به نام خدا

تمرین چهارم درس آزمون نرم افزار

آریا وارسته نژاد

9531345

آریا خلیق

9524014

حسین محمدی

9533081

1400

1-الف)

	G	F	C	Pc	Pf	Pg
1	F	F	F			
2	F	F	T	T	T	
3	F	T	F	T		
4	F	T	T		T	T
5	T	F	F		T	T
6	T	F	T	T		
7	T	T	F	T	T	
8	T	T	T			

RACC:

Pc: (5, 6)-(3, 4)

Pg: (7, 3)-(2, 6)

Pf: (7, 5)-(4, 2)

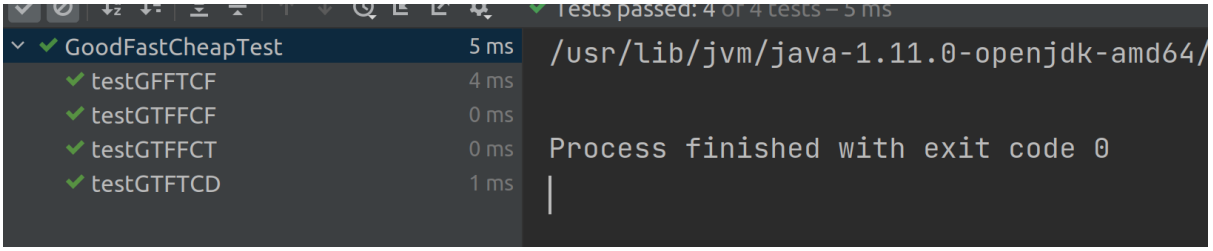
Min: (3, 5, 6, 7)

```
public class GoodFastCheapTest {
```

```
    GoodFastCheap gfc;

    @Before
    public void setUp() {
        gfc = new GoodFastCheap();
    }

    @After
    public void tearDown() {
        gfc = null;
    }

    // Test format: Good(T/F)Fast(T/F)Cheap(T/f)
    @Test
    public void testGFFTCF(){
        gfc.makeFast();
        assertFalse(gfc.isSatisfactory());
    }

    @Test
    public void testGTFFCF() {
        gfc.makeGood();
        assertFalse(gfc.isSatisfactory());
    }

    @Test
    public void testGTFFCT() {
        gfc.makeGood();
        gfc.makeCheap();
        assertTrue(gfc.isSatisfactory());
    }

    @Test
    public void testGTFTCD() {
        gfc.makeGood();
```

```
        gfc.makeFast();
        assertTrue(gfc.isSatisfactory());
    }
}
```

نتایج:

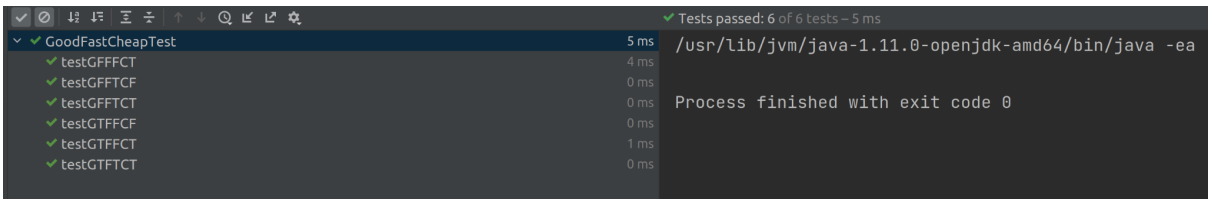


1-ب)

برآورده نمی کند، حالت های 2 و 4 کم هستند و بایستی اضافه شوند.

```
@Test
public void testGFFTCT() {
    gfc.makeFast();
    gfc.makeCheap();
    assertTrue(gfc.isSatisfactory());
}

@Test
public void testGFFFCT() {
    gfc.makeCheap();
    assertFalse(gfc.isSatisfactory());
}
```

نتیجه:



2-الف)

برای صدا زدن تابع remove بر روی هر iterator باید دو شرط زیر حتما برقرار باشد:

1) باید از قبل ()next صدا زده شده باشد.
2) نباید ()remove دو بار تا next بعدی صدا زده شود.

2-ب)

در جاوا ArrayList ها که در قسمت بعد از آن استفاده شده از این واسط را پیاده سازی میکند.

2-ج)

اگر شرط این که اول تابع next را صدا بزنیم a و این که remove صدا زده نشده باشد را b در نظر بگیریم و a^b=T باید ۳ حالت زیر را برای CACC بررسی کنیم:

Pa => b=T

Pb => a=T

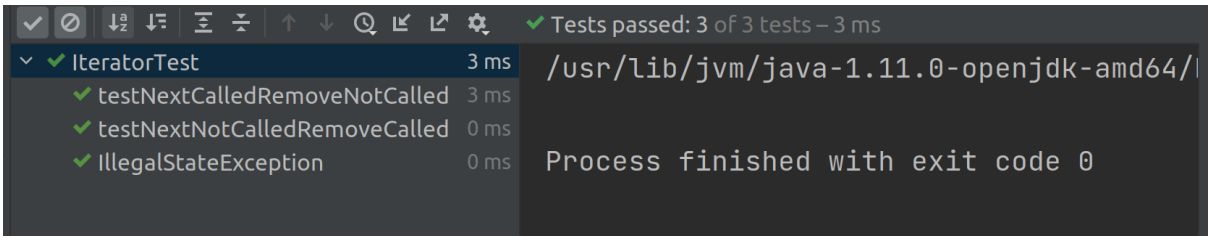
- a=T b=T
- a=T b=F
- a=F b=T

```
public class IteratorTest {
    List<String> list;

    @Before
    public void setUp() {
        list = new ArrayList<String>();
    }

    @After
```

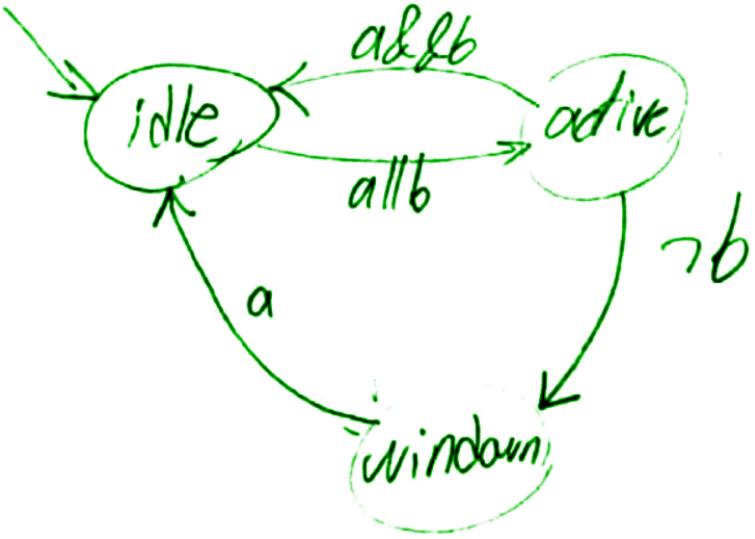
```
    public void tearDown() {
        list = null;
    }


    @Test(expected = IllegalStateException.class)
    public void testNextNotCalledRemoveCalled() {
        list.add("a");

        Iterator<String> iterator = list.iterator();

        iterator.remove();
    }


    @Test(expected = IllegalStateException.class)
    public void IllegalStateException() {
        list.add("a");
        list.add("b");

        Iterator<String> iterator = list.iterator();
        iterator.next();
        iterator.remove();

        iterator.remove();
    }

    @Test
    public void testNextCalledRemoveNotCalled() {
        list.add("a");
        list.add("b");

        Iterator<String> iterator = list.iterator();
        iterator.next();
```

```
        iterator.remove();

        List<String> newList = new
ArrayList<String>();
        newList.add("b");

        assertEquals(list, newList);
    }
}
```

نتیجه:



3-الف)



(ب-3

Wind down:

Not a

Idle:

Not(a or b) = Not a and Not b

Active:

Not(a and b) and Not(Not b) = Not a and Not b

(ج-3

Active to Active: TT, FF, FT

Active to Idle: TT, TF, FT

Active to Wind down: TT, TF