: “Doc”

Expected result:

|  |  |
| --- | --- |
| car1 | “Doc” |
| car2 | “Doc” |
| car3 | “Doc” |

Actual result:



### SetColour1()

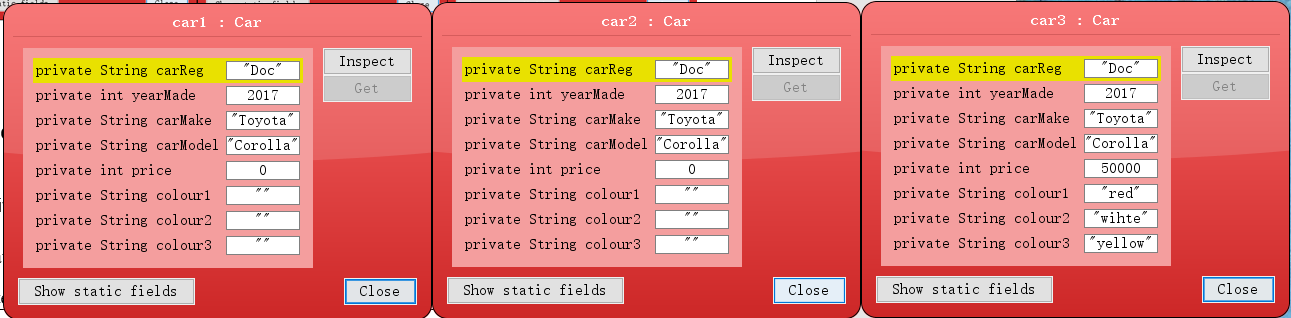
#### Negative test

Test data: “123”

Expected result:

|  |  |
| --- | --- |
| car1 | “” |
| car2 | “” |
| car3 | “red” |

Actual result:



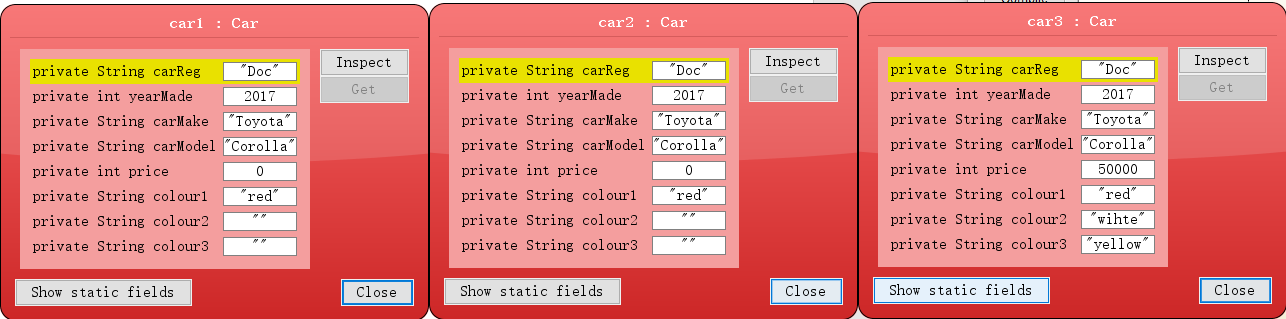
#### positive test

Test data: “red”

Expected result:

|  |  |
| --- | --- |
| car1 | “red” |
| car2 | “red” |
| car3 | “red” |

Actual result:



### SetColour2()

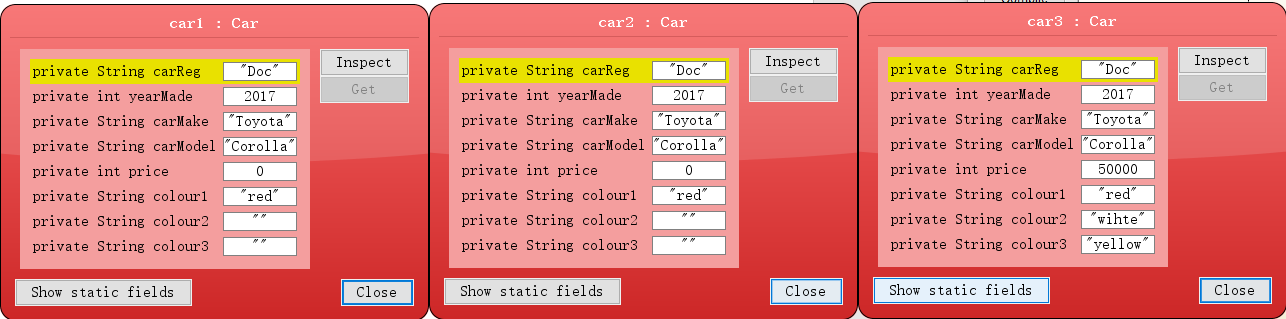
#### Negative test

Test data: “???”

Expected result:

|  |  |
| --- | --- |
| car1 | “” |
| car2 | “” |
| car3 | “wihte” |

Actual result:



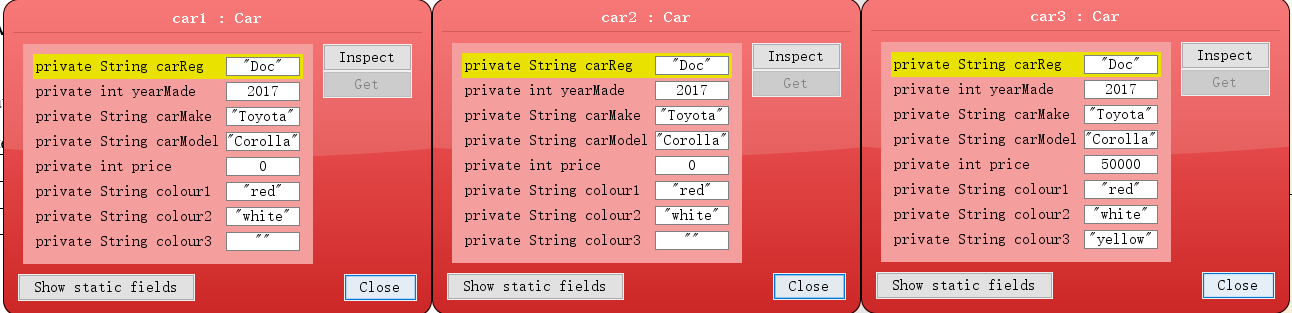
#### positive test

Test data: “white”

Expected result:

|  |  |
| --- | --- |
| car1 | “white” |
| car2 | “white” |
| car3 | “white” |

Actual result:



### SetColour3()

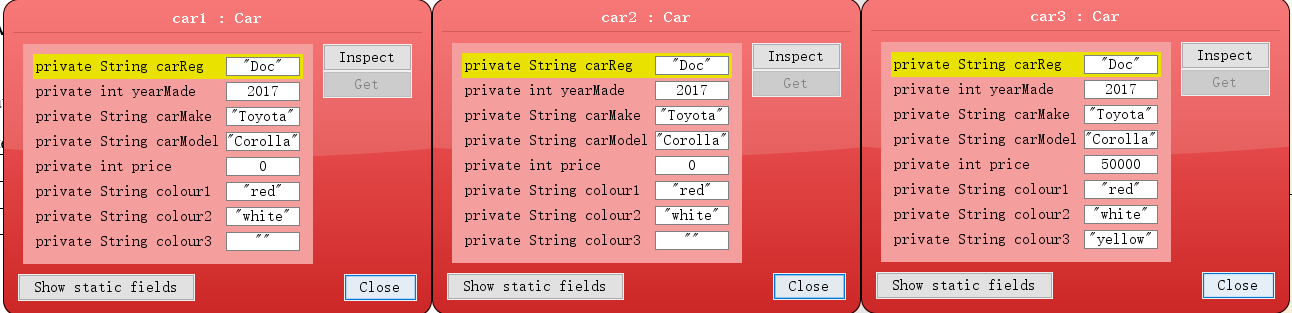
#### Negative test

Test data: “-\*/”

Expected result:

|  |  |
| --- | --- |
| car1 | “” |
| car2 | “” |
| car3 | “yellow” |

Actual result:



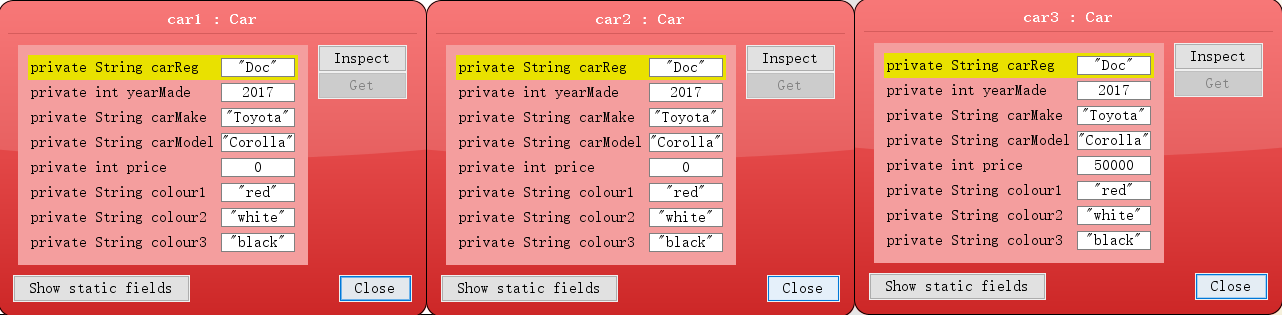
#### positive test

Test data: “black”

Expected result:

|  |  |
| --- | --- |
| car1 | “black” |
| car2 | “black” |
| car3 | “black” |

Actual result:



### SetPrice()

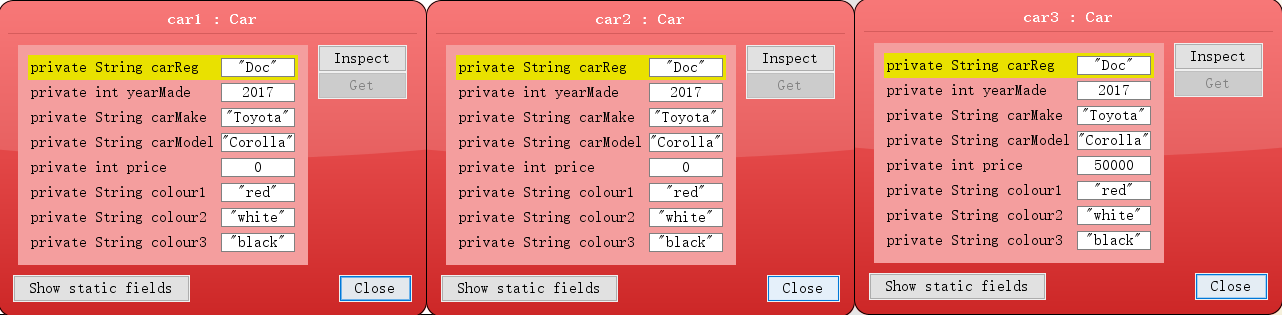
#### Negative test

Test data: “-200”

Expected result:

|  |  |
| --- | --- |
| car1 | 0 |
| car2 | 0 |
| car3 | 50000 |

Actual result:



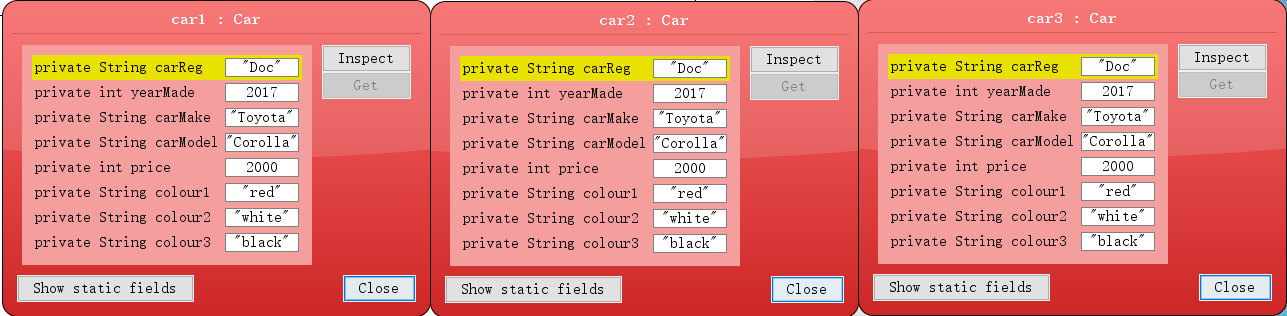
#### positive test

Test data: “2000”

Expected result:

|  |  |
| --- | --- |
| car1 | 2000 |
| car2 | 2000 |
| car3 | 2000 |

Actual result:



### SetYearMade()

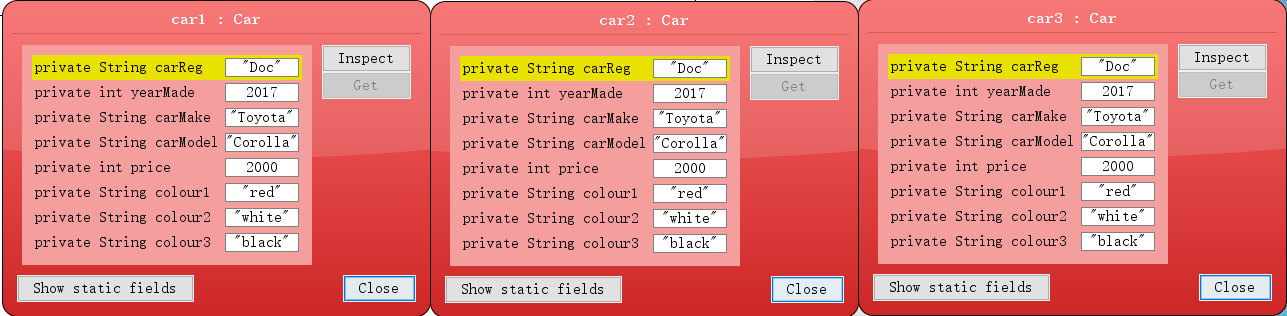
#### Negative test

Test data: “-2000”

Expected result:

|  |  |
| --- | --- |
| car1 | 2017 |
| car2 | 2017 |
| car3 | 2017 |

Actual result:



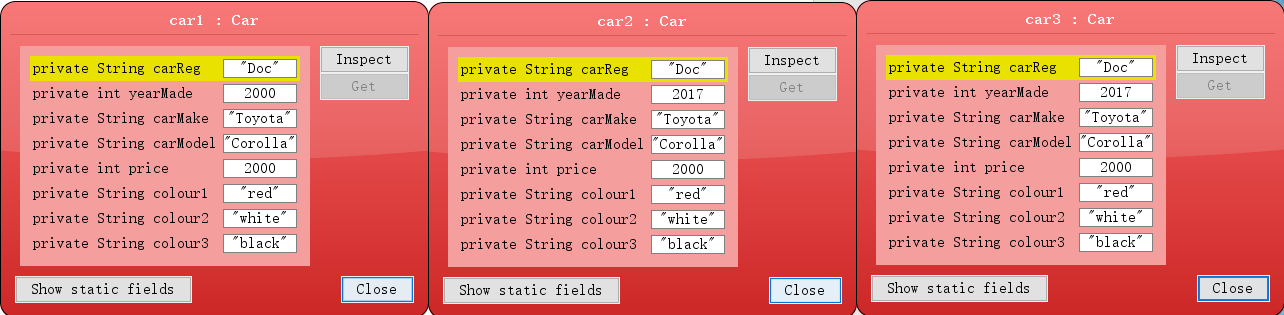
#### Negative test

Test data: “2000”

Expected result:

|  |  |
| --- | --- |
| car1 | 2000 |
| car2 | 2000 |
| car3 | 2000 |

Actual result:



### Test other methods

### ToString()

Test data: null

Expected result:

|  |  |
| --- | --- |
| car1 | “Doc,2000,red,white,black,Toyota,Corolla,2000” |
| car2 | “Doc,2000,red,white,black,Toyota,Corolla,2000” |
| car3 | “Doc,2000,red,white,black,Toyota,Corolla,2000” |

Actual result:

### 

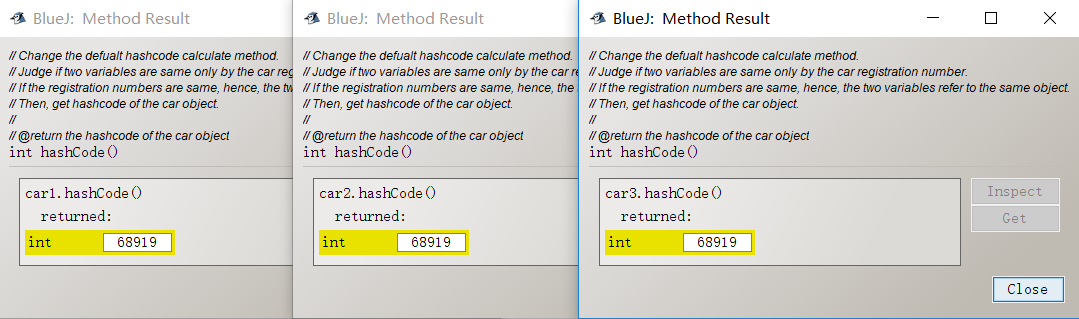
### Hashcode()

Test data: null

Expected result:

car1.hashCode() = car2.hashCode() = car3.hashCode()

Actual result:



### Equals()

Test data: car1 vs car2, car2 vs car3, car3 vs car1

Expected result:

car1= car2, car2 = car3, car3 = car1

Actual result:

