# Errefaktorizazioak

Asier Aizpurua
Juan Alagon
Xabier Artola

# Egin beharrekoa

- **"Write short units of code" (2. kapituloa)**
  Guideline:
  - Limit the length of code units to 15 lines of code.
  - Do this by not writing units that are longer than 15 lines of code in the first place, or by splitting long units into multiple smaller units until each unit has at most 15 lines of code.
  - This improves maintainability because small units easy to understand, easy to test, and easy to reuse.
- **"Write simple units of code" (3. kapituloa)**
  Guideline:
  - Limit the number of branch points per unit to 4.
  - Do this by splitting complex units into simpler ones and avoiding complex units altogether.
  - This improves maintainability because keeping the number of branch points low makes units easier to modify and test.
- **“Duplicate code” (4. kapituloa).** SonarLint-ek errepikatutako kodea non dagoen aurkezten digu.
  Guideline:
  - Do not copy code.
  - Do this by writing reusable, generic code and/or calling existing methods instead.
  - This improves maintainability because when code is copied, bugs need to be fixed at multiple places, which is inefficient and error-prone.
- **"Keep unit interfaces small" (5. kapituloa).**
  Guideline:
  - Limit the number of parameters per unit to at most 4.
  - Do this by extracting parameters into objects.
  - This improves maintainability because keeping the number of parameters low makes units easier to understand and reuse.

# Asier Aizpurua

## Write short units of code

### Hasierako kodea

DataAccess.java

```
public Mezua mezuaBidali(Pertsona m, Pertsona nori, String mezua) throws MezuaEzDaZuzena {
    db.getTransaction().begin();
    Mezua mez = null;
    Pertsona mezulariDB = db.find(Pertsona.class, m.getIzena());
    Pertsona noriDB = db.find(Pertsona.class, nori.getIzena());
    Boolean zuzenaMIN = Mezua.mezuaZuzenaDaMIN(mezua);
    Boolean zuzenaMAX = Mezua.mezuaZuzenaDaMAX(mezua);
    if (zuzenaMIN && zuzenaMAX) {
        if (mezulariDB instanceof Erabiltzailea && ((Erabiltzailea) mezulariDB).getBlokeoa() != null && !((Erabiltzailea) mezulariDB).getBlokeoa().getNor().equals(noriDB)) {
            db.getTransaction().rollback();
            return null;
        }
        mez = new Mezua(mezulariDB, noriDB, mezua);
        mezulariDB.gehituBidaliLista(mez);
        noriDB.gehituJasotakoLista(mez);
        db.persist(mez);
    } else {
        if (!zuzenaMIN) {
            db.getTransaction().rollback();
            throw new MezuaEzDaZuzena("Short_message");
        } else if (!zuzenaMAX) {
            db.getTransaction().rollback();
            throw new MezuaEzDaZuzena("Long_message");
        }
    }
    db.getTransaction().commit();
    return mez;
}
```

### Errefaktorizatuko kodea

DataAccess.java

```
public Mezua mezuaBidali(Pertsona m, Pertsona nori, String mezua) throws MezuaEzDaZuzena {
    db.getTransaction().begin();
    Mezua mez = null;
    Pertsona mezulariDB = db.find(Pertsona.class, m.getIzena());
    Pertsona noriDB = db.find(Pertsona.class, nori.getIzena());
    Boolean zuzenaMIN = Mezua.mezuaZuzenaDaMIN(mezua);
    Boolean zuzenaMAX = Mezua.mezuaZuzenaDaMAX(mezua);
    if (zuzenaMIN && zuzenaMAX) {
        mez = mezuZuzenaBidali(mezua, mezulariDB, noriDB);
        if (mez == null) {
            return null;
        }
    } else {
        mezuEzZuzena(zuzenaMIN, zuzenaMAX);
    }
    db.getTransaction().commit();
    return mez;
}

private void mezuEzZuzena(Boolean zuzenaMIN, Boolean zuzenaMAX) throws MezuaEzDaZuzena {
    if (!zuzenaMIN) {
        db.getTransaction().rollback();
        throw new MezuaEzDaZuzena("Short_message");
    } else if (!zuzenaMAX) {
        db.getTransaction().rollback();
        throw new MezuaEzDaZuzena("Long_message");
    }
}

private Mezua mezuZuzenaBidali(String mezua, Pertsona mezulariDB, Pertsona noriDB) {
    Mezua mez;
    if (mezulariDB instanceof Erabiltzailea && ((Erabiltzailea) mezulariDB).getBlokeoa() != null && !((Erabiltzailea) mezulariDB).getBlokeoa().getNor().equals(noriDB)) {
        db.getTransaction().rollback();
        return null;
    }
    mez = new Mezua(mezulariDB, noriDB, mezua);
    mezulariDB.gehituBidaliLista(mez);
    noriDB.gehituJasotakoLista(mez);
    db.persist(mez);
    return mez;
}
```

| **Egindako errefaktorizazioren deskribapena** |
| --- |
| Metodoak hasieran 15 lerro baino gehiagoko tamaina zuen, honek metodoa ulertzea zailtzen zuen. Hau konpontzeko 2 metodo berri sortu ditugu, hasierako metodoaren funtzionamenduarekin bat datoztenak eta kolokan ipintzen ez dituztenak. |

Write simple units of code

**Hasierako kodea**

DataAccess.java

```
public Apustua apustuaEgin(Erabiltzailea er, Kuota ki, Double diruKop) throws ApustuaEzDaEgin {
    db.getTransaction().begin();
    String izena = er.getIzena();
    Erabiltzailea erDB = db.find(Erabiltzailea.class, izena);
    Kuota kDB = db.find(Kuota.class, ki.getKuotaZenbakia());
    if (erDB != null && erDB.getBlokeoa() == null) {
        Boolean nahikoa = erDB.diruaNahikoa(diruKop);
        Boolean minimoaGaindtu = diruKop >= kDB.getQuestion().getBetMinimum();
        if (kDB.galderaEmaitzaDu()) {
            throw new ApustuaEzDaEgin("galdera_emaitza_du");
        } else if (nahikoa && minimoaGaindtu) {
            erDB.saldoaAldatu((-1) * diruKop);
            Mugimendua mugi = erDB.mugimenduaSortu((-1) * diruKop, "apustua_eginda");
            db.persist(mugi);
            Apustua apustua = erDB.apustuaSortu(diruKop, kDB);
            db.persist(apustua);
            kDB.apustuaGehitu(apustua);
            db.getTransaction().commit();
            this.apustuaJarraitu(apustua);
            return apustua;
        } else {
            if (!nahikoa)
                throw new ApustuaEzDaEgin("NoMoney");
            else if (!minimoaGaindtu)
                throw new ApustuaEzDaEgin("errorea_minimoa_gainditu");
        }
        return null;
    } else {
        db.getTransaction().commit();
        return null;
    }
}
```

## Errefaktorizatuko kodea

DataAccess.java

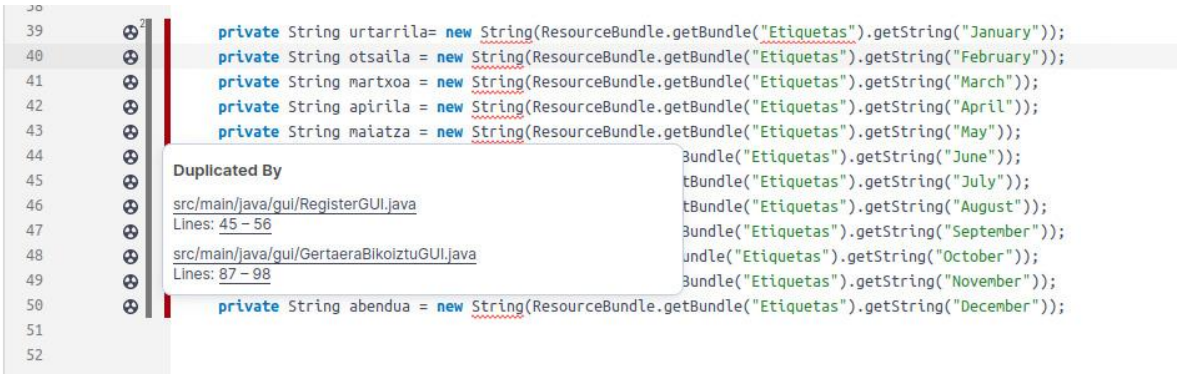
```java
public Apustua apustuaEgin(Erabiltzailea er, Kuota ki, Double diruKop) throws ApustuaEzDaEgin {
    db.getTransaction().begin();
    String izena = er.getIzena();
    Erabiltzailea erDB = db.find(Erabiltzailea.class, izena);
    Kuota kDB = db.find(Kuota.class, ki.getKuotaZenbakia());
    if (erDB != null && erDB.getBlokeoa() == null) {
        return apustuaBueltatu(diruKop, erDB, kDB);
    } else {
        db.getTransaction().commit();
        return null;
    }
}

private Apustua apustuaBueltatu(Double diruKop, Erabiltzailea erDB, Kuota kDB) throws ApustuaEzDaEgin {
    Boolean nahikoa = erDB.diruaNahikoa(diruKop);
    Boolean minimoaGaindtu = diruKop >= kDB.getQuestion().getBetMinimum();
    if (kDB.galderaEmaitzaDu()) {
        throw new ApustuaEzDaEgin("galdera_emaitza_du");
    } else if (nahikoa && minimoaGaindtu) {
        erDB.saldoaAldatu((-1) * diruKop);
        Mugimendua mugi = erDB.mugimenduaSortu((-1) * diruKop, "apustua_eginda");
        db.persist(mugi);
        Apustua apustua = erDB.apustuaSortu(diruKop, kDB);
        db.persist(apustua);
        kDB.apustuaGehitu(apustua);
        db.getTransaction().commit();
        this.apustuaJarraitu(apustua);
        return apustua;
    } else {
        if (!nahikoa)
            throw new ApustuaEzDaEgin("NoMoney");
        else if (!minimoaGaindtu)
            throw new ApustuaEzDaEgin("errorea_minimoa_gainditu");
    }
    return null;
}
```

## Egindako errefaktorizazioren deskribapena

Hasierako metodoan hainbat if beste baten barruan erabiltzen ziren eta metodoa oso korapilatsua bihurtzen zen. Hau zuzentzeko, lehenengo if-aren barruan dagoen kodea beste metodo batean sartu dugu.

# Duplicate code

## Sonarlint mezua




## Hasierako kodea

GertaeraSortuGUI.java

```
private DefaultComboBoxModel<Integer> eguna_model;
private JButton CreateNewButton;
private JLabel CreateNewEventLabel;

private String urtarrila= new String(ResourceBundle.getBundle("Etiquetas").getString("January"));
private String otsaila = new String(ResourceBundle.getBundle("Etiquetas").getString("February"));
private String martxoa = new String(ResourceBundle.getBundle("Etiquetas").getString("March"));
private String apirila = new String(ResourceBundle.getBundle("Etiquetas").getString("April"));
private String maiatza = new String(ResourceBundle.getBundle("Etiquetas").getString("May"));
private String ekaina = new String(ResourceBundle.getBundle("Etiquetas").getString("June"));
private String uztaila = new String(ResourceBundle.getBundle("Etiquetas").getString("July"));
private String abuztua = new String(ResourceBundle.getBundle("Etiquetas").getString("August"));
private String iraila = new String(ResourceBundle.getBundle("Etiquetas").getString("September"));
private String urria = new String(ResourceBundle.getBundle("Etiquetas").getString("October"));
private String azaroa = new String(ResourceBundle.getBundle("Etiquetas").getString("November"));
private String abendua = new String(ResourceBundle.getBundle("Etiquetas").getString("December"));


private static GertaeraSortuGUI frame;
```

RegisterGUI.java

```
private JLabel lblErrorea;

public static JFrame frame;

private String urtarrila= new String(ResourceBundle.getBundle("Etiquetas").getString("January"));
private String otsaila = new String(ResourceBundle.getBundle("Etiquetas").getString("February"));
private String martxoa = new String(ResourceBundle.getBundle("Etiquetas").getString("March"));
private String apirila = new String(ResourceBundle.getBundle("Etiquetas").getString("April"));
private String maiatza = new String(ResourceBundle.getBundle("Etiquetas").getString("May"));
private String ekaina = new String(ResourceBundle.getBundle("Etiquetas").getString("June"));
private String uztaila = new String(ResourceBundle.getBundle("Etiquetas").getString("July"));
private String abuztua = new String(ResourceBundle.getBundle("Etiquetas").getString("August"));
private String iraila = new String(ResourceBundle.getBundle("Etiquetas").getString("September"));
private String urria = new String(ResourceBundle.getBundle("Etiquetas").getString("October"));
private String azaroa = new String(ResourceBundle.getBundle("Etiquetas").getString("November"));
private String abendua = new String(ResourceBundle.getBundle("Etiquetas").getString("December"));
```

## Errefaktorizatuko kodea

GUIHilabetekoa.java

```
package gui;

import java.awt.GraphicsConfiguration;

public class GUIHilabetekoa extends GUI {

    protected String urtarrila = new String(ResourceBundle.getBundle("Etiquetas").getString("January"));
    protected String otsaila = new String(ResourceBundle.getBundle("Etiquetas").getString("February"));
    protected String martxoa = new String(ResourceBundle.getBundle("Etiquetas").getString("March"));
    protected String apirila = new String(ResourceBundle.getBundle("Etiquetas").getString("April"));
    protected String maiatza = new String(ResourceBundle.getBundle("Etiquetas").getString("May"));
    protected String ekaina = new String(ResourceBundle.getBundle("Etiquetas").getString("June"));
    protected String uztaila = new String(ResourceBundle.getBundle("Etiquetas").getString("July"));
    protected String abuztua = new String(ResourceBundle.getBundle("Etiquetas").getString("August"));
    protected String iraila = new String(ResourceBundle.getBundle("Etiquetas").getString("September"));
    protected String urria = new String(ResourceBundle.getBundle("Etiquetas").getString("October"));
    protected String azaroa = new String(ResourceBundle.getBundle("Etiquetas").getString("November"));
    protected String abendua = new String(ResourceBundle.getBundle("Etiquetas").getString("December"));

    public GUIHilabetekoa() throws HeadlessException {
        super();
    }

    public GUIHilabetekoa(GraphicsConfiguration gc) {
```

RegisterGUI.java

```
package gui;

import java.awt.BorderLayout;

public class RegisterGUI extends GUIHilabetekoa {

    private JPanel contentPane;
    private JPasswordField password;
    private JPasswordField confirm_passwordField;
    private JTextField user;
    private JComboBox urtea_comboBox;
    private DefaultComboBoxModel<Integer> urtea_model;
    private JComboBox hilabetea_comboBox;
    private DefaultComboBoxModel<String> hilabetea_model;
    private JComboBox eguna_comboBox;
    private DefaultComboBoxModel<Integer> eguna_model;
    private JLabel lblErrorea;

    public static JFrame frame;
```

GertaeraSortuGUI.java

```java
package gui;

import java.awt.BorderLayout;

public class GertaeraSortuGUI extends GUIHilabetekoa {

    private JPanel contentPane;
    private JTextField textFieldDescription;
    private JLabel DataLabel;
    private JLabel DescriptionLabel;
    private JComboBox YearBox;
    private JComboBox MonthBox;
    private JComboBox DayBox;
    private DefaultComboBoxModel<Integer> urtea_model;
    private DefaultComboBoxModel<String> hilabetea_model;
    private DefaultComboBoxModel<Integer> eguna_model;
    private JButton CreateNewButton;
    private JLabel CreateNewEventLabel;

    private static GertaeraSortuGUI frame;
```

**Egindako errefaktorizazioren deskribapena**

Bi klaseen atributuak ez errepikatzeko, goiko klase bat sortu dugu, eta atributuak bertan sartu.

# Keep unit interfaces small

| **Hasierako kodea** |
| --- |
| Ez ditut aurkitu |
| **Errefaktorizatuko kodea** |
| |
| **Egindako errefaktorizazioren deskribapena** |
| |

# Juan Alagon

## Write short units of code

**Hasierako kodea**

DataAccess.java

```
public boolean apustuaEzabatu(Apustua a, Erabiltzailea er) {
    Apustua aDB = db.find(Apustua.class, a.getApustuZenbakia());
    if (aDB != null) {
        Erabiltzailea erDB = aDB.getErabiltzailea();

        if (erDB.getIzena().equals(er.getIzena()) && aDB.ezabatuDaiteke() && erDB.getBlokeoa() == null) {
            db.getTransaction().begin();
            List<Kuota> kDBLista = aDB.getKuotak();
            Double diruKop = aDB.getDiruKop();
            erDB.saldoaAldatu(diruKop);
            Mugimendua m = erDB.mugimenduaSortu(diruKop, "apustua_ezabatuta");
            db.persist(m);
            erDB.apustuaEzabatuListatik(aDB);
            for (Kuota kiDB : kDBLista) {
                kiDB.apustuaEzabatuListatik(aDB);
            }
            db.remove(aDB);
            db.getTransaction().commit();
            return true;
        }
    }
    db.getTransaction().commit();
    return false;
}
```

## Errefaktorizatuko kodea

DataAccess.java

```java
public boolean apustuaEzabatu(Apustua a, Erabiltzailea er) {
    Apustua aDB = db.find(Apustua.class, a.getApustuZenbakia());
    if (aDB != null) {
        Erabiltzailea erDB = aDB.getErabiltzailea();

        if (erDB.getIzena().equals(er.getIzena()) && aDB.ezabatuDaiteke() && erDB.getBlokeoa() == null) {
            apustuaEzabatuDB(aDB, erDB);
            return true;
        }
    }
    return false;
}

private void apustuaEzabatuDB(Apustua aDB, Erabiltzailea erDB) {
    db.getTransaction().begin();
    List<Kuota> kDBLista = aDB.getKuotak();
    Double diruKop = aDB.getDiruKop();
    erDB.saldoaAldatu(diruKop);
    Mugimendua m = erDB.mugimenduaSortu(diruKop, "apustua_ezabatuta");
    db.persist(m);
    erDB.apustuaEzabatuListatik(aDB);
    for (Kuota kiDB : kDBLista) {
        kiDB.apustuaEzabatuListatik(aDB);
    }
    db.remove(aDB);
    db.getTransaction().commit();
}
```

## Egindako errefaktorizazioren deskribapena

Hasierako kodean apustua ezabatzeko metodoak 18 lerro zituen, gomendatzen den lerro kopurutik pasatzen zen. Konpontzeko datu-basetik apustua ezabatzen duen kode zatia metodo berri batera atera egin dut.

# Write simple units of code

## Hasierako kodea

DataAccess.java

```
public boolean erabiltzaileaJarraitu(Erabiltzailea unekoErab, Erabiltzailea aukeratutakoErabiltzailea,
        float diruMax) {
    if (unekoErab == null || aukeratutakoErabiltzailea == null || unekoErab.equals(aukeratutakoErabiltzailea)) {
        return false; // ezin duzu zure burua jarraitu
    }
    Erabiltzailea unErDB = db.find(Erabiltzailea.class, unekoErab.getIzena());
    Erabiltzailea erDB = db.find(Erabiltzailea.class, aukeratutakoErabiltzailea.getIzena());
    if (unErDB == null || erDB == null || unErDB.getBlokeoa() != null) {
        return false;
    } else {
        db.getTransaction().begin();
        Jarraitzen bJarraitu = unErDB.jarraitzenDu(erDB);
        if (bJarraitu != null) { // Jarraizten utzi
            unErDB.ezabatuJarraitzenListatik(bJarraitu);
            erDB.ezabatuJarraitzaileakListatik(unErDB);
        } else {
            Jarraitzen jB = unErDB.jarraitu(erDB, diruMax);
            erDB.gehituJarraitzaileakListara(unErDB);
            db.persist(jB);
        }
        db.getTransaction().commit();
    }

    return true;
}
```

## Errefaktorizatuko kodea

DataAccess.java

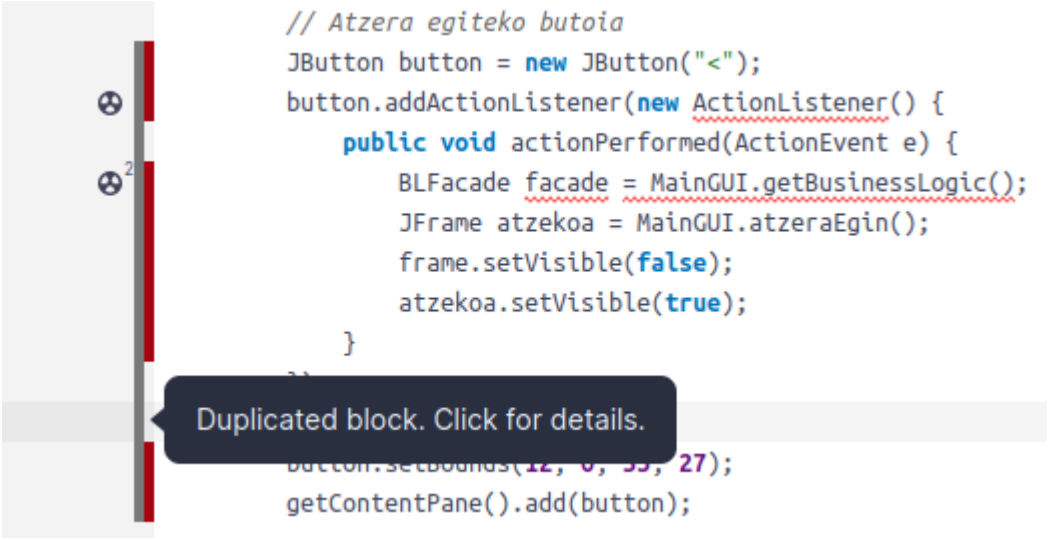
```
public boolean erabiltzaileaJarraitu(Erabiltzailea unekoErab, Erabiltzailea aukeratutakoErabiltzailea,
        float diruMax) {
    if (unekoErab == null || aukeratutakoErabiltzailea == null || unekoErab.equals(aukeratutakoErabiltzailea)) {
        return false; // ezin duzu zure burua jarraitu
    }
    Erabiltzailea unErDB = db.find(Erabiltzailea.class, unekoErab.getIzena());
    Erabiltzailea erDB = db.find(Erabiltzailea.class, aukeratutakoErabiltzailea.getIzena());
    if (unErDB == null || erDB == null || unErDB.getBlokeoa() != null) {
        return false;
    } else {
        erabiltzaileaJarraituDB(diruMax, unErDB, erDB);
    }

    return true;
}

private void erabiltzaileaJarraituDB(float diruMax, Erabiltzailea nor, Erabiltzailea nori) {
    db.getTransaction().begin();
    Jarraitzen bJarraitu = nor.jarraitzenDu(nori);
    if (bJarraitu != null) { // Jarraizten utzi
        nor.ezabatuJarraitzenListatik(bJarraitu);
        nori.ezabatuJarraitzaileakListatik(nor);
    } else {
        Jarraitzen jB = nor.jarraitu(nori, diruMax);
        nori.gehituJarraitzaileakListara(nor);
        db.persist(jB);
    }
    db.getTransaction().commit();
}
```

| **Egindako errefaktorizazioren deskribapena** |
| --- |
| Hasierako metodoak adar gehiegi zituen. Konpontzeko erabiltzailea jarraitzearen entitateak sortzen dituen kode lerroak beste metodo berri batera mugitu dut. Horrela adar kopurua bi metodoen artean banatu egin dut.<br>Orain goiko metodoan 3 adar ditugu eta beheko metodoan 2 adar. |

# Duplicate code

**Sonarlint mezua**

Interfaze guztietan atzera egiteko butoia sortzeko kodea errepikatuta dago. Sonarcloud-en abisatzen du bloke osoa errepikatuta dagoela



**Hasierako kodea**

JarraituGUI.java

```
87                  // Atzera egiteko butoia
88                  frame = this;
89                  button = new JButton("<");
90                  button.addActionListener(new ActionListener() {
91                          public void actionPerformed(ActionEvent e) {
92                                  BLFacade facade = MainGUI.getBusinessLogic();
93                                  JFrame atzekoa = MainGUI.atzeraEgin();
94                                  frame.setVisible(false);
95                                  atzekoa.setVisible(true);
96                          }
97                  });
98                  button.setBounds(12, 0, 41, 27);
99                  this.getContentPane().add(button);
```

ApustuakEginGUI.java

```
284                  frame = this;
285                  JButton button = new JButton("<");
286                  button.addActionListener(new ActionListener() {
287                          public void actionPerformed(ActionEvent e) {
288                                  JFrame atzekoa = MainGUI.atzeraEgin();
289                                  frame.setVisible(false);
290                                  atzekoa.setVisible(true);
291                          }
292                  });
293                  button.setBounds(21, 10, 41, 27);
294                  getContentPane().add(button);
```

ErabiltzaileGUI.java

```
89                  // Atzera egiteko butoia
90                  JButton button = new JButton("<");
91                  button.addActionListener(new ActionListener() {
92                          public void actionPerformed(ActionEvent e) {
93                                  JFrame atzekoa = MainGUI.atzeraEgin();
94                                  frame.setVisible(false);
95                                  atzekoa.setVisible(true);
96                          }
97                  });
98                  button.setBounds(12, 0, 41, 27);
99                  contentPane.add(button);
```

**Errefaktorizatuko kodea**

JarraituGUI.java

```
86          // Atzera egiteko butoia
87          frame = this;
88          atzeraButoiaSortu(frame);
```

ApustuakEginGUI.java

```
283          frame = this;
284          atzeraButoiaSortu(frame);
```

ErabiltzaileGUI.java

```
89          // Atzera egiteko butoia
90          atzeraButoiaSortu(frame);
```

GUI.java

```
package gui;

import java.awt.GraphicsConfiguration;

public class GUI extends JFrame {

    public GUI() throws HeadlessException {
        super();
    }

    public GUI(GraphicsConfiguration gc) {
        super(gc);
    }

    public GUI(String title) throws HeadlessException {
        super(title);
    }

    public GUI(String title, GraphicsConfiguration gc) {
        super(title, gc);
    }

    public void atzeraButoiaSortu(JFrame frame) {
        JButton button = new JButton("<");
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                JFrame atzekoa = MainGUI.atzeraEgin();
                frame.setVisible(false);
                atzekoa.setVisible(true);
            }
        });
        button.setBounds(21, 10, 41, 27);
        getContentPane().add(button);
    }

}
```

| **Egindako errefaktorizazioren deskribapena** |
| --- |
| GUI klaseetan atzera egiteko butoiaren kodea errepikatuta zegoen.<br><br>GUI guztiak JFrame klasearen umeak izanda “Pull up” ez zen posiblea. Errefaktorizazio hau gauzatzeko “Extract superclass…” errefaktorizazioa eginda GUI.java klasea sortu dut ( JFrame-ren umea dena ).<br><br>Orain GUI guztiak guraso berdina izanda, atzera egiteko butoia sortzen duen kodea “Extract method..” eginda eta gero metodo honeri “Pull up” eginda lortu dut metodoa GUI.java klase gurasoan edukitzea.<br><br>GUI guztietan errepikatzen denez eta eclipse ez dirudi detektatzen duelarik errepikatutako kodearen errefaktorizazioa egiteko, eskuz ordezkatu behar izan dut kode errepikatuta metodoarekin. |

# Keep unit interfaces small

| **Hasierako kodea** |
| --- |
| Ez ditut aurkitu |
| **Errefaktorizatuko kodea** |
| |
| **Egindako errefaktorizazioren deskribapena** |
| |

# Xabier Artola

## Write short units of code

**Hasierako kodea** (diruaSartu)

```
public boolean diruaSartu(Erabiltzailea erabiltzaile, String pasahitza, Double kantitatea) {
    Erabiltzailea e = db.find(Erabiltzailea.class, erabiltzaile.getIzena());
    if (e == null)
        return false;
    else if (!e.pasahitzaZuzena(pasahitza))
        return false;
    else {
        if (e.getBlokeoa() == null) {
            db.getTransaction().begin();
            e.saldoaAldatu(kantitatea);
            Mugimendua m = new Mugimendua(erabiltzaile, kantitatea, "dirua_sartu");
            e.mugimenduaGehitu(m);
            db.persist(m);
            db.getTransaction().commit();
            return true;
        } else
            return false;
    }
}
```

**Errefaktorizatuko kodea**

```
public boolean diruaSartu(Erabiltzailea erabiltzaile, String pasahitza, Double kantitatea) {
    Erabiltzailea e = db.find(Erabiltzailea.class, erabiltzaile.getIzena());
    if (e == null)
        return false;
    else if (!e.pasahitzaZuzena(pasahitza))
        return false;
    else {
        if (e.getBlokeoa() == null) {
            diruaSartuDB(erabiltzaile, kantitatea, e);
            return true;
        } else
            return false;
    }
}
```

sortutako metodo berria

```
private void diruaSartuDB(Erabiltzailea erabiltzaile, Double kantitatea, Erabiltzailea e) {
    db.getTransaction().begin();
    e.saldoaAldatu(kantitatea);
    Mugimendua m = new Mugimendua(erabiltzaile, kantitatea, "dirua_sartu");
    e.mugimenduaGehitu(m);
    db.persist(m);
    db.getTransaction().commit();
}
```

| **Egindako errefaktorizazioren deskribapena** |
| --- |
| Kodeak 16 lerro ditu eta gomendagarria denez 15 izatea gehienez, aprobetxatu dut beste metodo txikiago bat sortu eta horrela bi metodo motz izateko. Metodo berriari diruaSartuDB izena eman diot |

# Write simple units of code

**Hasierako kodea** (erabiltzaileaBlokeatu)

```
public Blokeoa erabiltzaileaBlokeatu(Admin a, Erabiltzailea ei, String arrazoia) throws MezuaEzDaZuzena {
        db.getTransaction().begin();
        Blokeoa bl = null;
        Admin aDB = db.find(Admin.class, a.getIzena());
        Boolean zuzenaMIN = false;
        Boolean zuzenaMAX = false;
        Erabiltzailea eDB = db.find(Erabiltzailea.class, ei.getIzena());
        if (arrazoia == null) {
                zuzenaMIN = true;
                zuzenaMAX = true;
        } else {
                zuzenaMIN = Mezua.mezuaZuzenaDaMIN(arrazoia);
                zuzenaMAX = Mezua.mezuaZuzenaDaMAX(arrazoia);
        }
        if (zuzenaMIN && zuzenaMAX) {
                if (eDB.getBlokeoa() == null) {
                        bl = new Blokeoa(aDB, eDB, arrazoia);
                        aDB.blokeoaGehituListan(bl);
                        eDB.blokeoaGehitu(bl);
                        db.persist(bl);
                } else {
                        bl = eDB.getBlokeoa();
                        aDB.blokeoaEzabatuListatik(bl);
                        eDB.blokeoaEzabatu();
                        db.remove(bl);
                }
        } else {
                if (!zuzenaMIN) {
                        throw new MezuaEzDaZuzena("Short_reason");
                } else if (!zuzenaMAX) {
                        throw new MezuaEzDaZuzena("Long_reason");
                }
        }
        db.getTransaction().commit();
        return bl;
}
```

## Errefaktorizatuko kodea

```
public Blokeoa erabiltzaileaBlokeatu(Admin a, Erabiltzailea ei, String arrazoia) throws MezuaEzDaZuzena {
    db.getTransaction().begin();
    Blokeoa bl = null;
    Admin aDB = db.find(Admin.class, a.getIzena());
    Boolean zuzenaMIN = false;
    Boolean zuzenaMAX = false;
    Erabiltzailea eDB = db.find(Erabiltzailea.class, ei.getIzena());
    if (arrazoia == null) {
        zuzenaMIN = true;
        zuzenaMAX = true;
    } else {
        zuzenaMIN = Mezua.mezuaZuzenaDaMIN(arrazoia);
        zuzenaMAX = Mezua.mezuaZuzenaDaMAX(arrazoia);
    }
    if (zuzenaMIN && zuzenaMAX) {
        bl = erbiltzaileaBlokeatuDB(arrazoia, aDB, eDB);
    } else {
        mezuLuzeraDesegokiaEszepzioa(zuzenaMIN, zuzenaMAX);
    }
    db.getTransaction().commit();
    return bl;
}
```

erabiltzaileaBlokeatuDB

```
private Blokeoa erbiltzaileaBlokeatuDB(String arrazoia, Admin aDB, Erabiltzailea eDB) {
    Blokeoa bl;
    if (eDB.getBlokeoa() == null) {
        bl = new Blokeoa(aDB, eDB, arrazoia);
        aDB.blokeoaGehituListan(bl);
        eDB.blokeoaGehitu(bl);
        db.persist(bl);
    } else {
        bl = eDB.getBlokeoa();
        aDB.blokeoaEzabatuListatik(bl);
        eDB.blokeoaEzabatu();
        db.remove(bl);
    }
    return bl;
}
```

mezuaDesegokiaEszepzioa

```
private void mezuLuzeraDesegokiaEszepzioa(Boolean zuzenaMIN, Boolean zuzenaMAX) throws MezuaEzDaZuzena {
    if (!zuzenaMIN) {
        throw new MezuaEzDaZuzena("Short_reason");
    } else if (!zuzenaMAX) {
        throw new MezuaEzDaZuzena("Long_reason");
    }
}
```

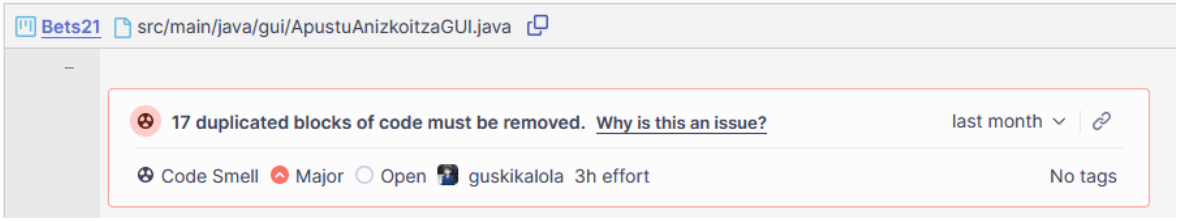
## Egindako errefaktorizazioren deskribapena

Irteerako hainbat balio desberdin eta kasu desberdin dituen metodo honek, 4 adar baina gehiago zitue. Horregatik metodo txikiagotan zatitu dut, bi metodo berri sortuz, erabiltzaileaBlokeatuDB eta mezuaDesegokiaEszepzioa izenekin.

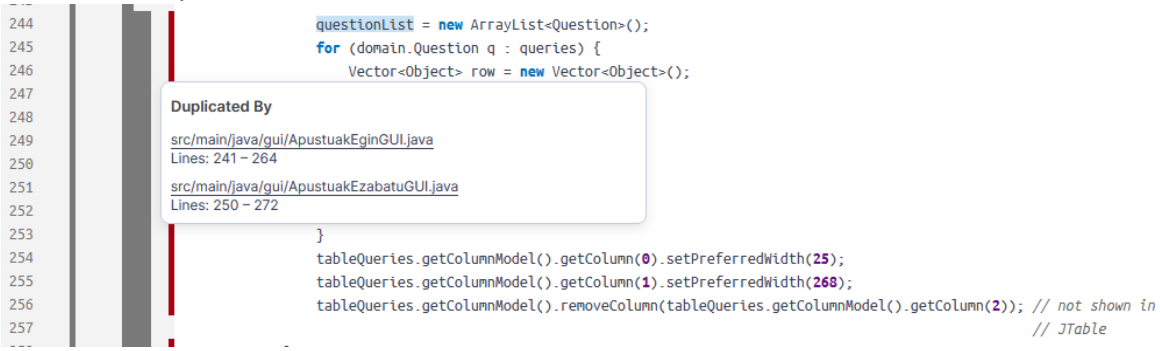
# Duplicate code

**Sonarlint mezua**

mezua




kodean kokapena



**Hasierako kodea** (ApustuAnizkoitzaGUI.java)

ApustuAnizkoitzaGUI.java(244-257 kode lerroak)

```
242                         + ev.getDescription());
243
244                 questionList = new ArrayList<Question>();
245                 for (domain.Question q : queries) {
246                     Vector<Object> row = new Vector<Object>();
247
248                     row.add(q.getQuestionNumber());
249                     row.add(q.getQuestion());
250                     row.add(q);
251                     tableModelQueries.addRow(row);
252                     questionList.add(q);
253                 }
254                 tableQueries.getColumnModel().getColumn(0).setPreferredWidth(25);
255                 tableQueries.getColumnModel().getColumn(1).setPreferredWidth(268);
256                 tableQueries.getColumnModel().removeColumn(tableQueries.getColumnModel().getColumn(2)); // not shown in
257                                                                                                         // JTable
258             }
```

ApustuakEginGU.javaI(241-253 kode lerroak)

```
236                         ResourceBundle.getBundle(etiketak).getString("NoQueries") + ": " + ev.getDescription());
237                 else
238                     jLabelQueries.setText(ResourceBundle.getBundle(etiketak).getString("SelectedEvent") + " "
239                             + ev.getDescription());
240
241                 questionList = new ArrayList<Question>();
242                 for (domain.Question q : queries) {
243                     Vector<Object> row = new Vector<Object>();
244
245                     row.add(q.getQuestionNumber());
246                     row.add(q.getQuestion());
247                     row.add(q);
248                     tableModelQueries.addRow(row);
249                     questionList.add(q);
250                 }
251                 tableQueries.getColumnModel().getColumn(0).setPreferredWidth(25);
252                 tableQueries.getColumnModel().getColumn(1).setPreferredWidth(268);
253                 tableQueries.getColumnModel().removeColumn(tableQueries.getColumnModel().getColumn(2)); // not shown in
254                                                                                                         // JTable
255             }
```

ApustuakEzabatuGUI.java(250 - 262 kode lerroak)

```
247
248
249
250                 questionList = new ArrayList<Question>();
251                 for (domain.Question q:queries){
252                     Vector<Object> row = new Vector<Object>();
253
254                     row.add(q.getQuestionNumber());
255                     row.add(q.getQuestion());
256                     row.add(q);
257                     tableModelQueries.addRow(row);
258                     questionList.add(q);
259                 }
260                 tableQueries.getColumnModel().getColumn(0).setPreferredWidth(25);
261                 tableQueries.getColumnModel().getColumn(1).setPreferredWidth(268);
262                 tableQueries.getColumnModel().removeColumn(tableQueries.getColumnModel().getColumn(2)); // not shown in JTable
263             }
```

## Errefaktorizatuko kodea

GUI.java guraso klasean sartutako metodo berriaren (gehituGaldera) kodea

```
public void gehituGaldera(Vector<Question> queries, DefaultTableModel defaultTableModel, ArrayList<Question> arrayList, JTable table) {
    for (domain.Question q : queries) {
        Vector<Object> row = new Vector<Object>();

        row.add(q.getQuestionNumber());
        row.add(q.getQuestion());
        row.add(q);
        defaultTableModel.addRow(row);
        arrayList.add(q);
    }
    table.getColumnModel().getColumn(0).setPreferredWidth(25);
    table.getColumnModel().getColumn(1).setPreferredWidth(268);
    table.getColumnModel().removeColumn(table.getColumnModel().getColumn(2)); // not shown in
}
```

ApustuAnizkoitzaGU.java klasean egindako aldaketa (245 kode lerroa)

```
226
227         tableEvents.addMouseListener(new MouseAdapter() {
228             @Override
229             public void mouseClicked(MouseEvent e) {
230                 int i = tableEvents.getSelectedRow();
231                 domain.Event ev = (domain.Event) tableModelEvents.getValueAt(i, 2); // obtain ev object
232                 Vector<Question> queries = ev.getQuestions();
233
234                 tableModelQueries.setDataVector(null, columnNamesQueries);
235                 tableModelKuotak.setDataVector(null, columnNamesKuota);
236                 tableModelQueries.setColumnCount(3);
237                 if (queries.isEmpty())
238                     jLabelQueries.setText(
239                             ResourceBundle.getBundle("Etiquetas").getString("NoQueries") + ": " + ev.getDescription());
240                 else
241                     jLabelQueries.setText(ResourceBundle.getBundle("Etiquetas").getString("SelectedEvent") + " "
242                             + ev.getDescription());
243
244                 questionList = new ArrayList<Question>();
245                 gehituGaldera(queries, tableModelQueries, questionList, tableQueries);
246
247             }
248         });
249
```

ApustuakEginGUI.java klasean egindako aldaketa (241 kode lerroa)

```
216         });
217
218         this.getContentPane().add(jCalendar1, null);
219
220         scrollPaneEvents.setBounds(new Rectangle(292, 50, 346, 150));
221         scrollPaneQueries.setBounds(new Rectangle(24, 234, 259, 116));
222         scrollPaneKuota.setBounds(new Rectangle(312, 230, 308, 123));
223
224         tableEvents.addMouseListener(new MouseAdapter() {
225             @Override
226             public void mouseClicked(MouseEvent e) {
227                 int i = tableEvents.getSelectedRow();
228                 domain.Event ev = (domain.Event) tableModelEvents.getValueAt(i, 2); // obtain ev object
229                 Vector<Question> queries = ev.getQuestions();
230
231                 tableModelQueries.setDataVector(null, columnNamesQueries);
232                 tableModelQueries.setColumnCount(3);
233                 if (queries.isEmpty())
234                     jLabelQueries.setText(
235                             ResourceBundle.getBundle(etiketak).getString("NoQueries") + ": " + ev.getDescription());
236                 else
237                     jLabelQueries.setText(ResourceBundle.getBundle(etiketak).getString("SelectedEvent") + " "
238                             + ev.getDescription());
239
240                 questionList = new ArrayList<Question>();
241                 gehituGaldera(queries, tableModelQueries, questionList, tableQueries);
242             }
243         });
244
245         tableQueries.addMouseListener(new MouseAdapter() {
```

ApustuakEzabatuGUI.java klasean egindako aldaketa (251 kode lerroa)

```
232         scrollPaneApustua.setBounds(new Rectangle(312, 230, 308, 123));
233
234         tableEvents.addMouseListener(new MouseAdapter() {
235             @Override
236             public void mouseClicked(MouseEvent e) {
237                 int i=tableEvents.getSelectedRow();
238                 domain.Event ev=(domain.Event)tableModelEvents.getValueAt(i,2); // obtain ev object
239                 Vector<Question> queries=ev.getQuestions();
240
241                 tableModelQueries.setDataVector(null, columnNamesQueries);
242                 tableModelQueries.setColumnCount(3);
243                 if (queries.isEmpty())
244                     jLabelQueries.setText(ResourceBundle.getBundle("Etiquetas").getString("NoQueries")+": "+ev.getDescription());
245                 else
246                     jLabelQueries.setText(ResourceBundle.getBundle("Etiquetas").getString("SelectedEvent")+" "+ev.getDescription());
247
248
249
250                 questionList = new ArrayList<Question>();
251                 gehituGaldera(queries, tableModelQueries, questionList, tableQueries);
252     }
```

| **Egindako errefaktorizazioren deskribapena** |
| --- |
| Errepikatutako kode zatia metodo bihurtu refactor → extract method bidez. Metodo berriari gehituGaldera galdera izena eman diot.<br><br>Ondoren metodoa birmoldatu beharrezko parametro guztiak onartzeko refactor →Introduce Parameter bidez.<br><br>Azkenik refactor → pull up bidez metodoa GUI.java guraso klasean kokatu eta eskuz metodoari deia jarri kode blokearen tokian aipatutako hiru klaseetan. |

## Keep unit interfaces small

| **Hasierako kodea** |
| --- |
| ez ditut aurkitu |
| **Errefaktorizatuko kodea** |
| |
| **Egindako errefaktorizazioren deskribapena** |
| |