

PMOBO – 2 Laborategiko txostena

Izena: Eneko Sampetro eta Andoni Bermo **Data:** 2015-02-17



Lortutako ezagutza eta gaitasunak

Behin 2. laborategia bukatu dudala, hurrengo atazak ondo egin ditudala pentsatzen dut., eta beraz etorkizunean errepikatu beharko banitu gai izango nintzateke. (markatu BAI ala EZ).

	Eneko Sampetro/Andoni Bermo	BAI	EZ
1	Eclipse abiarazi, lan egin defektuzkoa ez den <i>lanleku (workspace)</i> batetan (nirean hain zuzen ere), eta sortu Javaz proiektuak, paketeak eta klaseak	✓	
2	Identifikatu eta ulertu UML diagrama sinple bat	✓	
3	Ulertu proba kasuen funtzionamendua (<i>TestCase</i> klasea): <i>setUp()</i> eta <i>tearDown()</i> metodoen erabilgarritasuna, baita <i>assertion</i> eta proba kasuena.	✓	
4	Inplementatu proba kasu sinpleak, hau da, badakit metodo bakoitzaren proba (<i>Test</i>) egiteko, proba kasu guztiak probatu behar ditudala eta proba kasu bakoitzeko <i>assert</i> bat egongo dela.	✓	
5	Aurreko laborategian egindako akats guztiak konpontzen badakidanez, laborategi honetarako konpondu ditut (bukaeran zerrendatu zein zen errorea eta nola konpondu duzun).	✓	
6	<i>Junit</i> -a egikaritzean, metodo guztien testak gainditu ditut (denak berdez agertzen dira)	✓	
7	Inplementatu proba talde (<i>TestSuite</i> klase) bat, hainbat proba kasu izateko barne	✓	
8	Arauak betetzen dituzten izenak aukeratu ditut, bai klase, bai proiektu, pakete, atributu, metodo eta aldagaien izenak	✓	
	BESTE BATZUK: Azaldu ikasi duzun beste zerbait		

Ezezko erantzunen bat eman baduzu mesedez jarraian azaldu zergatia, eta ze nolako neurria hartuko duzun sortu zaizun arazoa konpontzeko.

--

1. Laborategiko akatsen zerrenda eta konponbidea

Azkenik, 1. atazaren soluzioa ipini aurretik, erantzun ondorengo galderak:

1	Inplementaturiko JUnit guztiak ongi funtzionatu al dute?	✓
2	Ezezko kasuan, aipatu ezazu zergaitia.	



1 Atazaren soluzioa: Pertsona klasearen proba kasuak

Idatzi *Pertsona* klaseko hurrengo metodoen inplementazioa *testPertsona()*, *testIdBerdina()*, *testGidatuDezake()*, *testIzenarenIniziala()*, *testAbizenarenIniziala()* y *testEmanDiezaiokeOdola()*.

```
private Pertsona p1;
private Pertsona p2;

@Before
public void setUp() throws Exception {
    p1=new Pertsona("Andoni Bermo", 21, 34, 'G', "Euskal Herria",
"A+");
    p2=new Pertsona("Eneko Sampedro", 18, 24, 'G', "Euskal Herria",
"O-");
}

@After
public void tearDown() throws Exception {
    p1=null;
    p2=null;
}
```

testIdBerdinaDu()

```
@Test
public void testIdBerdinaDu() {
    assertFalse(p1.idBerdinaDu(p2.getIdPertsona()));
    assertFalse(p2.idBerdinaDu(p1.getIdPertsona()));
    p1=new Pertsona("Andoni Bermo", 21, 24, 'G', "Euskal Herria",
"A+");
    assertTrue(p1.idBerdinaDu(p2.getIdPertsona()));
    assertTrue(p2.idBerdinaDu(p1.getIdPertsona()));
}
```

testGidatuDezake()

```
@Test
public void testGidatuDezake() {
    assertTrue(p1.gidatuDezake());
    assertTrue(p2.gidatuDezake());
    p1=new Pertsona("Andoni Bermo", 15, 34, 'G', "Euskal Herria",
"A+");
    assertFalse(p1.gidatuDezake());
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"A+");
    assertTrue(p1.gidatuDezake());
    p1=new Pertsona("Andoni Bermo", 17, 34, 'G', "Erresuma Batua",
"A+");
    assertTrue(p1.gidatuDezake());
    p1=new Pertsona("Andoni Bermo", 16, 34, 'G', "Erresuma Batua",
"A+");
    assertFalse(p1.gidatuDezake());
    p1=new Pertsona("Andoni Bermo", 15, 34, 'G', "Australia", "A+");
    assertFalse(p1.gidatuDezake());
    p1=new Pertsona("Andoni Bermo", 16, 34, 'G', "Australia", "A+");
```

```

        assertTrue(p1.gidatuDezake());
        p1=new Pertsona("Andoni Bermo", 16, 34, 'G', "Estatu Batuak",
"A+");
        assertTrue(p1.gidatuDezake());
        p1=new Pertsona("Andoni Bermo", 15, 34, 'G', "Estatu Batuak",
"A+");
        assertFalse(p1.gidatuDezake());
        p1=new Pertsona("Andoni Bermo", 14, 34, 'G', "Etiopia", "A+");
        assertTrue(p1.gidatuDezake());
        p1=new Pertsona("Andoni Bermo", 13, 34, 'G', "Etiopia", "A+");
        assertFalse(p1.gidatuDezake());
    }

```

testOdolaEmanDiezaioke()

```

@Test
public void testOdolaEmandiezaioke() {

    //eginda
    assertTrue(p2.odolaEmandiezaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"AB+");
    assertTrue(p2.odolaEmandiezaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"AB-");
    assertTrue(p2.odolaEmandiezaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"A-");
    assertTrue(p2.odolaEmandiezaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"B+");
    assertTrue(p2.odolaEmandiezaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"B-");
    assertTrue(p2.odolaEmandiezaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"O+");
    assertTrue(p2.odolaEmandiezaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"O-");
    assertTrue(p2.odolaEmandiezaioke(p1));

    p2=new Pertsona("Eneko Sampedro", 18, 24, 'G', "Euskal Herria",
"O+");
    assertFalse(p2.odolaEmandiezaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"AB+");
    assertTrue(p2.odolaEmandiezaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"AB-");
    assertFalse(p2.odolaEmandiezaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"A-");
    assertFalse(p2.odolaEmandiezaioke(p1));
}

```

```

    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"B+");
    assertTrue(p2.odolaEmandieziaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"B-");
    assertFalse(p2.odolaEmandieziaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"O+");
    assertTrue(p2.odolaEmandieziaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"A+");
    assertTrue(p2.odolaEmandieziaioke(p1));

    p2=new Pertsona("Eneko Sampedro", 18, 24, 'G', "Euskal Herria",
"B-");
    assertFalse(p2.odolaEmandieziaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"AB+");
    assertTrue(p2.odolaEmandieziaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"AB-");
    assertTrue(p2.odolaEmandieziaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"A-");
    assertFalse(p2.odolaEmandieziaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"B+");
    assertTrue(p2.odolaEmandieziaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"B-");
    assertTrue(p2.odolaEmandieziaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"O+");
    assertFalse(p2.odolaEmandieziaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"O-");
    assertFalse(p2.odolaEmandieziaioke(p1));

    p2=new Pertsona("Eneko Sampedro", 18, 24, 'G', "Euskal Herria",
"A-");
    assertFalse(p2.odolaEmandieziaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"AB+");
    assertTrue(p2.odolaEmandieziaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"AB-");
    assertTrue(p2.odolaEmandieziaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"A-");
    assertTrue(p2.odolaEmandieziaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"B+");
    assertFalse(p2.odolaEmandieziaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"B-");
    assertFalse(p2.odolaEmandieziaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"O+");
    assertFalse(p2.odolaEmandieziaioke(p1));
    p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"A+");

```

```

        assertTrue(p2.odolaEmandieziaioke(p1));

        p2=new Pertsona("Eneko Sampedro", 18, 24, 'G', "Euskal Herria",
"B+");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"AB+");
        assertTrue(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"AB-");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"A-");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"B+");
        assertTrue(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"B-");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"O+");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"O-");
        assertFalse(p2.odolaEmandieziaioke(p1));

        p2=new Pertsona("Eneko Sampedro", 18, 24, 'G', "Euskal Herria",
"A+");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"AB+");
        assertTrue(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"AB-");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"A-");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"B+");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"B-");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"O+");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"A+");
        assertTrue(p2.odolaEmandieziaioke(p1));

        p2=new Pertsona("Eneko Sampedro", 18, 24, 'G', "Euskal Herria",
"AB-");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"AB+");
        assertTrue(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"AB-");
        assertTrue(p2.odolaEmandieziaioke(p1));

```

```

        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"A-");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"B+");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"B-");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"O+");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"O-");
        assertFalse(p2.odolaEmandieziaioke(p1));

        p2=new Pertsona("Eneko Sampedro", 18, 24, 'G', "Euskal Herria",
"AB+");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"AB+");
        assertTrue(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"AB-");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"A-");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"B+");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"B-");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"O+");
        assertFalse(p2.odolaEmandieziaioke(p1));
        p1=new Pertsona("Andoni Bermo", 18, 34, 'G', "Euskal Herria",
"A+");
        assertFalse(p2.odolaEmandieziaioke(p1));
    }
}

```

testGetIzenarenIniziala()

```

@Test
public void testGetIzenarenIniziala() {
    assertTrue('A'==p1.getIzenarenIniziala());
    assertFalse('B'==p1.getIzenarenIniziala());
    assertTrue('E'==p2.getIzenarenIniziala());
    assertFalse('D'==p2.getIzenarenIniziala());
}

```

testGetAbizenarenIniziala()

```
@Test
public void testGetAbizenarenIniziala() {
    assertTrue('B'==p1.getAbizenarenIniziala());
    assertFalse('J'==p1.getAbizenarenIniziala());
    assertTrue('S'==p2.getAbizenarenIniziala());
    assertFalse('P'==p2.getAbizenarenIniziala());
    p1=new Pertsona("Andoni Bermo Pérez", 18, 34, 'G', "Euskal
Herria", "A+");
    assertTrue('P'==p1.getAbizenarenIniziala());
    assertFalse('J'==p1.getAbizenarenIniziala());
}
```




2 Atazaren soluzioa: Kotxe klasearen proba kasuak

Idatzi *Kotxe* klaseko hurrengo metodoen inplementazioa *testGetJabea()*, eta *testAldatuJabea()*.

testAldatuJabea()

```
@Test
public void testAldatuJabea() {
    k.aldatuJabea(p1);
    assertTrue("Eneko Sampetro"==k.getJabea().getIzenOsoa());
    assertFalse("Eneko Sampe"==k.getJabea().getIzenOsoa());
    k.aldatuJabea(p2);
    assertTrue("Andoni Bermo"==k.getJabea().getIzenOsoa());
    assertFalse("Andoni"==k.getJabea().getIzenOsoa());
}
```

testGetJabea()

```
@Test
public void testGetJabea() {
    k.aldatuJabea(p1);
    assertTrue("Eneko Sampetro"==k.getJabea().getIzenOsoa());
    assertFalse("Eneko Sampe"==k.getJabea().getIzenOsoa());
}
```