

Project 3

Java Programming
Department of Computer Science
Old Dominion University

Objectives:

The main objective of this assignment is to help you review some of the concepts related to Object Oriented Programming in Java. This assignment also helps you in understanding the test cases written for the methods used by the Java classes using the JUnit testing framework.

General Instructions:

1. Read the Background and Task Description below.
2. In this assignment, you will be provided with the required files to work with for code implementation as well as for testing the code.
3. Complete the methods where implementation of logic is required according to the task given below.
4. Incomplete methods need to be finished wherever you have instructed with a comment “Complete the method”.

Task Description:

Consider you are a car company which manufactures the car and designs new cars. There is one Car agency called A-one which is highly famous all over the world with its most reputed selling cars record all over the world. You want to advertise your new car via A-one agency to increase the new car's publicity. Thus, you want to introduce your new car to the A-one agency and let the agency launch your new car around the world. So, the program will display the cars launched by A-one in particular country with the name of car manufacturer company.

Remember that multiple car companies can introduced their new cars to the A-one car agency since A-one can publish multiple companies' car to the required location in the world.

Tasks:

Complete the program by completing the incomplete methods and related operations into methods. In particular, your program should implement the following tasks.

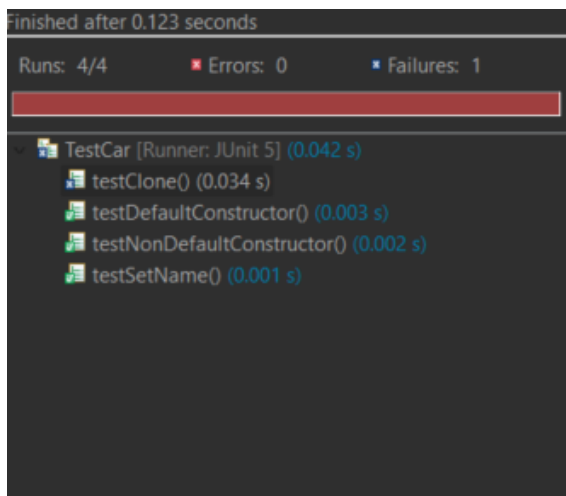
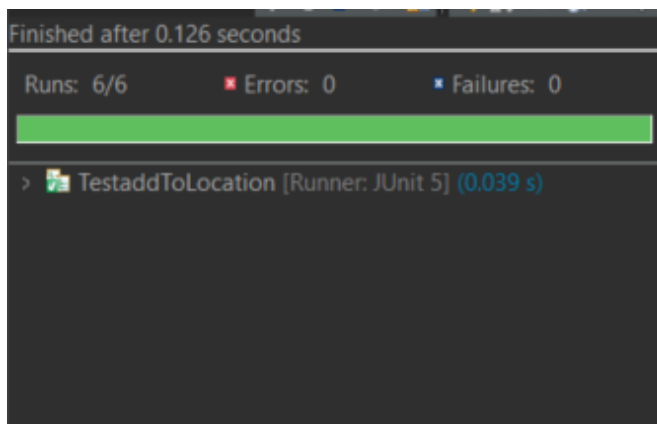
- Create a constructor in Car.java file which will initialize the **name** variable with a specified name which is passed as an argument.
- Create a constructor in addToLocation.java file which will initialize the **location** and **launchLimit** variables with the specified values which are passed as an argument.
- Complete the Launch method in addToLocation.java by fulfilling below requirements:
 - The name of the method given is: public boolean Launch(Car car).
 - Return **false** if the size of **Cars** is equal to **launchLimit** (Here **Cars** is defined as **LinkedHashSet** which contains the set of car companies who wants to launch their cars to A-one car agency).
 - Return **false** if car is already contained in **Cars**
 - If the above two cases are not true then add the car to **Cars** set and return **true**.
- Instantiate an object of type addToLocation in the main method to get the information about cars launched in a particular location. You should pass the arguments **launchLimit** and **location**.
 - **launchLimit**: This is an **int** type field which sets the launch limit to display the launched cars in particular location.
 - **Location**: This is a **String** type field used to display which location you want to launch the car.
- Display all the launched cars along with the location to which the car was launched.
- Implement the testDefaultConstructor() test method in TestCar.java file. (You can call other methods within this method by passing required variables as an argument).
- Implement the testNonDefaultConstructor() method in TestaddToLocation.java file:
 - Test all the methods by passing arguments as a parameters to the required methods.
 - We have skipped the methods **hashCode()** and **equals()** in testDefaultConstructor(). Implement both methods so that we may test over them in this test method.

Sample Output:

```
BMW Launched in USA by A-one Car Agency
Mercedes Launched in USA by A-one Car Agency
Ferrari Launched in USA by A-one Car Agency
Porsche Launched in USA by A-one Car Agency

USA -> 4 of 4 (100.00% full)
-BMW
-Mercedes
-Ferrari
-Porsche
```

Test Output:



Grading

Total points: 100 points

- Compilation (20)
- All required methods implemented (30)
- Unit tests (40)
 - Required test methods implemented (30)
 - Unit tests pass (10)
- Style (comments, indentation, etc...) (10)