Subject: Advance Quality Assurance

WHITE-BOX TESTING WITH JUNIT

Irfan Ullah

ROLL: 20I-2018

Car Rental Management System White-Box Testing with Junit

COVERAGE

STATEMENT COVERAGE:

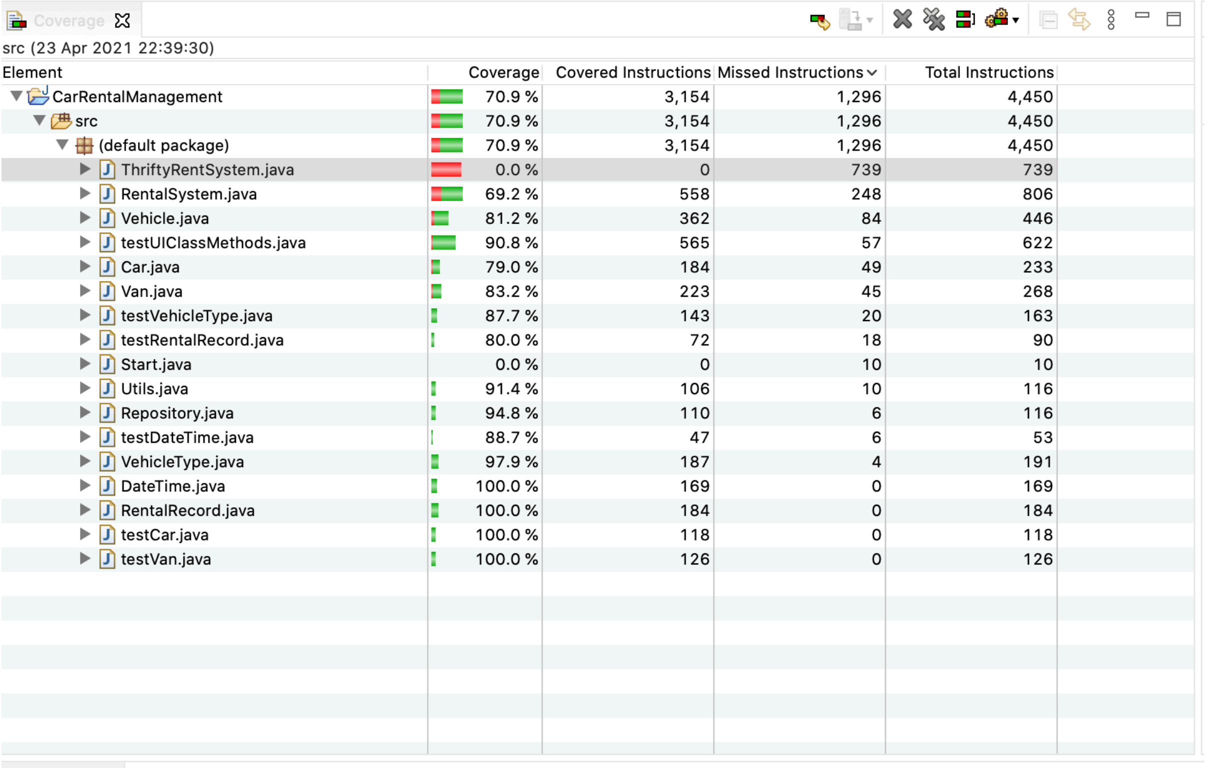
The statement coverage achieved is: **70.9%.** But, after exclusion of the UI class (Thrifty Rental System), the statement coverage becomes: **84%**

The calculation is:
Total Instructions: 4450
Covered Instructions: 3154
Missed Instructions: 1296
UI instructions that are missed: 739

So, after exclusion of the UI instructions from the total instructions:
Total instructions: 4450 – 739 = 3711

Statement Coverage: 3154 / 3711 * 100 = 84 %

The remaining 16% is missed due to exceptions, bugs; which will be listed below.



src (23 Apr 2021 22:39:30)

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
CarRentalManagement	70.9 %	3,154	1,296	4,450
src	70.9 %	3,154	1,296	4,450
(default package)	70.9 %	3,154	1,296	4,450
ThriftyRentSystem.java	0.0 %	0	739	739
RentalSystem.java	69.2 %	558	248	806
Vehicle.java	81.2 %	362	84	446
testUIClassMethods.java	90.8 %	565	57	622
Car.java	79.0 %	184	49	233
Van.java	83.2 %	223	45	268
testVehicleType.java	87.7 %	143	20	163
testRentalRecord.java	80.0 %	72	18	90
Start.java	0.0 %	0	10	10
Utils.java	91.4 %	106	10	116
Repository.java	94.8 %	110	6	116
testDateTime.java	88.7 %	47	6	53
VehicleType.java	97.9 %	187	4	191
DateTime.java	100.0 %	169	0	169
RentalRecord.java	100.0 %	184	0	184
testCar.java	100.0 %	118	0	118
testVan.java	100.0 %	126	0	126

DECISION COVERAGE:

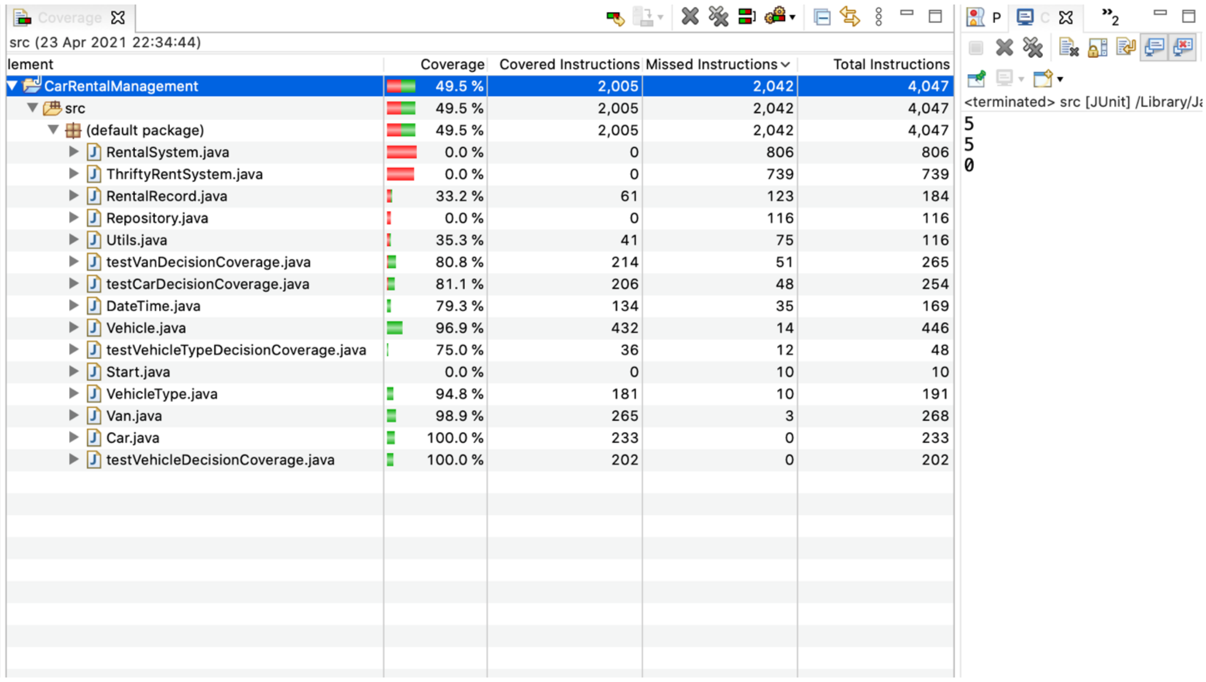
The decision coverage achieved is: **49.5%.** But, after exclusion of the UI class (Thrifty Rental System), the decision coverage becomes: **60%**

The calculation is:
Total Instructions: 4047
Covered Instructions: 2005
Missed Instructions: 2042
UI instructions that are missed: 739

So, after exclusion of the UI instructions from the total instructions:
Total instructions: 4047 – 739 = 3308

Decision Coverage: 2005 / 3308 * 100 = 60 %

The remaining 40% is missed either due to exceptions, bugs; or there were no decisions.



src (23 Apr 2021 22:34:44)

lement	Coverage	Covered Instructions	Missed Instructions	Total Instructions
CarRentalManagement	49.5 %	2,005	2,042	4,047
src	49.5 %	2,005	2,042	4,047
(default package)	49.5 %	2,005	2,042	4,047
RentalSystem.java	0.0 %	0	806	806
ThriftyRentSystem.java	0.0 %	0	739	739
RentalRecord.java	33.2 %	61	123	184
Repository.java	0.0 %	0	116	116
Utils.java	35.3 %	41	75	116
testVanDecisionCoverage.java	80.8 %	214	51	265
testCarDecisionCoverage.java	81.1 %	206	48	254
DateTime.java	79.3 %	134	35	169
Vehicle.java	96.9 %	432	14	446
testVehicleTypeDecisionCoverage.java	75.0 %	36	12	48
Start.java	0.0 %	0	10	10
VehicleType.java	94.8 %	181	10	191
Van.java	98.9 %	265	3	268
Car.java	100.0 %	233	0	233
testVehicleDecisionCoverage.java	100.0 %	202	0	202

```
<terminated> src [JUnit] /Library/Ja
5
5
0
```

STATEMENT COVERAGE AND DECISION COVERAGE COMBINE:

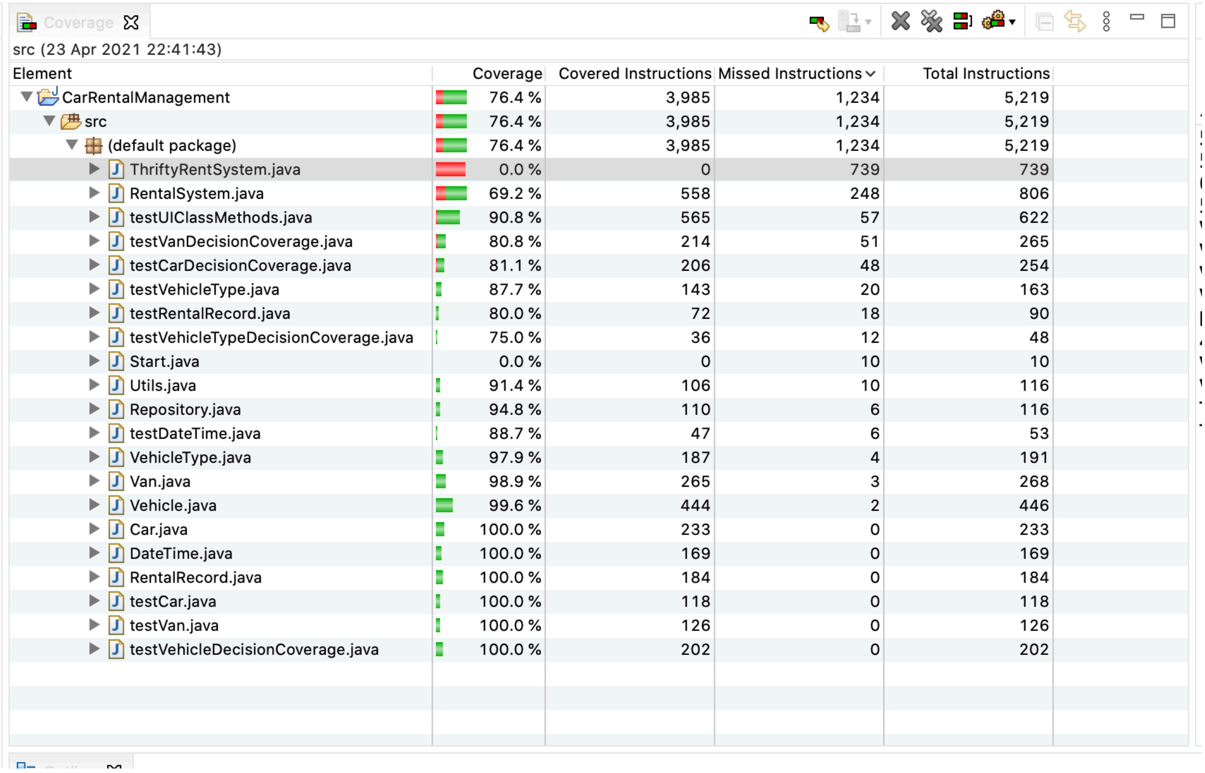
The combine coverage achieved is: **76.4%** But, after exclusion of the UI class (Thrifty Rental System), the coverage becomes: **88%**

The calculation is:
Total Instructions: 5219
Covered Instructions: 3985
Missed Instructions: 1234
UI instructions that are missed: 739

So, after exclusion of the UI instructions from the total instructions:
Total instructions: 5219 – 739 = 4480

Total Coverage: 3985 / 4480 * 100 = 88 %

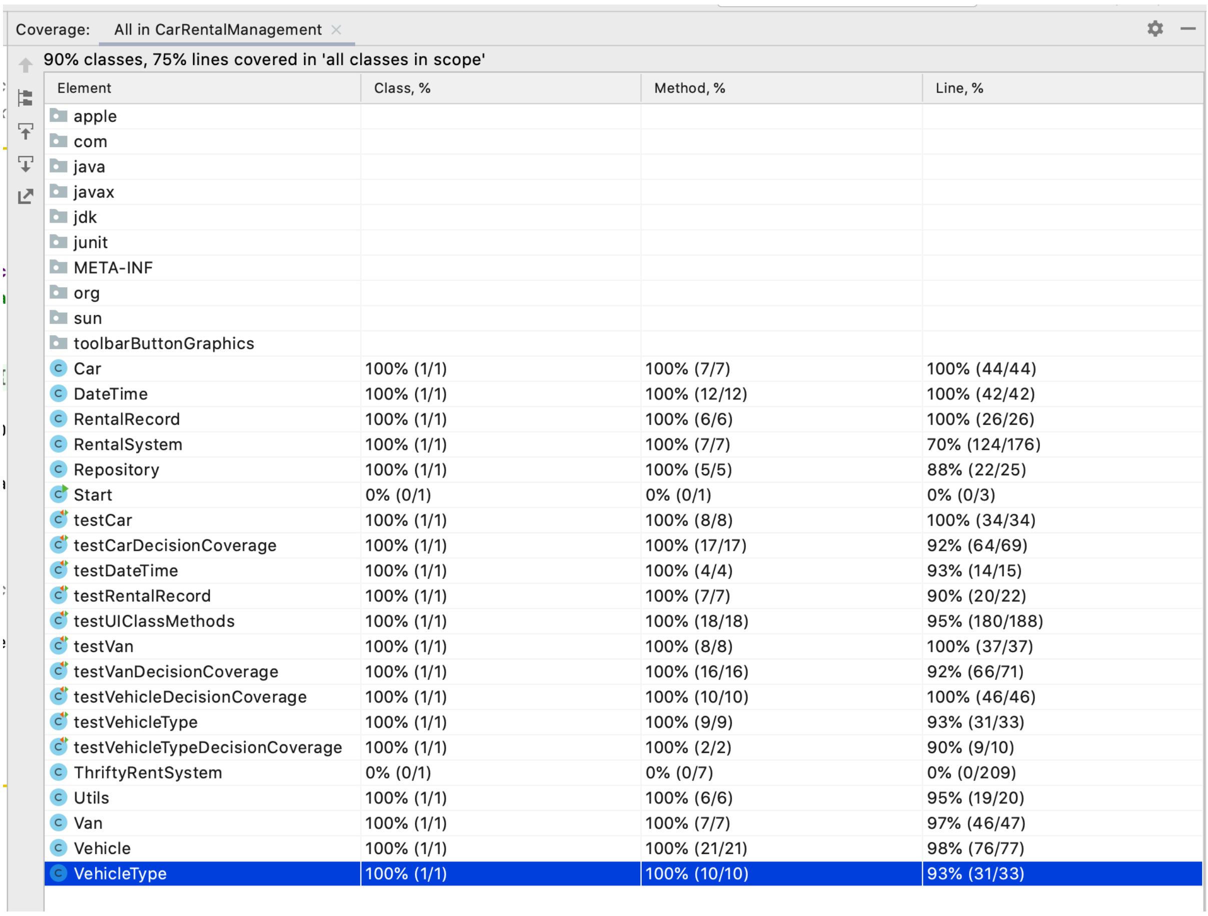
The remaining 12% is missed either due to exceptions, bugs; or unreachable statements.



src (23 Apr 2021 22:41:43)

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
CarRentalManagement	76.4 %	3,985	1,234	5,219
src	76.4 %	3,985	1,234	5,219
(default package)	76.4 %	3,985	1,234	5,219
ThriftyRentSystem.java	0.0 %	0	739	739
RentalSystem.java	69.2 %	558	248	806
testUIClassMethods.java	90.8 %	565	57	622
testVanDecisionCoverage.java	80.8 %	214	51	265
testCarDecisionCoverage.java	81.1 %	206	48	254
testVehicleType.java	87.7 %	143	20	163
testRentalRecord.java	80.0 %	72	18	90
testVehicleTypeDecisionCoverage.java	75.0 %	36	12	48
Start.java	0.0 %	0	10	10
Utils.java	91.4 %	106	10	116
Repository.java	94.8 %	110	6	116
testDateTime.java	88.7 %	47	6	53
VehicleType.java	97.9 %	187	4	191
Van.java	98.9 %	265	3	268
Vehicle.java	99.6 %	444	2	446
Car.java	100.0 %	233	0	233
DateTime.java	100.0 %	169	0	169
RentalRecord.java	100.0 %	184	0	184
testCar.java	100.0 %	118	0	118
testVan.java	100.0 %	126	0	126
testVehicleDecisionCoverage.java	100.0 %	202	0	202

Classes coverage in the total (combine) coverage:



90% classes, 75% lines covered in 'all classes in scope'

Element	Class, %	Method, %	Line, %
apple			
com			
java			
javax			
jdk			
junit			
META-INF			
org			
sun			
toolbarButtonGraphics			
Car	100% (1/1)	100% (7/7)	100% (44/44)
DateTime	100% (1/1)	100% (12/12)	100% (42/42)
RentalRecord	100% (1/1)	100% (6/6)	100% (26/26)
RentalSystem	100% (1/1)	100% (7/7)	70% (124/176)
Repository	100% (1/1)	100% (5/5)	88% (22/25)
Start	0% (0/1)	0% (0/1)	0% (0/3)
testCar	100% (1/1)	100% (8/8)	100% (34/34)
testCarDecisionCoverage	100% (1/1)	100% (17/17)	92% (64/69)
testDateTime	100% (1/1)	100% (4/4)	93% (14/15)
testRentalRecord	100% (1/1)	100% (7/7)	90% (20/22)
testUIClassMethods	100% (1/1)	100% (18/18)	95% (180/188)
testVan	100% (1/1)	100% (8/8)	100% (37/37)
testVanDecisionCoverage	100% (1/1)	100% (16/16)	92% (66/71)
testVehicleDecisionCoverage	100% (1/1)	100% (10/10)	100% (46/46)
testVehicleType	100% (1/1)	100% (9/9)	93% (31/33)
testVehicleTypeDecisionCoverage	100% (1/1)	100% (2/2)	90% (9/10)
ThriftyRentSystem	0% (0/1)	0% (0/7)	0% (0/209)
Utils	100% (1/1)	100% (6/6)	95% (19/20)
Van	100% (1/1)	100% (7/7)	97% (46/47)
Vehicle	100% (1/1)	100% (21/21)	98% (76/77)
VehicleType	100% (1/1)	100% (10/10)	93% (31/33)

Test Cases

Total Test Cases: 91
Passed: 75
Failed: 16

Bugs

Test case	Bug Type	coverage	Test case file
testReturnVehicleWhen_VehicleStatus_isInvalid	Invalid Vehicle Status	Decision Coverage	testCarDecisionCoverage
testReturnVehicleWhen_VehicleStatus_isInvalid	Vehicle is returned even invalid it has an invalid status	Decision Coverage	testCarDecisionCoverage
testReturnVehicleWhenVehicleIdIsInvalid	Vehicle is returned even if vehicle Id is invalid	Decision Coverage	testCarDecisionCoverage
testRentWhenVehicleId_isInvalid	Rent with invalid vehicle Id	Decision Coverage	testCarDecisionCoverage
testSetAdvance	Date don't advance	Statement Coverage	testDateTime
testGetEstimatedDate	Set and get estimate dates don't match	Statement Coverage	testRentalRecord
testCompleteMaintenanceCar	Bug type detail provided in the test case	Statement Coverage	testUIClassMethods
testUiGetDetails	Null Pointer exception	Statement Coverage	testUIClassMethods
testRentCarVehicle	Detail provided in the test case	Statement Coverage	testUIClassMethods
testVehicleMaintenanceVan	Detail provided in test case	Statement Coverage	testUIClassMethods
testVehicleMaintenanceWhenBothVehicleTypesAreAdded	Detail provided in testcase	Statement Coverage	testUIClassMethods
testReturnVehicleWhen_VehicleStatus_isInvalid	Vehicle is return, even if it is provide invalid Vehicle Id	Decision Coverage	testVanDecisionCoverage
testRentWhenVehicleId_isInvalid	Detail provided	Decision Coverage	testVanDecisionCoverage
testLastMaintenanceDate	Date set and get	Statement Coverage	testVehicleType

	are not same		
testIsUnderMaintenanceForCar	Null pointer exception for car	Statement Coverage	testVehicleType