

Software Ingeniaritzak II

Lab 2: JUnit + Mockito

A. Goñi, J. Iturrioz

Forum Aplikazioa

Del 17 de septiembre al 6 de octubre

TE DEVOLVEMOS **EL IVA**

CONDICIONES

Promoción válida **SÓLO PARA SOCIOS**. Ser socio es gratis e inmediato.

Necesario tener informado un **teléfono móvil** para recibir el descuento.

Promoción y cupón **no aplicable en outlets**.

El máximo importe del cupón de descuento es de 50€.

Únicamente se recibirá un cupón por socio.

Cupón válido para **un solo uso y no acumulable a otros vales, cupones o promociones**.

DINÁMICA DE LA PROMOCIÓN

Si tu gasto total (compras y devoluciones) del 17 de septiembre al 06 de octubre de 2019

es igual o superior a **30€**, a partir del **08 de octubre de 2019** recibirás un **SMS con importe equivalente al IVA de tu compra a canjear** hasta el **16 de octubre de 2019**.

¿CÓMO CANJEAR EL CUPÓN?

Podrás canjear el cupón obtenido **a partir del 8 de octubre de 2019 en tiendas online** (introduciendo el código del cupón en el momento de pago) **o en tiendas físicas** (presentando el SMS recibido en el momento de pago).

El cupón es válido para una **única compra igual o superior al importe recibido en el cupón**. Si el cupón no se canjea en su totalidad se perderá el importe restante.

En el cómputo del gasto se tendrán en cuenta **todas las compras y devoluciones realizadas entre el 17 de septiembre y el 06 de octubre**. (Incluidas devoluciones durante este periodo de artículos adquiridos antes de la promoción).

El cupón no consumido a 16 de octubre de 2019 no será válido para compras posteriores a esta fecha.

CAMBIOS O DEVOLUCIONES

La devolución de los productos comprados del 17 de septiembre al 06 de octubre de 2019 implica la obligación de la **devolución del importe del descuento obtenido** y por lo tanto la anulación del cupón recibido.

En cambio o devolución de producto, el vale no podrá volver a ser utilizado.

¿CÓMO SE CALCULA EL IVA DE CUALQUIER ARTÍCULO?

- Ejemplo. Para un producto de precio final 100€ con un IVA del 21%.

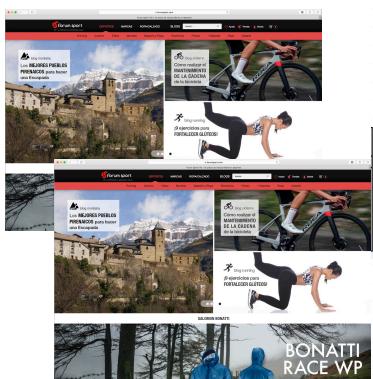
1. Precio de Venta Sin IVA = 82,65€

2. Añadimos el 21% IVA a 82,65€ = 17,35€

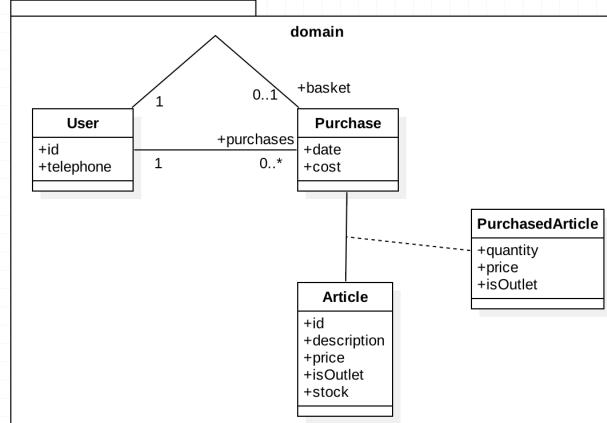
3. Precio final (IVA incluido): 82,65€ + 17,35€ = 100€

Por tanto, para un artículo con Precio final = 100€ con un IVA del 21% el importe a devolver será 17,35€.

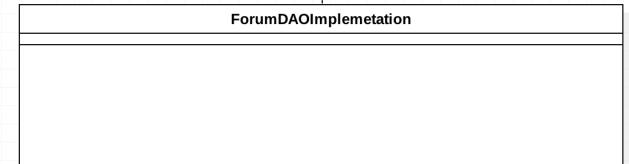
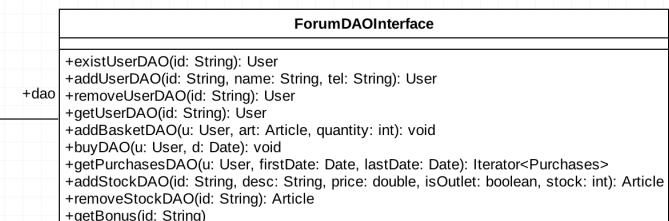
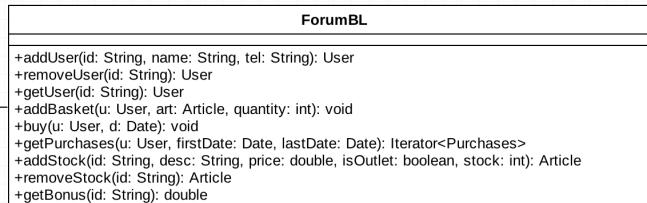
Forum implementatutako sistema



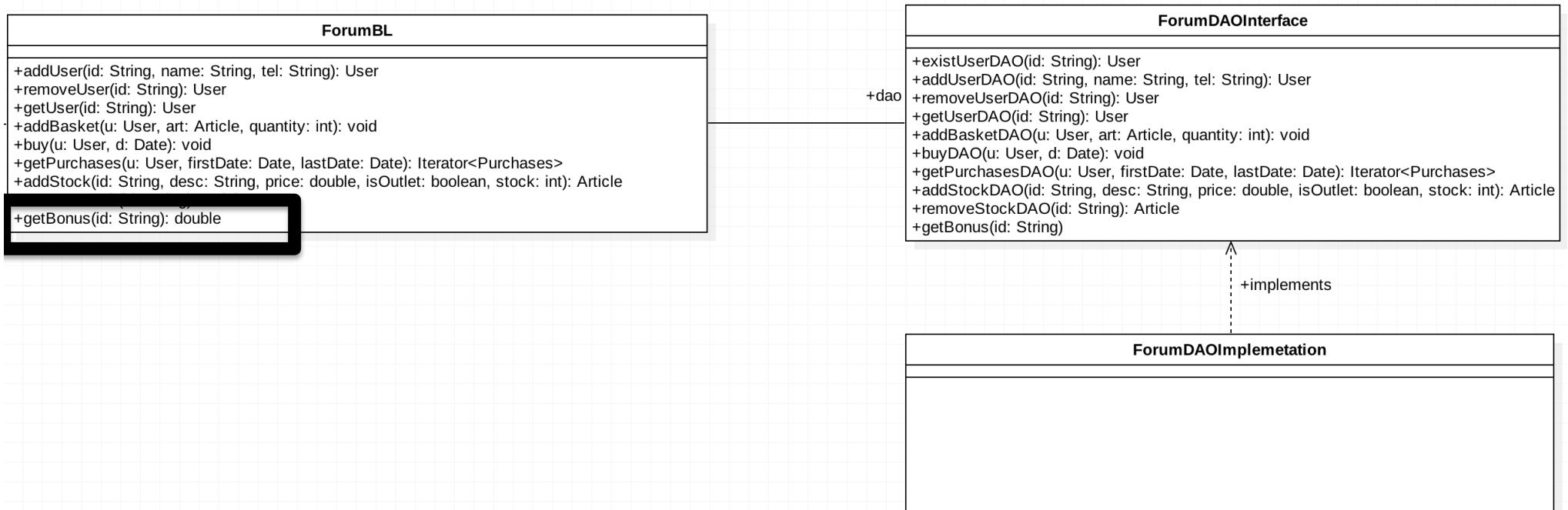
+bl



Domain Model



Frogatu nahi dugun metodoa



getBonus Baliokidetasun Klaseen analisia

Sarrera	Baliokidetasun Klase egokia	Baliokidetasun klase EZ egokiak
id nan bat da	Id nan-a da (5)	id null (0) id ez da nan-a (1) id karaktere bereziak (2)
id bazkidea da	id bazkidea (6)	id ez da bazkidea (3)
id mugikorra aktibatuta	id.telefono ez hutsa (7)	id.telefono hutsa (4)
Ez dira artikularik erosi	Ez dira artikularik erosi promozio epean(8)	
Artikuloak promozio epean erosi dira	Artikuloak erosi dira promozio epean (9)	
outlet artikuloa	Erositako artikuloa outlet da (10)	
EZ outlet artikuloa	Erositako artikuloa outlet EZ da (11)	
erosketen bez-a<50	Erositako erosketen BEZ- a<50 (12)	
erosketen bez-a>=50	Erositako erosketen BEZ- a>=50 (13)	
erosketak>=30	erositako erosketen batura>=30 (14)	
erosketa<30	erositako erosketen batura<30 (15)	

getBonus Proba kasuak

#test	Datubase Egoera	Parametroak	Estalitako BK-ak	Emaitza
0		null	0	Exc
1		75075708K	1	Exc
2		\$\$075708M	2	Exc
3	75075708M ez da bezero bat	75075708M	3	Exc
4	u=new User(75075708M,null)	75075708M	4	Exc
5	u=new User (75075708M, 654478722) NOT u.purchases in promozio denboraldian	75075708M	5,6,7,8	0
6	u=new User (75075708M, 654478722) art=new Article(id1, "asics kayano 16 44",99, true) addBasket(u, art) OUTLET DA buy(u, 25/09/19)	75075708M	5,6,7,9,10	0
7	u=new User (75075708M, 654478722) art2=new Article (id2, "asics nimbus ",150, false,5) addBasket(u, art2) EZ DA OUTLET buy(u, 25/09/19)	75075708M	5,6,7,9,11,12,14	26.02
8	u=new User (75075708M, 654478722) art3=new Article (id3, "orbea alma 29 xtr",1500, false) addBasket(u, art3) buy(u, 25/09/19)	75075708M	5,6,7,9,11,13,14	50
9	u=new User (75075708M, 654478722) art4=new Article (id4, "nike basket socks 39/42",15, false) addBasket(u, art4) buy(u, 25/09/19)	75075708M	5,6,7,9,11,12,15	0

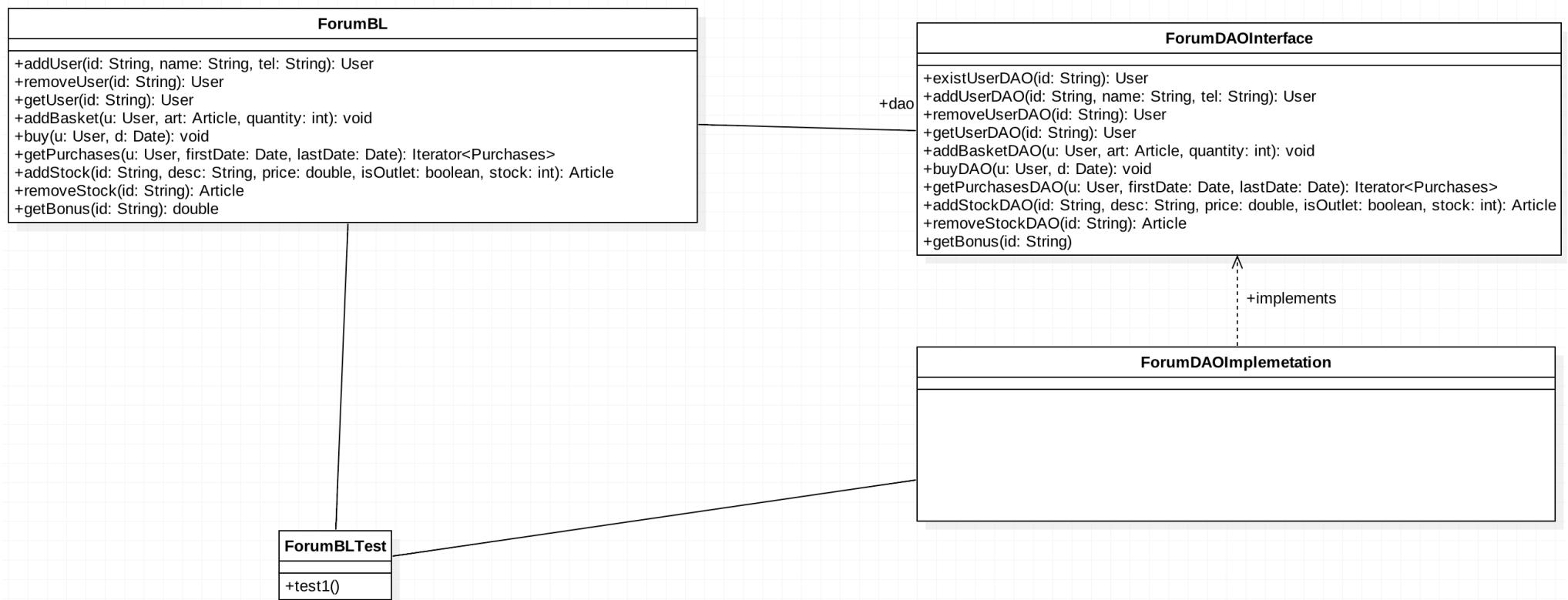
getBonus Proba kasuak

#test	Datubase Egoera	Parametroak	Estalitako BK-ak	Emaitza
7	u=new User (75075708M, 654478722) art2=new Article (id2, "asics nimbus ",150, false,5) addBasket(u, art2) EZ DA OUTLET buy(u, 25/09/19) PROMOZIO DATA BARRUAN	75075708M	5,6,7,9,11,12,14	26.02

2 Implementazio Desberdin

- DataBase mailaren implementazio bat
eskuragarri daukagu => JUnit
- DataBase mailaren interfazea daukagu soilik
=> Junit + Mockito

Implementazio 1: JUnit



getBonus(id) implementazioa

```
public float getBonus(String id) throws Exception {
    if ((id==null) || (!new ValidadorDNI(id).validar()))
        throw new Exception("id null or not valid");

    User u=dao.getUserDAO(id);

    if (u==null) throw new Exception("NAN not in Database");

    if (u.getTelephone()==null) throw new Exception(id+" not registered telephone");

    SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
    Date firstDate = sdf.parse("01/09/2020");
    Date lastDate = sdf.parse("06/12/2020");

    Iterator<Purchase> purchases=dao.getPurchasesDAO(u,firstDate, lastDate);
    float sumPurchases=0;
    float vat=0;
    while (purchases.hasNext()) {
        System.out.println("sum" + sumPurchases);
        Purchase c=purchases.next();
        Iterator <PurchasedArticle> articles=c.getPurchaseIterator();
        while (articles.hasNext()) {
            PurchasedArticle article=articles.next();
            if (!article.isOutlet()) sumPurchases=sumPurchases+article.getPrice()*article.getQuantity();
            System.out.println("sum" + sumPurchases);

        }
    }
    if (sumPurchases>30)
        if (sumPurchases>288) vat=50;
        else vat=(float) (sumPurchases*0.1735);

    return vat;
}
```

Proba kasua

#test	Datubase Egoera	Parametroak	Estalitako BK-ak	Emaitza
7	u=new User (75075708M, 76666666) art2=new Article (id2, "asics nimbus ",150, false,5) addBasket(u, art2) EZ DA OUTLET buy(u, 25/09/19) PROMOZIO DATA BARRUAN	75075708M	5,6,7,9,11,12,14	26.02

```

public class ForumBLTest {
    ForumDAOInterface forumDAO=new ForumInMemoryDAO();
    ForumBL sut=new ForumBL(forumDAO);

    @Test
    public void test() {
        //define parameters
        String nan="75075708M";
        float expectedValue=(float)26.025;

        //prepare the database
        User u=sut.addUser(nan, "jon", "7666666");
        Article art2=new Article("2","asics nimbus",150,false,5);
        sut.addBasket(u, art2, 1);
        sut.buy(u, new Date());

        try {
            //Invoke the method to test
            float bono=sut.getBonus(nan);

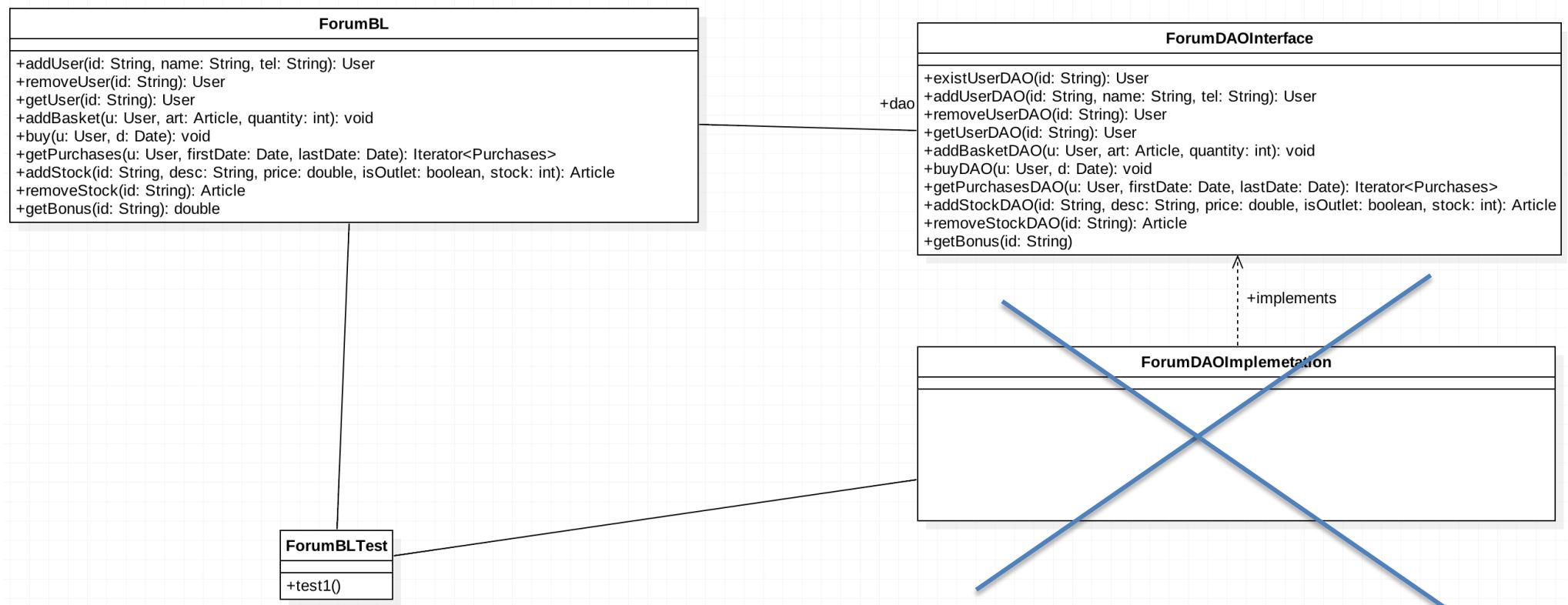
            //reestablish the database
            sut.removeUser(nan);
            sut.removeStock("2");

            //check the result
            assertTrue(bono==expectedValue);
        } catch (Exception e) {
            // TODO Auto-generated catch block
            //reestablish the database

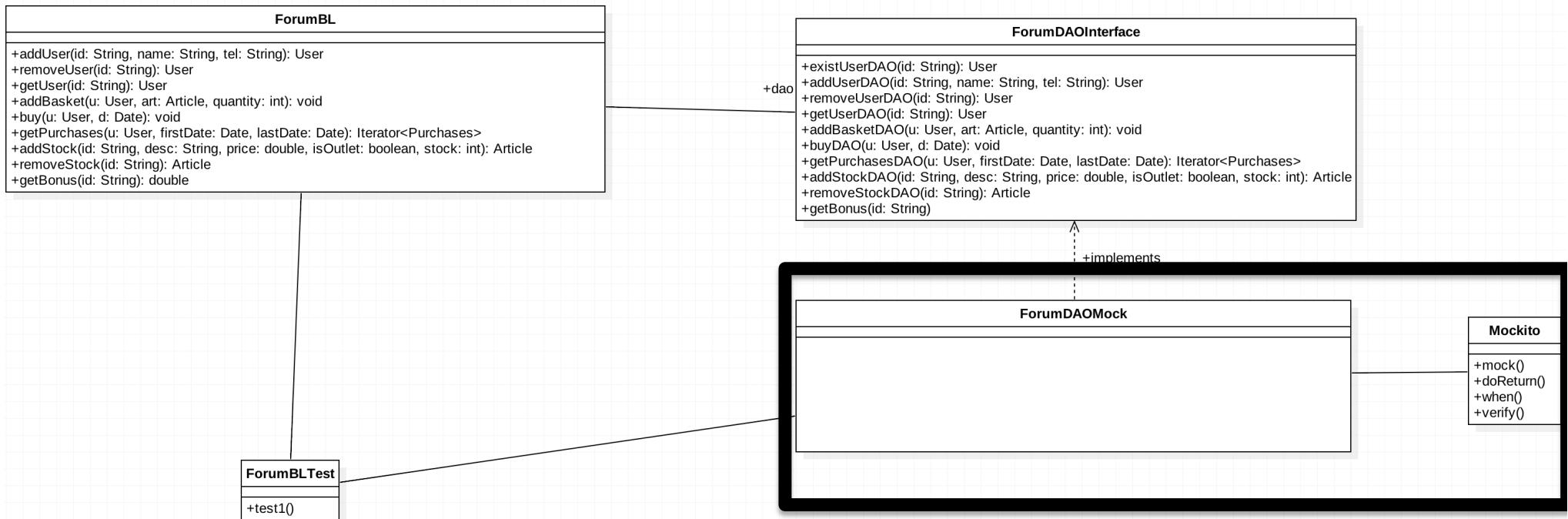
            sut.removeUser(nan);
            sut.removeStock("2");
            e.printStackTrace();
            fail();
        }
    }
}

```

Implementazio 2: Ez daukagu ForumDAOImplementation



Implementazio 2: JUnit + Mockito



Nola sortu Mock objektu bat

```
RunWith(MockitoJUnitRunner.class)
public class ForumPromoTestMockito {

    @Mock
    ForumDAOInterface forumDAO;

    //Equivalent to
    //ForumDAOInterface forumDAO=Mockito.mock(ForumDAOInterface.class);

    // sut means System Under Test
    @InjectMocks
    ForumBL sut;

    //Equivalent to
    //ForumBL sut=new ForumBL(forumDAO);
    @Test
    public void test0() {
        assertTrue(true);
    }
}
```

Eragiketak defektuzko balioak itzultzen dute

```
RunWith(MockitoJUnitRunner.class)
public class ForumPromoTestMockito {

    @Mock
    ForumDAOInterface forumDAO;

    //Equivalent to
    //ForumDAOInterface forumDAO=Mockito.mock(ForumDAOInterface.class);

    // sut means System Under Test
    @InjectMocks
    ForumBL sut;

    //Equivalent to
    //ForumBL sut=new ForumBL(forumDAO);
    @Test
    public void test0() {
        forumDAO.getUserDAO("75075708M") => null
    }
}
```

Mock konfiguratzen

```
RunWith(MockitoJUnitRunner.class)
public class ForumPromoTestMockito {

    @Mock
    ForumDAOInterface forumDAO;

    //Equivalent to
    //ForumDAOInterface forumDAO=Mockito.mock(ForumDAOInterface.class);

    // sut means System Under Test
    @InjectMocks
    ForumBL sut;

    //Equivalent to
    //ForumBL sut=new ForumBL(forumDAO);
    @Test
    public void test0() {

        User usr=new User("75075708M", "654478722");
        Article art2=new Article("2", "asics nimbus 16 44",150, false);
        usr.addBasket(art2,1);
        usr.buy(new Date());

        Mockito.doReturn(usr).when(forumDAO).getUser("75075708M");

        forumDAO.getUserDAO("75075708M") => usr
    }

}
```

7. Proba kasua JUnit + Mockito

```
@RunWith(MockitoJUnitRunner.class)
public class ForumPromoTestMockito {
    @Mock
    ForumDAOInterface forumDAO;

    // sut means System Under Test
    @InjectMocks
    ForumPromotionBL sut;

    @Test
    public void test7() {
        User usr=new User("75075708M", "654478722");
        Article art2=new Article("2", "asics nimbus 16 44",150, false);
        usr.addBasket(art2,1);
        usr.buy(new Date());

        SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
        try {
            Date firstDate = sdf.parse("17/09/2019");
            Date lastDate = sdf.parse("06/10/2019");

            Mockito.doReturn(usr).when(forumDAO).getUser("75075708M");

            Mockito.doReturn(usr.getPurchases(firstDate,lastDate))
                .when(forumDAO).getPurchases(Mockito.eq(usr), Mockito.any(Date.class), Mockito.any(Date.class));

        } catch (ParseException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
        float expectedValue=(float) 26.025;

        try {
            float bono=sut.getBonus("75075708M");
            assertTrue(bono==expectedValue);
        } catch (Exception e) {
            // TODO Auto-generated catch block
            fail();
        }
    }
}
```

Ariketa 1

- “654478722” telefono duen erabiltzaile batek “75075708M” ez outlet diren 100 eta 115eko bi artikulo erosi ditu erosketa batean
- Frogatu emaitza zuzena den JUnit eta JUnit+Mockito ikuspegia jarraituz.

Ariketa 2

- “654478722” telefono duen erabiltzaile batek “75075708M” ez outlet diren 100 eta 125eko bi artikulo erosi ditu erosketa batean, eta 80 eta 90eko beste bi artikulo beste erosketa batean, bigarrengo erosketa honetako bigarren artikulua outlet izanda.
- Test-ak implementatu JUnit eta JUnit+Mockito ikuspegia jarraituz.

Ariketa 3

- Muga-balioen analisia egin erosketaren BEZ-a eta erosketen prezioarako.
- Frogak kasuak implementatu
 - JUnit
 - JUnit+Mockito
- Beste proba interesgarriren bat probatuko zenuke?
- Akatsaren bat aurkitu ahal duzue?

Entrega

- Taldeko lana
- Entrega data. 24/09/19
- Balorazioa: GAI/EZ GAI