

# آزمهندسی نرم افزار

## گزارش آزمایش ششم

محسن کثیری ۹۶۱۰۶۰۲۲  
شایان چشم جهان ۹۷۱۱۰۰۴۷

سوال ۱.

ابتدا با انجام run all tests with coverage و گرفتن خروجی html اعداد پوشش آزمون را به دست می آوریم. در تصویر زیر خروجی قابل مشاهده است.

### Overall Coverage Summary

Package	Class, %	Method, %	Line, %
all classes	76.9% (10/13)	57.4% (31/54)	60.2% (50/83)

### Coverage Breakdown

Package	Class, %	Method, %	Line, %
com.unittest.codecoverage	0% (0/1)	0% (0/2)	0% (0/2)
com.unittest.codecoverage.exceptions	100% (2/2)	100% (3/3)	66.7% (4/6)
com.unittest.codecoverage.models	83.3% (5/6)	62.5% (20/32)	62.9% (22/35)
com.unittest.codecoverage.models.validators	100% (1/1)	80% (4/5)	90.9% (10/11)
com.unittest.codecoverage.repositories	0% (0/1)	0% (0/5)	0% (0/7)
com.unittest.codecoverage.services.impl	100% (2/2)	57.1% (4/7)	63.6% (14/22)

با رفتن به هر کلاس نیز می توانیم خط هایی که توسط تست مورد بررسی قرار گرفته اند را مشابه تصویر زیر مشاهده کنیم.

```

1 package com.unittest.codecoverage.models;
2
3 public class Footpassenger {
4
5     private boolean crossedTheCrosswalk;
6     private TrafficLigth crossedTrafficLigth;
7     private boolean lookedToTheRight;
8     private boolean lookedToTheLeft;
9     private boolean crossedTheRoad;
10
11     public boolean crossedTheCrosswalk() {
12         return crossedTheCrosswalk;
13     }
14     public void setCrossedTheCrosswalk(boolean crossedTheCrosswalk) {
15         this.crossedTheCrosswalk = crossedTheCrosswalk;
16     }
17     public TrafficLigth getCrossedTrafficLigth() {
18         return crossedTrafficLigth;
19     }
20     public void setCrossedTrafficLigth(TrafficLigth crossedSignaling) {
21         this.crossedTrafficLigth = crossedSignaling;
22     }
23     public boolean lookedToTheRight() {
24         return lookedToTheRight;
25     }
26     public void setLookedToTheRight(boolean lookedToTheRight) {
27         this.lookedToTheRight = lookedToTheRight;
28     }
29     public boolean lookedToTheLeft() {
30         return lookedToTheLeft;
31     }
32     public void setLookedToTheLeft(boolean lookedToTheLeft) {
33         this.lookedToTheLeft = lookedToTheLeft;
34     }
35     public boolean crossedTheRoad() {
36         return crossedTheRoad;
37     }
38     public void setCrossedTheRoad(boolean crossedTheStreet) {
39         this.crossedTheRoad = crossedTheStreet;
40     }
41
42 }
43

```

حال تلاش می کنیم اعداد پوشش آزمون را برای تمام کلاس ها افزایش دهیم.

با مشاهده کلاس `personRepository` می بینیم که خطوط این کلاس پوشش داده نشده اند.

```

1 package com.unittest.codecoverage.repositories;
2
3 import java.util.Objects;
4
5 import com.unittest.codecoverage.models.Person;
6
7 public class PersonRepository {
8
9     public Person insert(Person person) {
10         Objects.requireNonNull(person, "person can't be null");
11         return person;
12     }
13
14     public void update(Person person) {
15         Objects.requireNonNull(person, "person can't be null");
16     }
17
18     public void delete(String name) {
19         Objects.requireNonNull(name, "name can't be null");
20     }
21
22     public Person get(String name) {
23         Objects.requireNonNull(name, "name can't be null");
24         return null;
25     }
26 }
27

```

با اضافه کردن کد زیر کاری می‌کنیم تا این کلاس نیز تست شود.

```

@Test
public void testPersonRepository(){
    PersonRepository personRepository = new PersonRepository();
    Person person= new Person();
    person.setName("Tom");
    person.setAge(20);
    person.setGender(Gender.M);
    assertEquals(person.getAge(), actual: 20);
    personRepository.insert(person);
    personRepository.delete(person.getName());
}

```

همینطور مشاهده می‌کنیم `streetDirectionFlow` و چند مورد دیگر (مانند متد `setMaxSpeedAllowed`) می‌بینیم که اصلاً استفاده نشده‌اند. در سناریو زیر آن‌ها را هم پوشش می‌دهیم.

```

@Test
public void testStreetSituation() {
    Traffic currentTraffic = new Traffic();
    currentTraffic.setStreetDirectionFlow(StreetDirectionFlow.ONE_WAY);
    Footpassenger currentFootpassengerBehavior = new Footpassenger();
    currentFootpassengerBehavior.setCrossedTheCrosswalk(true);
    currentTraffic.setMaxSpeedAllowed((short) 50);
    currentTraffic.setMinSpeedAllowed((short) 20);
    assertTrue( condition: currentTraffic.getMaxSpeedAllowed() < (short) 60);
}

```

با کدهای زیر `PersonServiceImpl` و `PersonException` را نیز پوشش می‌دهیم.

```
@Test
public void testPersonException(){
    PersonException personException = new PersonException("exception");
}

@Test
public void testPersonServiceImpl(){
    PersonServiceImpl personService = new PersonServiceImpl();
    Person person = new Person();
    person.setGender(Gender.M);
    person.setName("Tom");
    person.setAge(20);
    personService.insert(person);
    personService.get(person.getName());
    personService.delete(person.getName());
}
```

حال در تصاویر زیر اعداد اولیه پوشش آزمون را مشاهده می‌کنیم تا با اعداد جدید مقایسه کنیم.

Class	Class, %	Method, %	Line, %
Footpassenger	100% (1/1)	81.8% (9/11)	81.8% (9/11)
Gender	100% (1/1)	100% (1/1)	100% (2/2)
Person	100% (1/1)	85.7% (6/7)	85.7% (6/7)
StreetDirectionFlow	0% (0/1)	0% (0/1)	0% (0/2)
Traffic	100% (1/1)	27.3% (3/11)	27.3% (3/11)
TrafficLigth	100% (1/1)	100% (1/1)	100% (2/2)

Class	Class, %	Method, %	Line, %
PersonValidator	100% (1/1)	80% (4/5)	90.9% (10/11)

Class	Class, %	Method, %	Line, %
PersonRepository	0% (0/1)	0% (0/5)	0% (0/7)




Class	Class, %	Method, %	Line, %
PersonServiceImpl	100% (1/1)	40% (2/5)	33.3% (4/12)
TrafficBehaviorServiceImpl	100% (1/1)	100% (2/2)	100% (10/10)

حال در تصاویر زیر می‌بینیم که اعداد پوشش آزمون برای تمامی کلاس‌ها افزایش یافته‌اند.

Coverage Breakdown			
Package	Class, %	Method, %	Line, %
com.unittest.codecoverage	0% (0/1)	0% (0/2)	0% (0/2)
com.unittest.codecoverage.exceptions	100% (2/2)	100% (3/3)	100% (6/6)
com.unittest.codecoverage.models	100% (6/6)	87.5% (28/32)	88.6% (31/35)
com.unittest.codecoverage.models.validators	100% (1/1)	100% (5/5)	100% (11/11)
com.unittest.codecoverage.repositories	100% (1/1)	60% (3/5)	57.1% (4/7)
com.unittest.codecoverage.services.impl	100% (2/2)	57.1% (4/7)	63.6% (14/22)

Class	Class, %	Method, %	Line, %
BehaviorException	100% (1/1)	100% (1/1)	100% (1/1)
PersonException	100% (1/1)	100% (2/2)	100% (5/5)

Class	Class, %	Method, %	Line, %
Footpassenger	100% (1/1)	100% (11/11)	100% (11/11)
Gender	100% (1/1)	100% (1/1)	100% (2/2)
Person	100% (1/1)	100% (7/7)	100% (7/7)
StreetDirectionFlow	100% (1/1)	100% (1/1)	100% (2/2)
Traffic	100% (1/1)	63.6% (7/11)	63.6% (7/11)
TrafficLigth	100% (1/1)	100% (1/1)	100% (2/2)

Class 	Class, %	Method, %	Line, %
<a href="#">PersonValidator</a>	100% (1/1)	100% (5/5)	100% (11/11)
Class 	Class, %	Method, %	Line, %
<a href="#">PersonRepository</a>	100% (1/1)	60% (3/5)	57.1% (4/7)
Class 	Class, %	Method, %	Line, %
<a href="#">PersonServiceImpl</a>	100% (1/1)	40% (2/5)	33.3% (4/12)
<a href="#">TrafficBehaviorServiceImpl</a>	100% (1/1)	100% (2/2)	100% (10/10)

تمامی کدها و نتایج در ریپازیتوری گیت هاب موجود است:

[https://github.com/ShayanJahan/SELAB\\_session6](https://github.com/ShayanJahan/SELAB_session6)