DataAccess-eko Errefaktorizazioak

Jon Ander Jimenez, Mijel Mujika eta Ibai Heras

Hasierako kodea: “Duplicate code”

```
if (Locale.getDefault().equals(new Locale("es"))) {
    q1=ev1.addQuestion("Â¿QuiÃ©n ganarÃ¡ el partido?",1);
    q2=ev1.addQuestion("Â¿QuiÃ©n meterÃ¡ el primer gol?",2);
    q3=ev11.addQuestion("Â¿QuiÃ©n ganarÃ¡ el partido?",1);
    q4=ev11.addQuestion("Â¿CuÃ¡ntos goles se marcarÃ¡n?",2);
    q5=ev17.addQuestion("Â¿QuiÃ©n ganarÃ¡ el partido?",1);
    q6=ev17.addQuestion("Â¿HabrÃ¡ goles en la primera parte?",2);

}
else if (Locale.getDefault().equals(new Locale("en"))) {
    q1=ev1.addQuestion("Who will win the match?",1);
    q2=ev1.addQuestion("Who will score first?",2);
    q3=ev11.addQuestion("Who will win the match?",1);
    q4=ev11.addQuestion("How many goals will be scored in the match?",2);
    q5=ev17.addQuestion("Who will win the match?",1);
    q6=ev17.addQuestion("Will there be goals in the first half?",2);
}
else {
    q1=ev1.addQuestion("Zeinek irabaziko du partidua?",1);
    q2=ev1.addQuestion("Zeinek sartuko du lehenengo gola?",2);
    q3=ev11.addQuestion("Zeinek irabaziko du partidua?",1);
    q4=ev11.addQuestion("Zenbat gol sartuko dira?",2);
    q5=ev17.addQuestion("Zeinek irabaziko du partidua?",1);
    q6=ev17.addQuestion("Golak sartuko dira lehenengo zatian?",2);
```

Errefaktorizatuko kodea:

```
if (Locale.getDefault().equals(new Locale("es"))) {
    String s = "Â¿QuiÃ©n ganarÃ¡ el partido?";
    q1=ev1.addQuestion(s,1);
    q2=ev1.addQuestion("Â¿QuiÃ©n meterÃ¡ el primer gol?",2);
    q3=ev11.addQuestion(s,1);
    q4=ev11.addQuestion("Â¿CuÃ¡ntos goles se marcarÃ¡n?",2);
    q5=ev17.addQuestion(s,1);
    q6=ev17.addQuestion("Â¿HabrÃ¡ goles en la primera parte?",2);

}
else if (Locale.getDefault().equals(new Locale("en"))) {
    String s = "Who will win the match?";
    q1=ev1.addQuestion(s,1);
    q2=ev1.addQuestion("Who will score first?",2);
    q3=ev11.addQuestion(s,1);
    q4=ev11.addQuestion("How many goals will be scored in the match?",2);
    q5=ev17.addQuestion(s,1);
    q6=ev17.addQuestion("Will there be goals in the first half?",2);
}
else {
    String s = "Zeinek irabaziko du partidua?";
    q1=ev1.addQuestion(s,1);
    q2=ev1.addQuestion("Zeinek sartuko du lehenengo gola?",2);
    q3=ev11.addQuestion(s,1);
    q4=ev11.addQuestion("Zenbat gol sartuko dira?",2);
    q5=ev17.addQuestion(s,1);
    q6=ev17.addQuestion("Golak sartuko dira lehenengo zatian?",2);
```

Azalpena:

Kode honetan galdera bera hiru aldiz errepikatzen da hizkuntza bakoitzeko. Horretarako bariable berri bat sortu dut string bakoitzerako eta honela string bera errepikatu ordez, sortutako bariablea erabili dut.

Ibai Heras

Hasierako kodea: “Duplicate code”

```
Transaction t1 = new Transaction(reg1, apA1.getBalioa(), new Date(), "ApustuEgin");
Transaction t3 = new Transaction(reg2, apA4.getBalioa(), new Date(), "ApustuEgin");
Transaction t4 = new Transaction(reg3, apA5.getBalioa(), new Date(), "ApustuEgin");
Transaction t5 = new Transaction(reg4, apA3.getBalioa(), new Date(), "ApustuEgin");
Transaction t6 = new Transaction(reg4, apA6.getBalioa(), new Date(), "ApustuEgin");
Transaction t7 = new Transaction(reg1, apA7.getBalioa(), new Date(), "ApustuEgin");
Transaction t8 = new Transaction(reg1, apA8.getBalioa(), new Date(), "ApustuEgin");
Transaction t9 = new Transaction(reg2, apA9.getBalioa(), new Date(), "ApustuEgin");
Transaction t10 = new Transaction(reg2, apA10.getBalioa(), new Date(), "ApustuEgin");
Transaction t11 = new Transaction(reg3, apA11.getBalioa(), new Date(), "ApustuEgin");
Transaction t12 = new Transaction(reg3, apA12.getBalioa(), new Date(), "ApustuEgin");
```

Errefaktorizatuko kodea:

```
String apustuEgin = "ApustuEgin";

Transaction t1 = new Transaction(reg1, apA1.getBalioa(), new Date(), apustuEgin );
Transaction t3 = new Transaction(reg2, apA4.getBalioa(), new Date(), apustuEgin);
Transaction t4 = new Transaction(reg3, apA5.getBalioa(), new Date(), apustuEgin);
Transaction t5 = new Transaction(reg4, apA3.getBalioa(), new Date(), apustuEgin);
Transaction t6 = new Transaction(reg4, apA6.getBalioa(), new Date(), apustuEgin);
Transaction t7 = new Transaction(reg1, apA7.getBalioa(), new Date(), apustuEgin);
Transaction t8 = new Transaction(reg1, apA8.getBalioa(), new Date(), apustuEgin);
Transaction t9 = new Transaction(reg2, apA9.getBalioa(), new Date(), apustuEgin);
Transaction t10 = new Transaction(reg2, apA10.getBalioa(), new Date(), apustuEgin);
Transaction t11 = new Transaction(reg3, apA11.getBalioa(), new Date(), apustuEgin);
Transaction t12 = new Transaction(reg3, apA12.getBalioa(), new Date(), apustuEgin);
```

Azalpena:

Kode honetan transakzio bakoitza sortzerako orduan, “ApustuEgin” String-a errepikatzen da kode lerro bakoitzean. Horretarako, String berri bat sortu dut apustuEgin izenekoa eta errepikatzen den String-aren ordez ordezkatu dut.

Ibai Heras

Hasierako kodea: “Duplicate code”

```
this.DiruaSartu(reg1, 50.0, new Date(), "DiruaSartu");
this.DiruaSartu(reg2, 50.0, new Date(), "DiruaSartu");
this.DiruaSartu(reg3, 50.0, new Date(), "DiruaSartu");
this.DiruaSartu(reg4, 50.0, new Date(), "DiruaSartu");
```

Errefaktorizatuko kodea:

```
String diruSartu = "DiruaSartu";
this.DiruaSartu(reg1, 50.0, new Date(), diruSartu);
this.DiruaSartu(reg2, 50.0, new Date(), diruSartu);
this.DiruaSartu(reg3, 50.0, new Date(), diruSartu);
this.DiruaSartu(reg4, 50.0, new Date(), diruSartu);
```

Azalpena:

Kode zati honetan String bera 4 aldiz agertzen da, hori ekiditeko String berri bat sortu dut diruSartu izenekoa eta errepikapen bakoitzarekin ordezkatu dut.

Ibai Heras

Hasierako kodea: "Write short units of code" eta "Write simple units of code"

```
public boolean ApustuaEgin(Registered u, LinkedList<Quote> quote, Double balioa, Integer apustuBikoitzaGalarazi) {
    Registered user = (Registered) db.find(Registered.class, u.getUsername());
    Boolean b;
    if(user.getDirukop()>=balioa) {
        db.getTransaction().begin();
        ApustuAnitza apustuAnitza = new ApustuAnitza(user, balioa);
        db.persist(apustuAnitza);
        for(Quote quo: quote) {
            Quote kuote = db.find(Quote.class, quo.getQuoteNumber());
            Apustua ap = new Apustua(apustuAnitza, kuote);
            db.persist(ap);
            apustuAnitza.addApustua(ap);
            kuote.addApustua(ap);
        }
        db.getTransaction().commit();
        db.getTransaction().begin();
        if(apustuBikoitzaGalarazi==-1) {
            apustuBikoitzaGalarazi=apustuAnitza.getApustuAnitzaNumber();
        }
        apustuAnitza.setApustuKopia(apustuBikoitzaGalarazi);
        user.updateDiruKontua(-balioa);
        Transaction t = new Transaction(user, balioa, new Date(), "ApustuaEgin");
        user.addApustuAnitza(apustuAnitza);
        for(Apustua a: apustuAnitza.getApustuak()) {
            Apustua apu = db.find(Apustua.class, a.getApostuaNumber());
            Quote q = db.find(Quote.class, apu.getKuota().getQuoteNumber());
            Sport spo =q.getQuestion().getEvent().getSport();
            spo.setApustuKantitatea(spo.getApustuKantitatea()+1);
        }
        user.addTransaction(t);
        db.persist(t);
        db.getTransaction().commit();
        for(Jarraitzailea reg:user.getJarraitzaileLista()) {
            Jarraitzailea erab=db.find(Jarraitzailea.class, reg.getJarraitzaileaNumber());
            b=true;
            for(ApustuAnitza apu: erab.getNork().getApustuAnitzak()) {
                if(apu.getApustuKopia()==apustuAnitza.getApustuKopia()) {
                    b=false;
                }
            }
            if(b) {
                if(erab.getNork().getDiruLimitea()<balioa) {
                    this.ApustuaEgin(erab.getNork(), quote, erab.getNork().getDiruLimitea(), apustuBikoitzaGalarazi);
                }else{
                    this.ApustuaEgin(erab.getNork(), quote, balioa, apustuBikoitzaGalarazi);
                }
            }
        }
        return true;
    }else{
        return false;
    }
}
```

Errefaktorizatuko kodea:

```
public boolean ApustuaEgin(Registered u, LinkedList<Quote> quote, Double balioa, Integer apustuBikoitzaGalarazi) {
    Registered user = getUser(u);

    if (user.getDirukop() >= balioa) {
        ApustuAnitza apustuAnitza = createApustuAnitza(user, balioa, quote, apustuBikoitzaGalarazi);
        updateSport(apustuAnitza);
        addTransactionsAndFollowers(user, apustuAnitza, quote, balioa, apustuBikoitzaGalarazi);

        return true;
    } else {
        return false;
    }
}

private Registered getUser(Registered u) {
    return (Registered) db.find(Registered.class, u.getUsername());
}

private ApustuAnitza createApustuAnitza(Registered user, Double balioa, LinkedList<Quote> quote, Integer apustuBikoitzaGalarazi) {
    db.getTransaction().begin();
    ApustuAnitza apustuAnitza = new ApustuAnitza(user, balioa);
    db.persist(apustuAnitza);
    for (Quote quo : quote) {
        Quote kuote = db.find(Quote.class, quo.getQuoteNumber());
        Apustua ap = new Apustua(apustuAnitza, kuote);
        db.persist(ap);
        apustuAnitza.addApustua(ap);
        kuote.addApustua(ap);
    }
    db.getTransaction().commit();
    db.getTransaction().begin();
    if (apustuBikoitzaGalarazi == -1) {
        apustuBikoitzaGalarazi = apustuAnitza.getApustuAnitzaNumber();
    }
    apustuAnitza.setApustuKopia(apustuBikoitzaGalarazi);
    user.updateDiruKontua(-balioa);
    return apustuAnitza;
}

private void updateSport(ApustuAnitza apustuAnitza) {
    for (Apustua a : apustuAnitza.getApustuak()) {
        Apustua apu = db.find(Apustua.class, a.getApostuaNumber());
        Quote q = db.find(Quote.class, apu.getKuota().getQuoteNumber());
        Sport spo = q.getQuestion().getEvent().getSport();
        spo.setApustuKantitatea(spo.getApustuKantitatea() + 1);
    }
}
```

```
private void addTransactionsAndFollowers(Registered user, ApustuAnitza apustuAnitza, LinkedList<Quote> quote, Double balioa, Integer apustuBikoitzaGalarazi) {
    Transaction t = new Transaction(user, balioa, new Date(), "ApustuaEgin");
    user.addApustuAnitza(apustuAnitza);
    user.addTransaction(t);
    db.persist(t);
    db.getTransaction().commit();

    for (Jarraitzailea reg : user.getJarraitzaileLista()) {
        Jarraitzailea erab = db.find(Jarraitzailea.class, reg.getJarraitzaileaNumber());
        boolean b = true;
        for (ApustuAnitza apu : erab.getNork().getApustuAnitzak()) {
            if (apu.getApustuKopia() == apustuAnitza.getApustuKopia()) {
                b = false;
            }
        }
        if (b) {
            if (erab.getNork().getDiruLimitea() < balioa) {
                this.ApustuaEgin(erab.getNork(), quote, erab.getNork().getDiruLimitea(), apustuBikoitzaGalarazi);
            } else {
                this.ApustuaEgin(erab.getNork(), quote, balioa, apustuBikoitzaGalarazi);
            }
        }
    }
}
```

Azalpena:

Funtzio hau errezagoa eta motzagoa izateko 4 azpi funtzioetan banatu dut. Bat apustu anitza sortzeko, beste bat apustu anitzak eguneratzeko, datu basean usuarioa bilatzeko eta azkenik tratsakzioak gehitzeko. Honela funtzio hau sinpleago eta motzagoa geratzen da bere funtzionamendua aldatu gabe.

Ibai Heras

Write short units of code

Hasierako kodea:

```
public void apustuaEzabatu(Registered user1, ApustuAnitza ap) {
    Registered user = (Registered) db.find(Registered.class, user1.getUsername());
    ApustuAnitza apustuAnitza = db.find(ApustuAnitza.class, ap.getApustuAnitzaNumber());
    db.getTransaction().begin();
    user.updateDiruKontua(apustuAnitza.getBalioa());
    Transaction t = new Transaction(user, apustuAnitza.getBalioa(), new Date(), "ApustuaEzabatu");
    user.addTransaction(t);
    db.persist(t);
    user.removeApustua(apustuAnitza);
    for(int i=0; i<apustuAnitza.getApustuak().size(); i++) {
        apustuAnitza.getApustuak().get(i).getKuota().removeApustua(apustuAnitza.getApustuak().get(i));
        Sport spo =apustuAnitza.getApustuak().get(i).getKuota().getQuestion().getEvent().getSport();
        spo.setApustuKantitatea(spo.getApustuKantitatea()-1);
    }
    db.remove(apustuAnitza);
    db.getTransaction().commit();
}
```

Errefaktorizatuko kodea:

```
public void apustuaEzabatu(Registered user1, ApustuAnitza ap) {
    Registered user = (Registered) db.find(Registered.class, user1.getUsername());
    ApustuAnitza apustuAnitza = db.find(ApustuAnitza.class, ap.getApustuAnitzaNumber());
    db.getTransaction().begin();
    user.updateDiruKontua(apustuAnitza.getBalioa());
    Transaction t = new Transaction(user, apustuAnitza.getBalioa(), new Date(), "ApustuaEzabatu");
    user.addTransaction(t);
    db.persist(t);
    user.removeApustua(apustuAnitza);
    apusutuAnitzaBerritu(apustuAnitza);
    db.remove(apustuAnitza);
    db.getTransaction().commit();
}

public void apusutuAnitzaBerritu(ApustuAnitza ap) {
    ApustuAnitza apustuAnitza = db.find(ApustuAnitza.class, ap.getApustuAnitzaNumber());
    db.getTransaction().begin();
    for(int i=0; i<apustuAnitza.getApustuak().size(); i++) {
        apustuAnitza.getApustuak().get(i).getKuota().removeApustua(apustuAnitza.getApustuak().get(i));
        Sport spo =apustuAnitza.getApustuak().get(i).getKuota().getQuestion().getEvent().getSport();
        spo.setApustuKantitatea(spo.getApustuKantitatea()-1);
    }
    db.getTransaction().commit();
}
```

Ikusten den bezala metodo berri bat sortu dut. For-ra hartu dut eta beste metodo batean sartu dut 15 lerro baino gehiago zuelako metodoa, hala ere egiterakoan eta datu basean aldaketak egiteko begin eta commit jarri dut eta apustuanitza datu basean bilatu dut.

Jon Ander Jimenez

Write simple units of code

Hasierako kodea:

```
public boolean gertaerakSortu(String description,Date eventDate, String sport) {
    boolean b = true;
    db.getTransaction().begin();
    Sport spo =db.find(Sport.class, sport);
    if(spo!=null) {
        TypedQuery<Event> Equery = db.createQuery("SELECT e FROM Event e WHERE e.getEventDate() =?1 ",Event.class);
        Equery.setParameter(1, eventDate);
        for(Event ev: Equery.getResultList()) {
            if(ev.getDescription().equals(description)) {
                b = false;
            }
        }
        if (b) {
            String[] taldeak = description.split("-");
            Team lokala = new Team(taldeak[0]);
            Team kanpokoa = new Team(taldeak[1]);
            Event e = new Event(description, eventDate, lokala, kanpokoa);
            e.setSport(spo);
            spo.addEvent(e);
            db.persist(e);
        }
    } else {
        db.getTransaction().commit();
        return false;
    }
    db.getTransaction().commit();
    return b;
}
```

Errefaktorizatuko kodea:

```
public boolean gertaerakSortu(String description, Date eventDate, String sport) {
    boolean b = true;
    db.getTransaction().begin();
    Sport spo = db.find(Sport.class, sport);
    if (spo != null) {
        b = eventDescriptionGS(description, eventDate);
        if (b) {
            gertaeraDatuBaseanSartuGS(description, eventDate, spo);
        }
    } else {
        db.getTransaction().commit();
        return false;
    }
    db.getTransaction().commit();
    return b;
}

public void gertaeraDatuBaseanSartuGS(String description, Date eventDate, Sport sport) {
    db.getTransaction().begin();
    Sport spo = db.find(Sport.class, sport);
    String[] taldeak = description.split("-");
    Team lokala = new Team(taldeak[0]);
    Team kanpokoa = new Team(taldeak[1]);
    Event e = new Event(description, eventDate, lokala, kanpokoa);
    e.setSport(spo);
    spo.addEvent(e);
    db.persist(e);
    db.getTransaction().commit();
}

public boolean eventDescriptionGS(String description, Date eventDate) {
    db.getTransaction().begin();
    boolean b = true;
    TypedQuery<Event> Equery = db.createQuery("SELECT e FROM Event e WHERE e.getEventDate() =?1 ", Event.class);
    Equery.setParameter(1, eventDate);
    for (Event ev : Equery.getResultList()) {
        if (ev.getDescription().equals(description)) {
            b = false;
        }
    }
    db.getTransaction().commit();
    return b;

}
```

gertaeraSortu metodoa errazago eta zatitan banatzeko best bi azpimetodo egin ditut. Batek gertaera datu basean sartzen du eta beste for baten bitartez konparatzen ditu eventuaren descriptionak ea berdin bat dagoen. Hauek bi metodo nagusian erabiltzen dira eta programaren portaera berdina izango da.

Jon Ander Jimenez

Duplicate code

Hasierako kodea:

```
public BLFacadeImplementation()  {
    System.out.println("Creating BLFacadeImplementation instance");
    ConfigXML c=ConfigXML.getInstance();

    if (c.getDataBaseOpenMode().equals("initialize")) {
        dbManager=new DataAccess(c.getDataBaseOpenMode().equals("initialize"));
        dbManager.initializeDB();
        } else
         dbManager=new DataAccess();
    dbManager.close();


}

public BLFacadeImplementation(DataAccess da)  {

    System.out.println("Creating BLFacadeImplementation instance with DataAccess parameter");
    ConfigXML c=ConfigXML.getInstance();

    if (c.getDataBaseOpenMode().equals("initialize")) {
        da.open(true);
        da.initializeDB();
        da.close();

    }
    dbManager=da;
}
```

Errefaktorizatuko kodea:

```
public BLFacadeImplementation()  {
    System.out.println("Creating BLFacadeImplementation instance");
    ConfigXML c=ConfigXML.getInstance();

    if (dateBaseIsInitialize(c)) {
        dbManager=new DataAccess(dateBaseIsInitialize(c));
        dbManager.initializeDB();
        } else
         dbManager=new DataAccess();
    dbManager.close();


}



public BLFacadeImplementation(DataAccess da)  {

    System.out.println("Creating BLFacadeImplementation instance with DataAccess parameter");
    ConfigXML c=ConfigXML.getInstance();

    if (dateBaseIsInitialize(c)) {
        da.open(true);
        da.initializeDB();
        da.close();

    }
    dbManager=da;
}
private boolean dateBaseIsInitialize(ConfigXML c) {
    return c.getDataBaseOpenMode().equals("initialize");
}
```

Kasu honetan BLFacadeImplementation metodora joan naiz, data access-en duplicate kode gehiago ez daudelako. Lehenengo argazkian ikusten dugu bi metodoetan if berdina egiten dela, horregatik funtzio boolear bat egin dut kodea ez errepikatzeko. Kasu honetan private metodo bat egin dut bakarrik klase horretan exekutatzea nahi dudalako.

Jon Ander Jimenez

Keep unit interfaces small

Hasierako kodea:

```
private ApustuAnitza createApustuAnitza(Registered user, Double balioa, LinkedList<Quote> quote, Integer apustuBikoitzaGalarazi) {
    db.getTransaction().begin();
    ApustuAnitza apustuAnitza = new ApustuAnitza(user, balioa);
    db.persist(apustuAnitza);
    for (Quote quo : quote) {
        Quote kuote = db.find(Quote.class, quo.getQuoteNumber());
        Apustua ap = new Apustua(apustuAnitza, kuote);
        db.persist(ap);
        apustuAnitza.addApustua(ap);
        kuote.addApustua(ap);
    }
    db.getTransaction().commit();
    db.getTransaction().begin();
    if (apustuBikoitzaGalarazi == -1) {
        apustuBikoitzaGalarazi = apustuAnitza.getApustuAnitzaNumber();
    }
    apustuAnitza.setApustuKopia(apustuBikoitzaGalarazi);
    user.updateDiruKontua(-balioa);
    return apustuAnitza;
}
```

Errefaktorizatuko kodea:

```
public ApustuAnitza createApustuAnitza(Registered user, Double balioa, LinkedList<Quote> quote,Integer apustuBikoitzaGalarazi) {
    ApustuAnitza apustuAnitza = createNewApustuAnitzaCAA(user, balioa);
    associateQuotesWithApustuCAA(quote, apustuAnitza);
    updateApustuKopiaCAA(apustuBikoitzaGalarazi, apustuAnitza);
    updateDiruaCAA(user, balioa);

    return apustuAnitza;
}

private ApustuAnitza createNewApustuAnitzaCAA(Registered user, Double balioa) {
    db.getTransaction().begin();
    ApustuAnitza apustuAnitza = new ApustuAnitza(user, balioa);
    db.persist(apustuAnitza);
    db.getTransaction().commit();
    return apustuAnitza;
}

private void associateQuotesWithApustuCAA(LinkedList<Quote> quote, ApustuAnitza apustuAnitza) {
    db.getTransaction().begin();
    for (Quote quo : quote) {
        Quote kuote = db.find(Quote.class, quo.getQuoteNumber());
        Apustua ap = new Apustua(apustuAnitza, kuote);
        db.persist(ap);
        apustuAnitza.addApustua(ap);
        kuote.addApustua(ap);
    }
    db.getTransaction().commit();
}

private void updateApustuKopiaCAA(Integer apustuBikoitzaGalarazi, ApustuAnitza apustuAnitza) {
    if (apustuBikoitzaGalarazi == -1) {
        apustuBikoitzaGalarazi = apustuAnitza.getApustuAnitzaNumber();
    }
    db.getTransaction().begin();
    apustuAnitza.setApustuKopia(apustuBikoitzaGalarazi);
    db.getTransaction().commit();
}

private void updateDiruaCAA(Registered user, Double balioa) {
    db.getTransaction().begin();
    user.updateDiruKontua(-balioa);
    db.getTransaction().commit();
}
```

createApustuAnitza sarrerako parametro asko ditu eta sinplifikatzeko eta klase berri bat ez egiteko, hainbat metodo desberdinetan banatu dut. Horrela metodo erraza izango, lerro kopuru txikiagoa du eta aldaketak egiteko ulergarriagoa izango da etorkizunean.

Jon Ander Jimenez

Write short units of code

Hasierako kodea:

```
private void addTransactionsAndFollowers(Registered user, ApustuAnitza apustuAnitza, LinkedList<Quote> quote,
        Double balioa, Integer apustuBikoitzaGalarazi) {
    Transaction t = new Transaction(user, balioa, new Date(), "ApustuaEgin");
    user.addApustuAnitza(apustuAnitza);
    user.addTransaction(t);
    db.persist(t);
    db.getTransaction().commit();

    for (Jarraitzailea reg : user.getJarraitzaileLista()) {
        Jarraitzailea erab = db.find(Jarraitzailea.class, reg.getJarraitzaileaNumber());
        boolean b = true;
        for (ApustuAnitza apu : erab.getNork().getApustuAnitzak()) {
            if (apu.getApustuKopia() == apustuAnitza.getApustuKopia()) {
                b = false;
            }
        }
        if (b) {
            if (erab.getNork().getDiruLimitea() < balioa) {
                this.ApustuaEgin(erab.getNork(), quote, erab.getNork().getDiruLimitea(), apustuBikoitzaGalarazi);
            } else {
                this.ApustuaEgin(erab.getNork(), quote, balioa, apustuBikoitzaGalarazi);
            }
        }
    }
}
```

Errefaktorizatuko kodea:

```
private void addTransactionsAndFollowers(Registered user, ApustuAnitza apustuAnitza, LinkedList<Quote> quote,
        Double balioa, Integer apustuBikoitzaGalarazi) {
    Transaction t = new Transaction(user, balioa, new Date(), "ApustuaEgin");
    user.addApustuAnitza(apustuAnitza);
    user.addTransaction(t);
    db.persist(t);
    db.getTransaction().commit();

    for (Jarraitzailea reg : user.getJarraitzaileLista()) {
        Jarraitzailea erab = db.find(Jarraitzailea.class, reg.getJarraitzaileaNumber());
        boolean b = apustuAnitzaErab(erab,apustuAnitza);
        if (b) {
            if (erab.getNork().getDiruLimitea() < balioa) {
                this.ApustuaEgin(erab.getNork(), quote, erab.getNork().getDiruLimitea(), apustuBikoitzaGalarazi)
            } else {
                this.ApustuaEgin(erab.getNork(), quote, balioa, apustuBikoitzaGalarazi);
            }
        }
    }
}

public boolean apustuAnitzaErab(Jarraitzailea erab,ApustuAnitza apustuAnitza) {
    for (ApustuAnitza apu : erab.getNork().getApustuAnitzak()) {
        if (apu.getApustuKopia() == apustuAnitza.getApustuKopia()) {
            return false;
        }
    }
    return true;
}
```

Ikusten den bezala metodo berri bat sortu dut. For-ra hartu dut eta beste metodo batean sartu dut 15 lerro baino gehiago zuelako metodoa.

Mikel Mugica

Write simple units of code

Hasierako kodea:

```
private void addTransactionsAndFollowers(Registered user, ApustuAnitza apustuAnitza, LinkedList<Quote> quote,
        Double balioa, Integer apustuBikoitzaGalarazi) {
    Transaction t = new Transaction(user, balioa, new Date(), "ApustuaEgin");
    user.addApustuAnitza(apustuAnitza);
    user.addTransaction(t);
    db.persist(t);
    db.getTransaction().commit();

    for (Jarraitzailea reg : user.getJarraitzaileLista()) {
        Jarraitzailea erab = db.find(Jarraitzailea.class, reg.getJarraitzaileaNumber());
        boolean b = true;
        for (ApustuAnitza apu : erab.getNork().getApustuAnitzak()) {
            if (apu.getApustuKopia() == apustuAnitza.getApustuKopia()) {
                b = false;
            }
        }
        if (b) {
            if (erab.getNork().getDiruLimitea() < balioa) {
                this.ApustuaEgin(erab.getNork(), quote, erab.getNork().getDiruLimitea(), apustuBikoitzaGalarazi);
            } else {
                this.ApustuaEgin(erab.getNork(), quote, balioa, apustuBikoitzaGalarazi);
            }
        }
    }
}
```

Errefaktorizatuko kodea:

```
private void addTransactionsAndFollowers(Registered user, ApustuAnitza apustuAnitza, LinkedList<Quote> quote,
        Double balioa, Integer apustuBikoitzaGalarazi) {
    Transaction t = new Transaction(user, balioa, new Date(), "ApustuaEgin");
    user.addApustuAnitza(apustuAnitza);
    user.addTransaction(t);
    db.persist(t);
    db.getTransaction().commit();

    for (Jarraitzailea reg : user.getJarraitzaileLista()) {
        Jarraitzailea erab = db.find(Jarraitzailea.class, reg.getJarraitzaileaNumber());
        boolean b = apustuAnitzaErab(erab,apustuAnitza);
        if (b) {
            apustuaEginGS(erab, quote, balioa, apustuBikoitzaGalarazi);
        }
    }
}
public boolean apustuAnitzaErab(Jarraitzailea erab,ApustuAnitza apustuAnitza) {
    for (ApustuAnitza apu : erab.getNork().getApustuAnitzak()) {
        if (apu.getApustuKopia() == apustuAnitza.getApustuKopia()) {
            return false;
        }
    }
    return true;
}
public void apustuaEginGS(Jarraitzailea erab, LinkedList<Quote> quote,
        Double balioa, Integer apustuBikoitzaGalarazi) {
    if (erab.getNork().getDiruLimitea() < balioa) {
        this.ApustuaEgin(erab.getNork(), quote, erab.getNork().getDiruLimitea(), apustuBikoitzaGalarazi);
    } else {
        this.ApustuaEgin(erab.getNork(), quote, balioa, apustuBikoitzaGalarazi);
    }
}
```

Goiko adibide berdina da baina aplikatzen da adibide honetan, beste metodo bat gehitu dut kodea modulu gehiagotan banatzeko eta argiagoa izateko,

Mikel Mugica

Duplicate code

Hasierako kodea:

```
q7 = ev21.addQuestion("Emaitza?", 1);
q8 = ev21.addQuestion("Emaitza?", 1);
q9 = ev22.addQuestion("Irabazlea?", 1.5);
q10 = ev26.addQuestion("Irabazlea?", 1.5);
```

Errefaktorizatuko kodea:

```
String emaitza = "Emaitza?";
String irabazlea = "Irabazlea?";

q7 = ev21.addQuestion(emaitza, 1);
q8 = ev21.addQuestion(emaitza, 1);
q9 = ev22.addQuestion(irabazlea, 1.5);
q10 = ev26.addQuestion(irabazlea, 1.5);
```

String motako aldagaiak definitu ditut bi String pare berak ez idazteko.

Mikel Mugica

Write simple units of code

Hasierako kodea:

```
public void ApustuIrabazi(ApustuAnitza apustua) {
    ApustuAnitza apustuAnitza = db.find(ApustuAnitza.class, apustua.getApustuAnitzaNumber());
    Registered reg = (Registered) apustuAnitza.getUser();
    Registered r = (Registered) db.find(Registered.class, reg.getUsername());
    db.getTransaction().begin();
    apustuAnitza.setEgoera("irabazita");
    Double d = apustuAnitza.getBalioa();
    for (Apustua ap : apustuAnitza.getApustuak()) {
        d = d * ap.getKuota().getQuote();
    }
    r.updateDiruKontua(d);
    r.setIrabazitakoa(r.getIrabazitakoa() + d);
    r.setZenbat(r.getZenbat() + 1);
    Transaction t = new Transaction(r, d, new Date(), "ApustuIrabazi");
    db.persist(t);
    db.getTransaction().commit();
}
```

Errefaktorizatuko kodea:

```
public void ApustuIrabazi(ApustuAnitza apustua) {
    ApustuAnitza apustuAnitza = db.find(ApustuAnitza.class, apustua.getApustuAnitzaNumber());
    Registered reg = (Registered) apustuAnitza.getUser();
    Registered r = (Registered) db.find(Registered.class, reg.getUsername());
    apustuIrabaziGS(r,apustuAnitza);
}

public void apustuIrabaziGS(Registered r, ApustuAnitza apustuAnitza) {
    db.getTransaction().begin();
    apustuAnitza.setEgoera("irabazita");
    Double d = apustuAnitza.getBalioa();
    for (Apustua ap : apustuAnitza.getApustuak()) {
        d = d * ap.getKuota().getQuote();
    }
    r.updateDiruKontua(d);
    r.setIrabazitakoa(r.getIrabazitakoa() + d);
    r.setZenbat(r.getZenbat() + 1);
    Transaction t = new Transaction(r, d, new Date(), "ApustuIrabazi");
    db.persist(t);
    db.getTransaction().commit();
}
```

apustuIrabazi metodoa luzea denez beste metodo batean banatu dut, metodo erraza izango da, argiago eta ulergarriago. Aldaketak egiteko ere errazagoa izango da.

Mikel Mugica