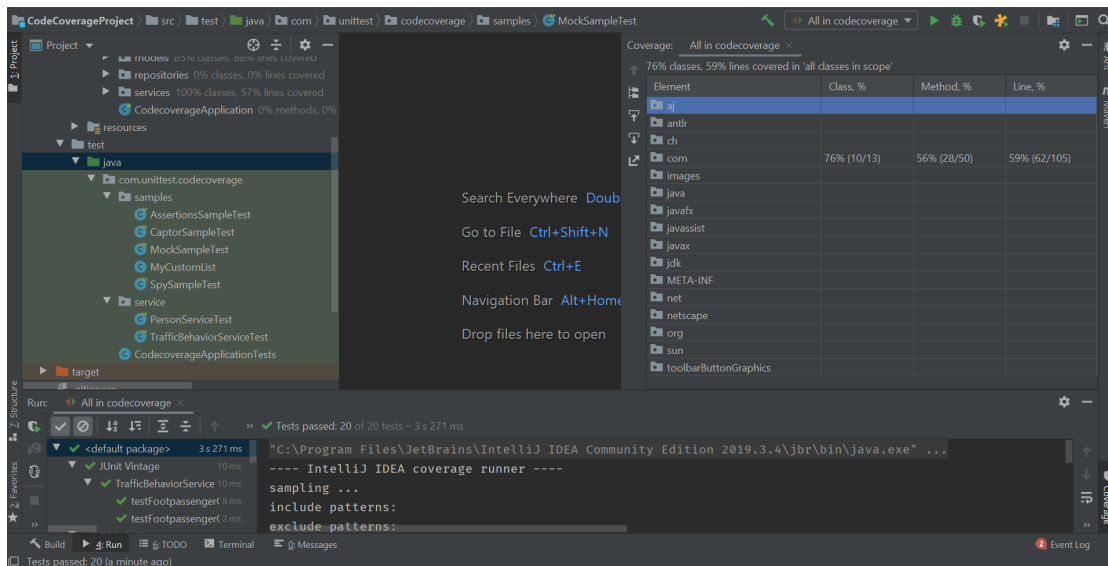


گزارش کار آزمایش ششم

کیمیا نوربخش 97107256

مبینا پورنعمت 97105833

در ابتدا از کد اولیه coverage میگیریم و نتیجه را مشاهده می کنیم:



Coverage: All in codecoverage

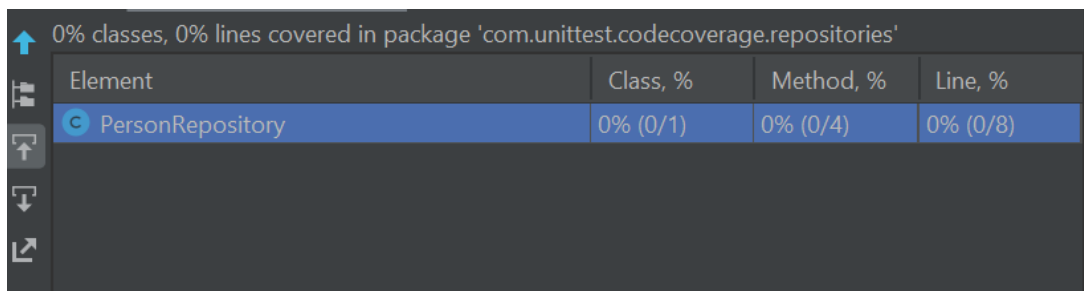
76% classes, 59% lines covered in package 'com.unittest.codecoverage'

Element	Class, %	Method, %	Line, %
exceptions	100% (2/2)	100% (3/3)	66% (6/9)
models	85% (6/7)	61% (22/36)	68% (41/60)
repositories	0% (0/1)	0% (0/4)	0% (0/8)
services	100% (2/2)	50% (3/6)	57% (15/26)
CodecoverageApplication	0% (0/1)	0% (0/1)	0% (0/2)

در ادامه برای بهبود دادن coverage برای کلاس های با coverage پایین تست هایی را اضافه می کنیم. در ادامه coverage پیش از آپدیت کردن تست ها را مشاهده می کنیم و سپس تست هایی را اضافه می کنیم و در ادامه coverage را پس از افزودن تست های جدید بررسی می کنیم. در ادامه این فرایند را برای تمام کلاس ها انجام می دهیم.

- کلاس `PersonRepository`:

تصویر coverage پیش از بهبود تست ها:



0% classes, 0% lines covered in package 'com.unittest.codecoverage.repositories'

Element	Class, %	Method, %	Line, %
PersonRepository	0% (0/1)	0% (0/4)	0% (0/8)

تست اضافه شده:

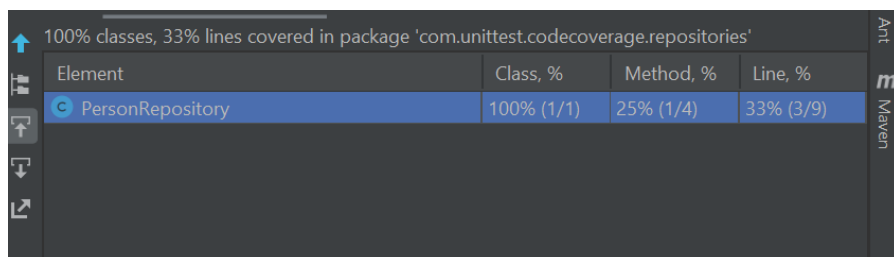
```
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.*;

class PersonRepositoryTest {

    @Test
    void insert() {
        Person person = new Person();
        PersonRepository personRepository = new PersonRepository();
        assertEquals(personRepository.insert(person), person);
    }
}
```

مشاهده مقدار coverage پس از اضافه کردن تست جدید:



100% classes, 33% lines covered in package 'com.unittest.codecoverage.repositories'

Element	Class, %	Method, %	Line, %
PersonRepository	100% (1/1)	25% (1/4)	33% (3/9)

در پکیج models، تعدادی کلاس وجود دارد که coverage هر مورد را در تصویر زیر مشاهده می کنید. در ادامه برای هر کلاس تست هایی اضافه می کنیم تا coverage افزایش پیدا کند.

85% classes, 68% lines covered in package 'com.unittest.codecoverage.models'

Element	Class, %	Method, %	Line, %
validators	100% (1/1)	75% (3/4)	91% (11/12)
Footpassenger	100% (1/1)	80% (8/10)	81% (13/16)
Gender	100% (1/1)	100% (2/2)	100% (2/2)
Person	100% (1/1)	83% (5/6)	90% (9/10)
StreetDirectionFlow	0% (0/1)	0% (0/2)	0% (0/2)
Traffic	100% (1/1)	20% (2/10)	25% (4/16)
TrafficLigth	100% (1/1)	100% (2/2)	100% (2/2)

- کلاس Footpassenger:
تست اضافه شده:

```
@Test
public void testCrossedStreet(){
    Footpassenger footpassenger = new Footpassenger();
    footpassenger.setCrossedTheCrosswalk(true);
    assertEquals( expected: true, footpassenger.crossedTheCrosswalk());
}
```

حالا مجددا coverage این کلاس را بررسی می کنیم:

Footpassenger	100% (1/1)	100% (10/10)	100% (16/16)
---------------	------------	--------------	--------------

- کلاس Person:
تست اضافه شده (به تست قبلی assertion اضافه شده است):

```

@Test
public void testInsert_shouldInsertPersonWithSuccessWhenAllPersonsInfoIsFilled() {
    Person person = new Person();
    person.setName("Name");
    person.setAge(21);
    person.setGender(Gender.M);
    assertEquals(person.getAge(), actual: 21);
    when(repository.insert(any(Person.class))).thenReturn(person);

    service.insert(person);
}

```

حالا مجددا coverage این کلاس را بررسی می کنیم:

Person	100% (1/1)	100% (6/6)	100% (10/10)
--------	------------	------------	--------------

- کلاس StreetDirectionFlow و Traffic:

تست اضافه شده:

```

@Test
public void streetDirectionTest(){
    Traffic traffic = new Traffic();
    traffic.setStreetDirectionFlow(StreetDirectionFlow.ONE_WAY);
    if(traffic.getStreetDirectionFlow() == StreetDirectionFlow.ONE_WAY){
        traffic.setMaxSpeedAllowed((short) 100);
    }
    else {
        traffic.setMaxSpeedAllowed((short) 70);
    }
    assertEquals(traffic.getMaxSpeedAllowed(), (short) 100);
}

```

حالا مجددا coverage این دو کلاس را بررسی می کنیم:

StreetDirectionFlow	100% (1/1)	100% (2/2)	100% (2/2)
Traffic	100% (1/1)	60% (6/10)	62% (10/16)

- کلاس PersonValidator و PersonServiceImpl :
مقدار coverage را پیش از اضافه کردن تست های جدید بررسی میکنیم:

Element	Class, %	Method, %	Line, %
PersonValidator	100% (1/1)	75% (3/4)	91% (11/12)

Element	Class, %	Method, %	Line, %
PersonServiceImpl	100% (1/1)	40% (2/5)	33% (5/15)

تست اضافه شده:

```
@Test
public void testRequiredNameString(){
    Person person = new Person();
    person.setName("KimiaMobina");
    person.setGender(Gender.F);
    service.update(person);
    Person person1 = service.get("KimiaMobina");
    assertEquals(person1, actual: null);
}
```

حالا مجددا coverage این دو کلاس را بررسی می کنیم:

Element	Class, %	Method, %	Line, %
PersonValidator	100% (1/1)	100% (4/4)	100% (12/...

Element	Class, %	Method, %	Line, %
PersonServiceImpl	100% (1/1)	80% (4/5)	66% (10/15)