

به نام خدا

آزمایشگاه مهندسی نرم افزار

آزمایش سوم

مهسا شیخی - ۹۶۱۰۵۸۸۶

پریشان تیموری - ۹۶۱۰۴۸۸۱

هدف آزمایش

آشنایی با اصول و مفاهیم پوشش آزمون یا Test with coverage با پیاده سازی یک سری تست ساده

شرح آزمایش

ابتدا روی پکیج usermanagement در پروژه ی CodeCoverageProject کلیک راست میکنیم و گزینه ی Run Tests in usermanagement with Coverage را میزنیم. نتیجه اولیه این تست ها بدون تغییر در تصویر زیر و همین طور در فولدر report که از خود اجرای پروژه export شده است دیده میشود:

Coverage: usermanagement in personservice ×			
🚩 ⬆ ⬇ ↗			
Element ▲	Class, %	Method, %	Line, %
> 📁 usermanagement	33% (2/6)	20% (6/30)	10% (8/75)

حالا در قدم بعدی قصد داریم با افزودن تست هایی به برنامه درصد های پوشش آزمون را اضافه کنیم. به این منظور برای کد ها و کلاس های مشخص شده تست های معنادار مینویسم و نتیجه آن را دوباره گزارش میکنیم.

تست های اضافه شده به کلاس PersonTest طبق خواسته آزمایش مشاهده میشود:

```
public class PersonTest {  
  
    mahsaShv *  
    @Test  
    public void test_person_default_constructor() {  
        Person testClass = new Person();  
  
        testClass.setmName("mName1");  
        assertEquals( expected: "mName1", testClass.getmName());  
  
        testClass.setfName("fName1");  
        assertEquals( expected: "fName1", testClass.getfName());  
  
        testClass.setlName("lName1");  
        assertEquals( expected: "lName1", testClass.getlName());  
  
        testClass.setCompanyName("companyName1");  
        assertEquals( expected: "companyName1", testClass.getCompanyName());  
  
        testClass.setPersonId(20);  
        assertEquals( expected: 20, testClass.getPersonId());  
    }  
}
```

Coverage: usermanagement in personservice ×			
Element ▲	Class, %	Method, %	Line, %
> usermanagement	33% (2/6)	33% (10/30)	16% (12/75)

در این عکس افزایش درصدهای اعداد پوشش آزمون را مشاهده میکنیم

به همین ترتیب تعدادی تست به کلاس `userNotFoundExceptionTest` اضافه میکنیم و افزایش اعداد پوشش آزمون را مشاهده میکنیم.

```

mahsaShv *
public class UserNotFoundExceptionTest {
    new *
    @Test
    public void userNotFoundExceptionTest() {
        UserNotFoundException exception = new UserNotFoundException("message", 20);
        assertEquals( expected: 20, exception.getUserId().intValue());
    }
}

```

کلاس و تست اضافه شده

Coverage: usermanagement in personservice ×			
Element ▲	Class, %	Method, %	Line, %
> usermanagement	50% (3/6)	40% (12/30)	20% (15/75)

درصد های پوشش آزمون افزایش یافته

حالا تعدادی تست به کلاس transformServiceTest اضافه میکنیم و افزایش اعداد پوششش آزمون را مشاهده میکنیم.

```
new *
@Test
public void toUserDomainTest() {
    Person person = new Person();
    person.setfName("fName");
    person.setlName("lName");
    person.setCompanyName("company");
    User user = transformService.toUserDomain(person);
    assertEquals( expected: "company", user.getCompanyName());
    assertEquals( expected: "fName", user.getFirstName());
    assertEquals( expected: "lName", user.getLastName());
}
```

```
new *
@Test
public void toUserEntityTest() {
    User user = new User();
    user.setCompanyName("company");
    user.setFirstName("fName");
    user.setLastName("lName");
    Person person = transformService.toUserEntity(user);
    assertEquals( expected: "company", person.getCompanyName());
    assertEquals( expected: "fName", person.getfName());
    assertEquals( expected: "lName", person.getlName());
}
```

تست های اضافه شده به کلاس transformServiceTest

Element ▲			
	Class, %	Method, %	Line, %
> usermanagement	66% (4/6)	73% (22/30)	47% (36/76)

درصد های افزایش یافته پوششش آزمون بعد از اضافه کردن تست ها

```
new *
@Test
public void test_user_setters_getters() {
    User user1 = new User();
    user1.setUserId(1);
    user1.setFirstName("Mahsa");
    user1.setLastName("Sheikhi");
    user1.setCompanyName("myCompany");

    assertEquals(user1.getUserId().intValue(), actual: 1);
    assertEquals(user1.getFirstName(), actual: "Mahsa");
    assertEquals(user1.getCompanyName(), actual: "myCompany");
    assertEquals(user1.getLastName(), actual: "Sheikhi");
}
```

```
new *
@Test
public void test_user_equals() {
    User user1 = new User();
    User user2 = new User();

    user2.setUserId(1);
    user1.setUserId(1);
    assertEquals(user1.equals(user2), actual: true);

    user2 = null;
    assertEquals(user1.equals(user2), actual: false);
}
```

تست های اضافه شده به کلاس userTest

Coverage: usermanagement in personservice ×			
Element ▲	Class, %	Method, %	Line, %
> usermanagement	66% (4/6)	76% (23/30)	57% (44/76)

اعداد افزایش یافته پوشش آزمون

متأسفانه تست هایی که برای کلاس userServiceImpl نوشتم با ارور مواجه میشد و نتوانستم مشکل آن را برطرف کنم زیرا برای درست کردن آن ها باید به سورس برنامه دست میزدیم که برای این آزمایش هدف این نبود.

```

mahsaShv *
public class UserServiceImplTest extends TestCase {
    new *
    @Test
    public void test_save_and_find() {
        User user1 = new User();
        User user2 = new User();
        user1.setUserId(1);
        UserServiceImpl userService = new UserServiceImpl();

        userService.save(user1);
        userService.save(user2);

        assertEquals(userService.findById(1).equals(user1), true);
    }
}

```

تست نوشته شده برای این کلاس

```

UserServiceImplTest.test_save_and_find
Tests failed: 1 of 1 test - 18 ms

java.lang.NullPointerException: Cannot invoke "usermanagement.service.TransformService.toUserEntity(usermanagement.service.User)" because "this.transformer" is null
    at usermanagement.service.UserServiceImpl.save(UserServiceImpl.java:58)
    at usermanagement.service.UserServiceImplTest.test_save_and_find(UserServiceImplTest.java:16) <18 internal lines>

```

اروری که مربوط به سورس اصلی برنامه بود

لینک پروژه

https://github.com/mahsaShv/SE_LAB_HW3.git