Unit Testing Car Park Wednesday 2/9/2020

Instructions

Time Allowed: 2.0 hrs (9.15 – 11.15)

1. Turn on webcam and microphone
2. Share your screen
3. Record your screen (upload link for recording will be available after the exam. Upload the screen recording before 13.00

Upload to moodle a .zip file with you .java files – CarParkSpace.java, CarParkSpaceTest.java, CarPark.java and CarParkTest.java). Make sure that the code you submit compiles. If you break it while adding a test, leave time to revert back to a compiling version. Upload screen recording to separate link.

NB Make sure you upload the correct files.

N.B. Name your test methods as specified in this document, otherwise I may not be able to correct your assessment.

Notes on Marking Scheme

1. **Your production code and your test code must compile.**
2. **You test should be passing (green bar) before it will be marked. No marks for partially completed/failing tests or for partially completed code or code with compilation errors. Green bar doesn't mean it is correct.**
3. **Implement the tests one by one as specified in this document.**
4. **Name the tests as specified in the document**

UML Diagram

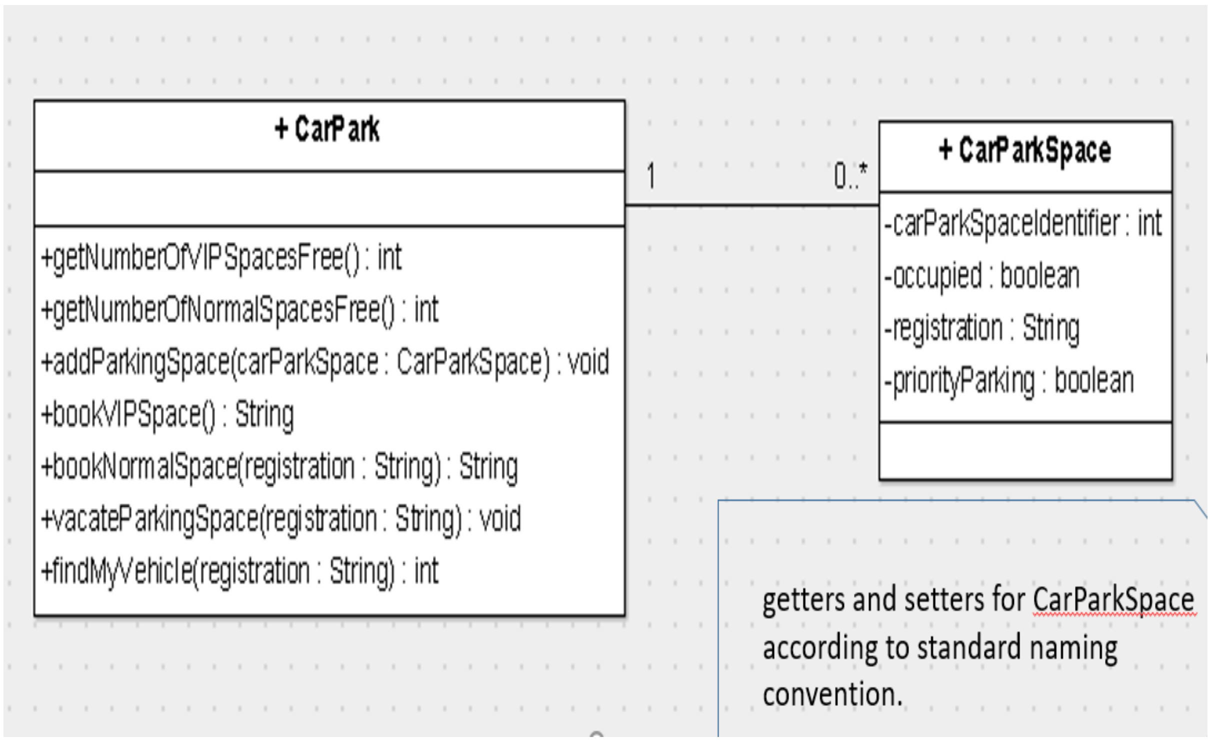
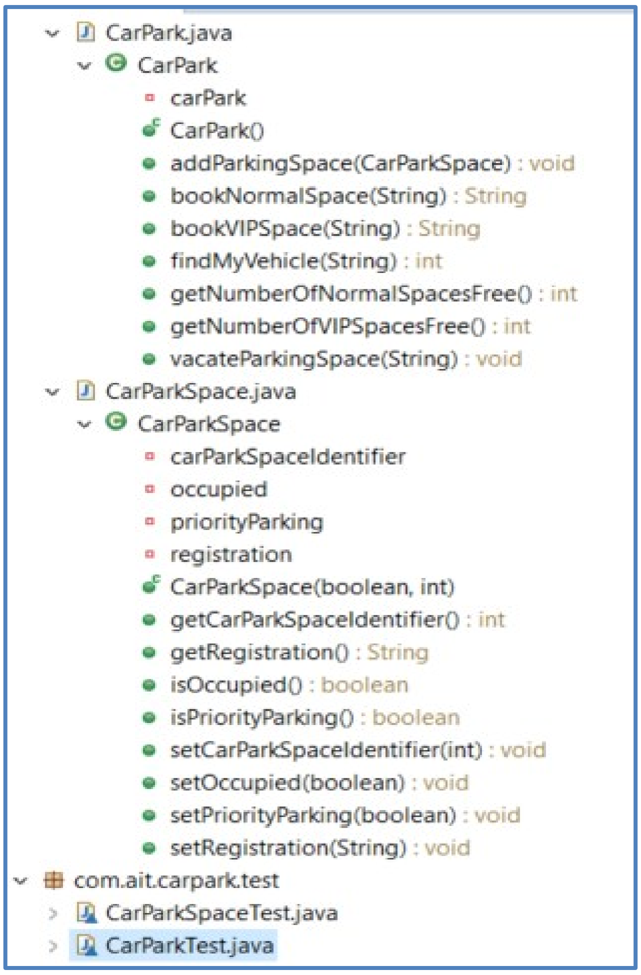

+ CarPark
+getNumberOfVIPSpacesFree() : int
+getNumberOfNormalSpacesFree() : int
+addParkingSpace(carParkSpace : CarParkSpace) : void
+bookVIPSpace() : String
+bookNormalSpace(registration : String) : String
+vacateParkingSpace(registration : String) : void
+findMyVehicle(registration : String) : int
1
0..*
+ CarParkSpace
-carParkSpaceIdentifier : int
-occupied : boolean
-registration : String
-priorityParking : boolean
getters and setters for CarParkSpace according to standard naming convention.

Figure 1



```
  CarPark.java
    CarPark
        carPark
        CarPark()
        addParkingSpace(CarParkSpace) : void
        bookNormalSpace(String) : String
        bookVIPSpace(String) : String
        findMyVehicle(String) : int
        getNumberOfNormalSpacesFree() : int
        getNumberOfVIPSpacesFree() : int
        vacateParkingSpace(String) : void
  CarParkSpace.java
    CarParkSpace
        carParkSpaceIdentifier
        occupied
        priorityParking
        registration
        CarParkSpace(boolean, int)
        getCarParkSpaceIdentifier() : int
        getRegistration() : String
        isOccupied() : boolean
        isPriorityParking() : boolean
        setCarParkSpaceIdentifier(int) : void
        setOccupied(boolean) : void
        setPriorityParking(boolean) : void
        setRegistration(String) : void
com.ait.carpark.test
  CarParkSpaceTest.java
  CarParkTest.java
```

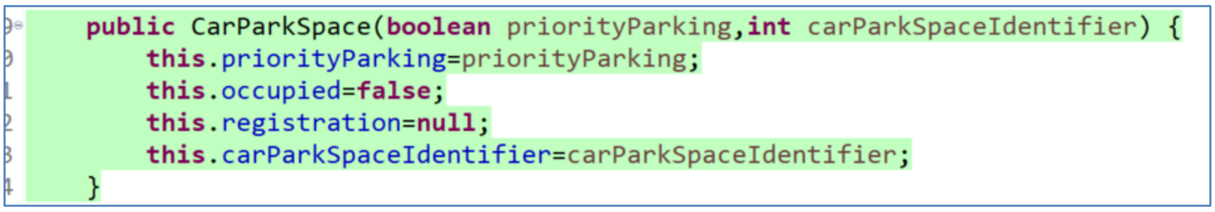
Figure 2

A Car Park is a collection of Parking Spaces. A CarPark contains zero to many CarParkSpaces. Each parking space has attributes to represent

1. If the car parking space is occupied or not (boolean)
2. If the car parking space is a normal space or reserved for use by VIP customers only (boolean)
3. The registration of the vehicle occupying the space (String). This value is only valid if the space is occupied by a vehicle.
4. The identifier of the parking space – an int –assume that this is unique.

CarParkSpace has a constructor that takes 2 arguments (a boolean to indicate if the space is for priority/VIP customers and an int which is the carParkSpaceIdentifier).

There are getter and setter methods for all four instance variables, named according to the standard naming convention. The constructor for CarParkSpace is given here.



```java
    public CarParkSpace(boolean priorityParking,int carParkSpaceIdentifier) {
        this.priorityParking=priorityParking;
        this.occupied=false;
        this.registration=null;
        this.carParkSpaceIdentifier=carParkSpaceIdentifier;
    }
```

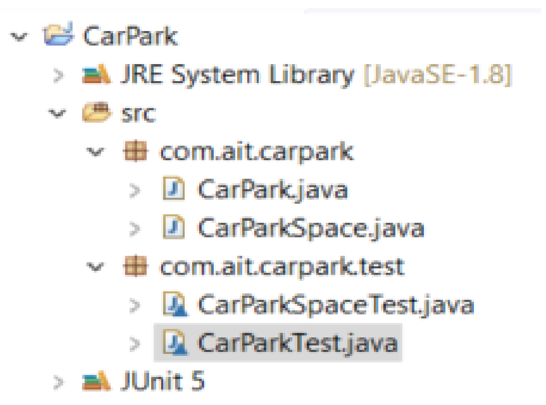
Figure 3

The CarPark System will implement the following User Stories

1. As a car park operator I want to check how many car park spaces are free so that I can display current information on space availability.
2. As a car park operator I want to add a car park space so that I can extend my car park if the business grows.
3. As a driver I want to be able to book a space in the car park so that I can park my car.
4. As a car park operator I want to release a space in the car park when the space is vacated so that another customer can use it.
5. As a cark park operator I want to check if a particular vehicle registration is using a space so that I can assist owners if they cannot remember where the parked their vehicle.

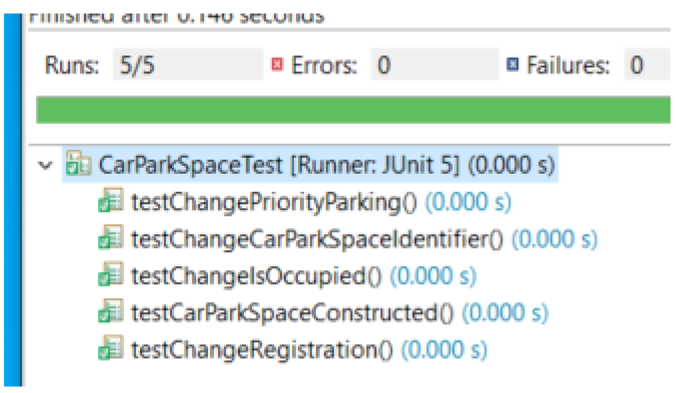
First write a unit test class `CarParkSpaceTest` that tests the constructor and the getters and setters of the `CarParkSpace` class. The tests should be called `testCarParkSpaceConstructed`, `testChangePriorityParking`, `testChangeOccupied`, `testChangeRegistration and` `testChangeCarParkSpaceIdentifier`.

1. Starting the project – Create a separate package for your test code.
2. Make a package called `com.ait.carpark` and `com.ait.carpark.test` in the src folder. Add a class called CarParkSpace to the src and a junit class called `CarParkSpaceTest` as show below. `CarParkSpaceTest` should test the constructor and the getters and setters. Name your tests as shown in Figure 4.



```
CarPark
  JRE System Library [JavaSE-1.8]
  src
    com.ait.carpark
      CarPark.java
      CarParkSpace.java
    com.ait.carpark.test
      CarParkSpaceTest.java
      CarParkTest.java
  JUnit 5
```

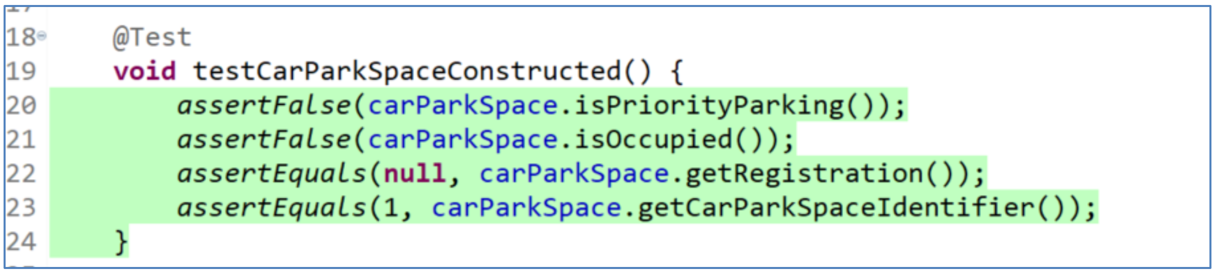
Figure 4



```
Finished after 0.140 seconds
Runs: 5/5    Errors: 0    Failures: 0

CarParkSpaceTest [Runner: JUnit 5] (0.000 s)
    testChangePriorityParking() (0.000 s)
    testChangeCarParkSpaceIdentifier() (0.000 s)
    testChangeIsOccupied() (0.000 s)
    testCarParkSpaceConstructed() (0.000 s)
    testChangeRegistration() (0.000 s)
```

Figure 5



```java
    @Test
    void testCarParkSpaceConstructed() {
        assertFalse(carParkSpace.isPriorityParking());
        assertFalse(carParkSpace.isOccupied());
        assertEquals(null, carParkSpace.getRegistration());
        assertEquals(1, carParkSpace.getCarParkSpaceIdentifier());
    }
```

Figure 6

Now we will test the CarPark functionality, user story by user story.

User Story 1 As a car park operator I want to check how many car park spaces are free so that I can display current information on space availability

Test 1-1 – Call this test `testNoVIPSpacesAvailable`. You must instantiate a CarPark object instance. Before any spaces are added to the CarPark, there should be no available VIP car park spaces in the Car Park. Test that the method `getNumOfVIPSpaces` returns 0 before any spaces are added to the CarPark.

Test 1-2 – Call this test `testNoNormalSpacesAvailable`. Before any spaces are added to the CarPark, there should be no available normal (non VIP) car park spaces in the Car Park. Test that `getNumOfNormalSpaces` returns zero.

User Story 2 – As a car park operator I want to add a car park space so that I can extend my car park if the business grows.

Test 2-1 –Call this test `testOneVIPAndOneNormalSpaceAvailable`. Add a VIP Space and a Normal Space to the CarPark by using method `addParkingSpace` passing in a CarParkSpace object to add CarParkSpace. Check that correct values are returned from methods `getNumberOfVIPSpacesFree` and `getNumberOFNormalSpacesFree` – 1 in each case.

User Story 3 As a driver I want to be able to book a space in the car park so that I can park my car.

Test 3-1 Call this test `testBookVIPSpaceOK`.

The CarPark should contain one normal space and one VIP Space. Test that `bookVIPSpace` method returns String “OK”. Test that there are zero free VIP spaces now.

Test 3-2 Call this test `testBookNormalSpaceOK`.

The CarPark should contain one normal space and one VIP Space. Test that `bookNormalSpace` method returns String “OK”. Test that there are zero free normal spaces now.

Test 3-3 Call this test `testBookNormalSpaceNotAvailable`.

The CarPark should contain one normal space and one VIP Space. Book a normal space using method `bookNormalSpace` (returns “OK”). And now try to book another normal space. `bookNormalSpace` should return the string “NOT AVAILABLE”.

Test 3-3 Call this test `testBookVIPSpaceNotAvailable`.

The CarPark should be contain one normal space and one VIP Space. Book a VIP space using method `bookVIPSpace` (returns “OK”). And now try to book another VIP space. `bookVIPSpace` should return the string “NOT AVAILABLE”.

User Story 4 As a car park operator I want to release a space as unoccupied in the car park when the space is vacated so that another customer can use it.

Test 4-1 Call this test `testReleaseParkingSpace`

The CarPark should contain one normal space. Book the space using method `bookNormalSpace`. Test that there are now zero normal spaces free. Call the method `vacateParkingSpace` with the same registration that was passed in `bookNormalSpace`. Check that the number of normal spaces free (`getNumberOfNormalSpacesFree`) is now one.

Test 4-2 Call this test `testReleaseParkingSpaceInvalidReg`

The CarPark should contain one normal space. Book the space using method `bookNormalSpace`. Test that there are now zero normal spaces free. Call the method `vacateParkingSpace` with a different (invalid) registration to the one that was passed in `bookNormalSpace`. Check that the number of normal spaces free (`getNumberOfNormalSpacesFree`) is still zero.

User Story 5 As a cark park operator I want to check if a particular vehicle registration is using a space so that I can assist owners if they cannot remember where the parked their vehicle.

Test 5-1 Call this test `testFindVehicleOK`.

The Car Park should have one normal parking space, which is booked using `bookNormalSpace`. Call the method `findMyVehicle` with the same registration used in `bookNormalSpace`. Check that the correct CarParkSpaceIdentifier is returned.

Test 5-2 Call this test `testFindVehicleNotFound`

The Car Park should have one normal parking space, which is booked using `bookNormalSpace`. Call the method `findMyVehicle` with a different registration that the one used in `bookNormalSpace`. Check that the correct the value 0 is returned from `findMyVehicle`. Note: Assume 0 is not a valid space identifier and 0 is returned if the vehicle registration is not found in a car park space.