Junit test screenshots
actual tests are in the test folder

```java
public class FNCDsimTest {
    @Test
    public void checkLoggerAndTracker() {
        //testing that a broker, logger and tracker are made when FNCD sim object is created
        FNCDsim newSim = new FNCDsim();
        assertNotEquals( message: "Verify a broker object is made",  unexpected: null, FNCDsim.broker);
        assertNotEquals( message: "Verify tracker singleton created",  unexpected: null, Tracker.getTracker());
        assertNotEquals( message: "Verify logger singleton created",  unexpected: null, Logger.getLogger());
    }
    @Test
    public void checkMakingStaffAndVehicles (){
        FNCDsim newSim = new FNCDsim();
        OpenShop.openShop();
        //check number of cars should be the set number of cars for each type
        assertEquals( message: "Verify num pickup objects", OpenShop.setCarNum, Vehicle.getVehiclesByType(FNCDsim.inventory(), Enums.VehicleType.Pickup).size());
        assertEquals( message: "Verify num car objects", OpenShop.setCarNum, Vehicle.getVehiclesByType(FNCDsim.inventory(), Enums.VehicleType.Car).size());
        assertEquals( message: "Verify num perfCar objects", OpenShop.setCarNum, Vehicle.getVehiclesByType(FNCDsim.inventory(), Enums.VehicleType.PerfCar).size());
        assertEquals( message: "Verify num Motorcycle objects", OpenShop.setCarNum, Vehicle.getVehiclesByType(FNCDsim.inventory(), Enums.VehicleType.Motorcycle).size());
        assertEquals( message: "Verify num Monster objects", OpenShop.setCarNum, Vehicle.getVehiclesByType(FNCDsim.inventory(), Enums.VehicleType.Monster).size());
        assertEquals( message: "Verify num Sedan objects", OpenShop.setCarNum, Vehicle.getVehiclesByType(FNCDsim.inventory(), Enums.VehicleType.Sedan).size());
        assertEquals( message: "Verify num SUV objects", OpenShop.setCarNum, Vehicle.getVehiclesByType(FNCDsim.inventory(), Enums.VehicleType.SUV).size());
```

FNCDsimTest

Tests passed: 2 of 2 tests – 88 ms

FNCDsimTest (FNCDsim.src)	88 ms
checkMakingStaffAndVehicles	85 ms
checkLoggerAndTracker	3 ms

```
Buying: New Convertible added to inventory: Used Clean Mazda MX-5 Miata for $29176.8 cost.
Buying: New Convertible added to inventory: Used Dirty Rolls-Royce Dawn for $17473.6 cost.
Buying: New Convertible added to inventory: Broken Clean Mercedes-Benz S-Class Cabriolet for $23315.5 cost.
Buying: New Convertible added to inventory: LikeNew Dirty Chevrolet Camaro Convertible0 for $39122.0 cost.

Process finished with exit code 0
```

```java
public class InternTest {

    @Test
    public void promoteIntern() {
        FNCDsim sim = new FNCDsim();
        sim.run( runTime: 1);
        Employee intern = Employee.getStaffByType(FNCDsim.currentStaff(), Enums.StaffType.Intern).get(0);
        assertEquals( message: "Returning intern type", Enums.StaffType.Intern, intern.getType());
        //promotion check
        Intern intern1=(Intern) intern;
        assertEquals( message: "Promte to sales check ", Enums.StaffType.Salesperson, intern1.promoteIntern(Enums.StaffType.Salesperson).getType());
        assertEquals( message: "Promte to sales check ", Enums.StaffType.Mechanic, intern1.promoteIntern(Enums.StaffType.Mechanic).getType());

    }
}
```

Run: InternTest

Tests passed: 1 of 1 test – 132 ms

InternTest (FNCDsim.src) 132 ms

promoteIntern 132 ms

```
Fixing: Mechanic Enzo U. fixed the Bandit and made it like LikeNew(earned a $300.0 bonus)
Fixing: Car cleanliness for the Bandit was downgraded to Dirty
////////////////////////////////
Fixing cars at FNCD_South...
////////////////////////////////
Fixing: Mechanic Teagan T. fixed the Atlas2 and made it like LikeNew(earned a $450.0 bonus)
Fixing: Car cleanliness for the Atlas2 was downgraded to Dirty
Fixing: Mechanic Luka T. fixed the Commando0 and made it like Used(earned a $300.0 bonus)
Fixing: Mechanic Kienan I. fixed the Elantra and made it like Used(earned a $450.0 bonus)
```

Customer.java | DailyActivity.java | Employee.java | FNCDsimTest.java | EmployeeTest.java | SalespersonTest.java

```java
package FNCDsim.src;

import ...

public class EmployeeTest {

    @Test
    public void testNumGetStaffByType() {
        ArrayList<Employee> interns = new ArrayList<>();
        interns.add(new Intern());
        interns.add(new Intern());
        interns.add(new Intern());
        interns.add(new Mechanic());
        interns.add(new Salesperson());
        assertEquals( message: "Number of interns should be 3",  expected: 3, Employee.getStaffByType(interns, Enums.StaffType.Intern).size());
    }
}
```

Run: EmployeeTest

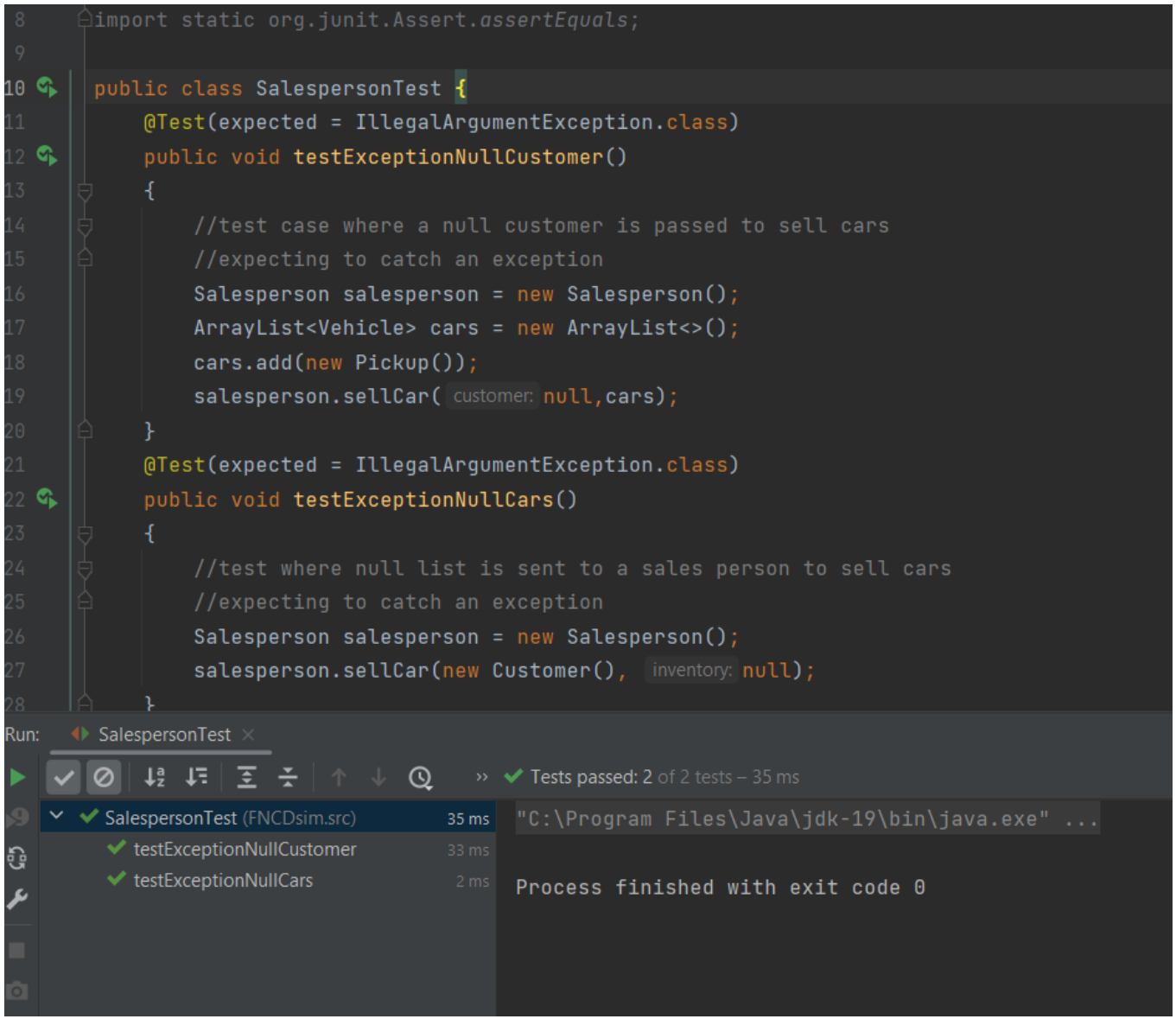
Tests passed: 1 of 1 test – 31 ms

EmployeeTest (FNCDsim.src) 31 ms

testNumGetStaffByType 31 ms

```
"C:\Program Files\Java\jdk-19\bin\java.exe" ...

Process finished with exit code 0
```


import static org.junit.Assert.assertEquals;
public class SalespersonTest {
@Test(expected = IllegalArgumentException.class)
public void testExceptionNullCustomer()
{
//test case where a null customer is passed to sell cars
//expecting to catch an exception
Salesperson salesperson = new Salesperson();
ArrayList<Vehicle> cars = new ArrayList<>();
cars.add(new Pickup());
salesperson.sellCar(customer: null,cars);
}
@Test(expected = IllegalArgumentException.class)
public void testExceptionNullCars()
{
//test where null list is sent to a sales person to sell cars
//expecting to catch an exception
Salesperson salesperson = new Salesperson();
salesperson.sellCar(new Customer(), inventory: null);
}
Run:
SalespersonTest
Tests passed: 2 of 2 tests – 35 ms
SalespersonTest (FNCDsim.src) 35 ms
testExceptionNullCustomer 33 ms
testExceptionNullCars 2 ms
"C:\Program Files\Java\jdk-19\bin\java.exe" ...
Process finished with exit code 0