

بسم الله الرحمن الرحيم

تمرین سوم درس آزمون نرم افزار

امیرحسین راحتی ۸۱۰۱۰۰۱۴۴

علی ممتحن ۸۱۰۱۰۰۲۱۳

آدرس مخزن

<https://github.com/AmirhosseinRHT/Software-Testing/>

شناسه کامیت

8dd3835e735e6f7303f4b30bcf8c4f68e162b978

## سوال 1

از `@WebMvcTest` برای تست لایه وب یعنی صرفاً کنترلر استفاده می شود. و لازم نیست کل برنامه لود شود. اما در انوتیشن دیگر کل برنامه باید لود شود یعنی دیتابیس و سرویس و دیگر عناصر برنامه باید لود شوند که در واقع برای تست integration تست استفاده می شود و مناسب تر از `WebMvcTest` برای تست یکپارچگی کل برنامه می باشد.

## سوال 2

(الف)

id	A	B	C	Predicate
1	T	T	T	T
2	T	F	T	F
3	T	T	F	F
4	T	F	F	T
5	F	T	T	T
6	F	F	T	F
7	f	T	F	T
8	F	F	F	T

(ب)

اگر a اکتیو باشد: (7,3)

اگر b اکتیو باشد: (1,2) (1,6) (3, 4) (5,2) (5,6)

اگر c اکتیو باشد : (2,4) (2,8) (3,1) (6,4) (6,8)

(پ)

اگر a اکتیو باشد: (7و3)

اگر b اکتیو باشد: (1و2) (3و4) (5و6)

اگر c اکتیو باشد: (1و3) (2و4) (6و8)

خیر زیرا به وضوح تعداد جفت های قسمت ب بیشتر از پ است. جدا آ آن کاملاً برعکس است یعنی این قسمت پ است که زیرمجموعه قسمت ب می باشد.

(ت)

خیر با زوج 1 و 8 clause coverage داریم اما predicate coverage نداریم.

### سوال 3

۳ حالت وجود دارد

(a) حالتی که ورودی ها غیر مجاز هستند و در ابتدا با شرط اول مقدار input invalid برگردانده می شود.

(b) حالتی که ورودی ها مجاز هستند و  $price < minPurchase$  می باشد.

(c) حالتی که ورودی ها مجاز هستند ولی  $price > minPurchase$  است.

با توجه به حالات بالا ورودی ها و assertion آن ها به صورت زیر می شود.

1.  $minPurchase=-50, discountRate=-0.1, price=-10$

`assertEquals("Invalid input", calculateDiscountedPrice(-10, -0.1, -50));`

2.  $minPurchase=0, discountRate=0, price=-10$

`assertEquals("Invalid input", calculateDiscountedPrice(-10, 0, 0));`

3.  $minPurchase=50, discountRate=1.2, price=-10$

`assertEquals("Invalid input", calculateDiscountedPrice(-10, 1.2, 50));`

minPurchase=0, discountRate=-0.1, price=0 .4  
assertEquals("Invalid input", calculateDiscountedPrice(0, -0.1, 0));

minPurchase=50, discountRate=0, price=100 .5  
assertEquals("100", calculateDiscountedPrice(100, 0, 50));

minPurchase=-50, discountRate=1.2, price=0 .6  
assertEquals("Invalid input", calculateDiscountedPrice(0, 1.2, -50));

minPurchase=50, discountRate=-0.1, price=100 .7  
assertEquals("Invalid input", calculateDiscountedPrice(100, -0.1, 50));

minPurchase=50, discountRate=0, price=10 .8  
assertEquals("10", calculateDiscountedPrice(10, 0, 50));

minPurchase=50, discountRate=0.5, price=100 .9  
assertEquals("50.0", calculateDiscountedPrice(100, 0.5, 50));