



Software Ingeniaritza

2015-ko Ekainak 12

1. DISEINUA (2 puntu)

EHUK graduako ikasleei matrikula egiteko orduan lagunduko dien programa bat diseinatzea eskatu digu. Horretarako graduan eskaintzen diren ikasgai guztien katalogo bat eman digu, bakoitzarentzat bere identifikadorea, kreditu kopurua eta aurrebaldintza diren ikasgaiak dituelarik (ikasgai horretan matrikulatzeko ikaslea gaingaita izan beharreko ikasgaiak). Honetaz gain, beste katalogo batetan gradua egiten ari diren ikasleen informazioa eman digute. Bakoitzarentzat bere espediente zenbakia eta gradua hasi zuenetik egindako azterketa zerrenda ditu. Azterketa bakoitzak ikasgaiaren izena eta lortutako nota dituenek, ikasle bakoitzak ikasgai berdinentzat azterketa bat baino gehiago izatea posible da (lehenengoan gaingaitza lortu ez badu).

Eskatzen da:

- EHUK emandako informazioa gorde ahal izateko beharrezko klaseak identifikatu, bere atributuak (motarekin) adieraziz. Ariketaren **b)** atala egin ahal izateko beharrezkoak diren metodoak gehitu.
- matrikulaPosiblea** metodoaren sekuentzia diagrama egin. Metodo honek ikasgai baten izena eta ikasle baten espediente zenbakia emanda, boolear bat bueltatuko du ikasle horrek ikasgai matrikula egin dezakeen edo ez adieraziz.

2. PATROIAK (2 puntu)

Joko pertsonalizatua (1 puntu):

Psikologiako Fakultateko ikerlari talde batek populazioaren atentzio zuzendua neurtzen duen joko bat garatzeko laguntza eskatu digu. Aplikazioak honako funtzionamendua izatea eskatzen digute:

- Jokoa hasi baino lehen erabiltzaileari bere adina eta ama hizkuntza eskatuko zaizkio. Onartutako hizkuntzak momentuz euskara eta gaztelania izango dira, baina etorkizunean hizkuntza gehiagotara zabaltzea espero da.
- Ondoren, bere ezaugarrien arabera joko espezifikoak sortuko da. 18 urtetik beherako erabiltzaileek piktogrametan oinarritutako joko batetara jolastuko dute, helduek estrategia joko batetara jolasten duten bitartean. Bi jokoek funtzionamendu desberdina izango dute.
- Jokoa bukaeran, adingabekoek zorientzen dituen animazio bat ikusiko dute, helduek aurretik aukeratutako hizkuntzan galdetegi txiki bat erantzuten duten bitartean.

Eskatzen da:

- Zein patroi erabiliko zen(it)u(z)ke esandako softwarea garatzeko? Erantzuna arrazoitu.
- UML klase diagrama zehaztu klase nagusiekin eta garrantzitsuak iruditzen zaizkizun metodoekin.
- Aurreko klaseen implementazioa idatzi.

Fabrika automatizatua (1 puntu):

Fabrika automatizatu baten softwarea garatzea eskatzen digute. Espero den funtzionamendua honakoa da:

- Martxan jartzen denean softwareak argiak piztuko ditu, hiru produkzio lerroak aktibatuko ditu, biltegitik beharrezko materiala aterako du eta, fabrikazio plantak tenperatura asko igoarazten duenez, bulegoen aire egokitua piztuko du.



- Matxura elektriko bat egonez gero, sistemak argi guztiak itzaliko ditu eta larrialdikoak piztuko ditu, aire egokitua eta biltegi modulua itzaliko ditu eta hiru lerroak larrialdi moduan jarriko ditu (hau da, larrialdietarako sorgailu batek emango die energia).
- Produkzioa gelditu behar izanez gero, sistemak lerroak geldituko ditu eta biltegia, argiak eta aire egokitua itzaliko ditu.

Eskatzen da:

- a) Zein patroil erabiliko zen(it)u(z)ke esandako softwarea garatzeko? Erantzuna arrazoitu.
- b) UML klase diagrama zehaztu klase nagusiekin eta garrantzitsuak iruditzen zaizkizun metodoekin.
- c) Aurreko klaseen implementazioa idatzi.

OHARRA: Metodoen inplementazioan nahikoa da metodoak egin beharko lukeena pantailaratzeara. Bigarren ariketan kontutan izan fabrikaren zati bakoitzaren softwarea etorkizunean aldatu daitekeela.

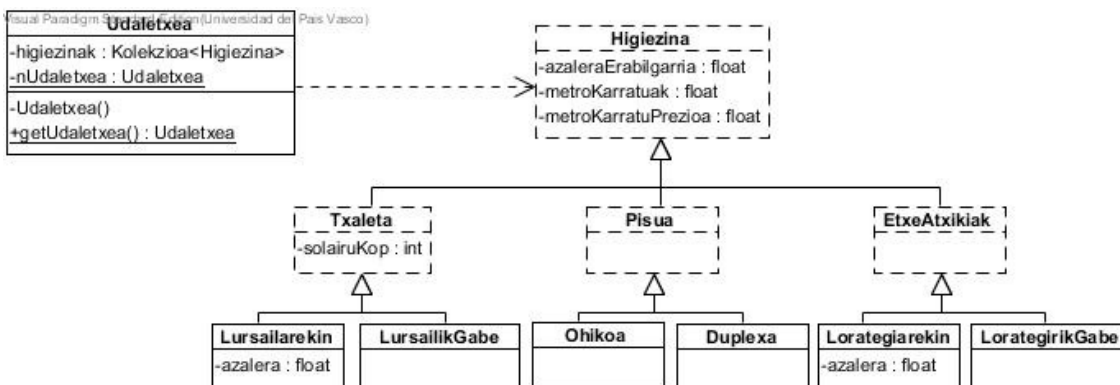
3. HERENTZIA (1 puntu)

Hiriaren udaletxeak *Ondasun Higiezin*en gaineko *Zerga (OHZ)* kobratzeko fakturen sorrera automatizatzen duen aplikazio bat dauka. Aplikazio honek eraikin desberdinak definitzen ditu, **I. Irudian** ikusi daitekeen bezala. Higiezin bakoitzak azalera erabilgarria izango du (m^2), m^2 prezioa eta eraikinaren balioa izango ditu. Datu hauen arabera aplikazioak OHZ-ren balioa kalkulatu du.

Krisiari aurre egiteko, aurten udaletxean higiezin “bereziei” zenbateko osagarri bat kobratzea erabaki du. Higiezin “bereziei” bezala txaletak (lursailarekin edo gabe) eta lorategia duten atxikitako etxeak klasifikatu dira.

Eskatzen da:

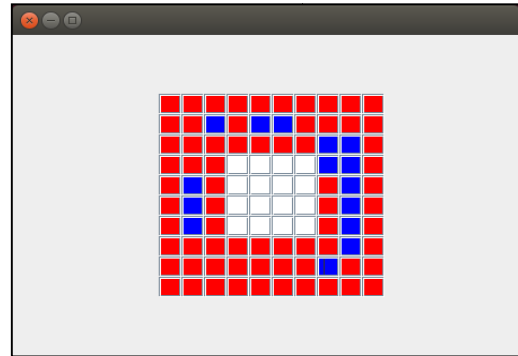
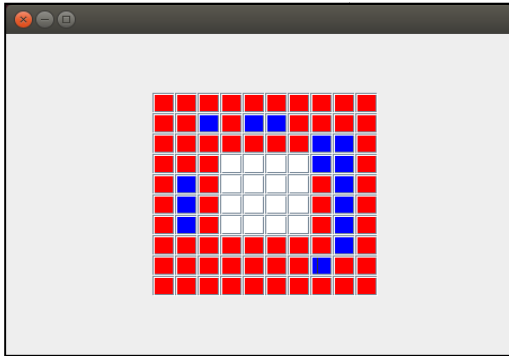
- a) Epe laburrekoa izan daitekeen erabaki hau kontutan hartzeko hierarkia osatu, eraikin berezien OHZ-ari gehitu beharreko zenbatekoa kalkulatzeko duen **osagarriaKalkulatu** metodoa non inplementatuko litzatekeen adieraziz..
- b) Higiezin “bereziei” esker lortutako zenbateko totala kalkulatzeko duen **osagarrienTotala** metodoa gehitu eta inplementatu.



I. Irudia: Higiezinen Hierarkia

4. INTERFAZE GRAFIKOAK (1 puntu)

Kirol estadio bateko eserlekuen erreserba egiten duen sistema inplementatu nahi dugu. **2. Irudiak** sistema honen leiho posible bat erakusten du. Bertan zelaia lauki zuriekin irudikatzen da eta grada koloreztatutako laukiekin. Lauki ilunek erreserbatutako eserlekuak adierazten dituzte eta argiagoek eserleku libreak. Erreserba bat egiteko erabiltzaileak edozein lauki libretan klik egin dezake eta ezeztatzeko lauki erreserbatu batetan klik egingo du. Edozein unetan erabiltzaile desberdinak egon daitezke aplikazio bera exekutatzen, beraz, garrantzitsua da denek erreserben egoera berdina ikustea. Hau da, leiho desberdinek estadioaren egoera berdina erakutsi beharko dute.



2. Irudia: Bi erabiltzaile desberdinek eserleku esleipen berdina “ikusten” dute

Honako kodea sistema honen inplementazioaren zati bat da.

Lauki batetan klik egitean bista ereduarekin komunikatu behar da biek duten informazioa berdina dela ziurtatzeko eta hala beste edozein leihok eserleku esleipen bera duela.

Eskatzen da:

Erabiltzaile batek erreserba bat egin edo ezeztatzen duenean beste erabiltzaileen leihoak egoki eguneratzeko beharrezkoa den kodea gehitu.

```
public class Estadioa //OSATU {
    private static final int SIZE = 10;
    private static final int ALDATUEZIN = -1;
    private static final int LIBRE = 0;
    private static final int OKUPATUTA = 1;
    private int[][] matizea;
    private static Estadioa eredia = new Estadioa();
    private Estadioa() { //Eraikitzailea. Estadioa hasieratzen du eserleku libreak sortuz eta zelaia aldatu ezin bezala markatuz }
    public static Estadioa getInstance(){return eredia;}
    public void change(int row, int col){
        // Balioa alderantzizkora aldatzen du baldin eta aldagarria bada
        if (matizea[row][col] != ALDATUEZIN)
            if (matizea[row][col] == LIBRE) matizea[row][col] = OKUPATUTA;
            else matizea[row][col] = LIBRE;
    }
    //OSATU
}

public int[][] getMat(){return matizea;}

public class EstadioBista extends JFrame //OSATU {
    private static final int SIZE = 10;
    private static final int ALDATUEZIN = -1;
```



```
private static final int LIBRE = 0;
private static final int OKUPATUTA = 1;
private JPanel contentPane;
private ArrayList<JTextField> textFields = new ArrayList<JTextField>();
private Controller controller = new Controller();
public EstadioBista() {
    ...
    //Leihoaren edukiontzi eta osagaien sorrera. Ez ikutu.
    JPanel gridPanel = new JPanel(new GridLayout(SIZE, SIZE));
    for (int row = 0; row < SIZE; row++) {
        for (int col = 0; col < SIZE; col++) {
            JTextField newTextField = new JTextField(" ");
            gridPanel.add(newTextField);
            textFields.add(newTextField);
            newTextField.addActionListener(controller);
        }
    }
    this.actualizarMatriz();
    setLayout(new GridBagLayout()); // to center component added
    this.add(gridPanel);
}

private class Controller //OSATU {

    public void actionPerformed(ActionEvent arg0) {
        JTextField textField = (JTextField) arg0.getSource();
        int row = lerroaBilatu(textField); int col = zutabeaBilatu(textField);
        //OSATU
    }
}

private void matizeaEguneratu() { // Bista ereduaren elementuekin eguneratzen du.
    int[][] mat = Estadioa.getInstance().getMat();
    for (int row = 0; row < Estadioa.getInstance().getSize(); row++) {
        for (int col = 0; col < Estadioa.getInstance().getSize(); col++) {
            int pos = row * Estadioa.getInstance().getSize() + col;
            if (mat[row][col] == Estadioa.getInstance().getLIBRE())
                textFields.get(pos).setBackground(Color.red);
            else if (mat[row][col] == Estadioa.getInstance().getOKUPATUTA())
                textFields.get(pos).setBackground(Color.blue);
            else textFields.get(pos).setBackground(Color.white); //ALDATUEZINA
        }
    }
}

//OSATU

}

public class Azterketa2DeialdiaMain {
    public static void main(String[] args) {
        EstadioBista b = new EstadioBista ();
        EstadioBista b2 = new EstadioBista();
        Estadioa m = Estadioa.getInstance();
        //OSATU

        b.setVisible(true);
        b2.setVisible(true);
    }
}
```