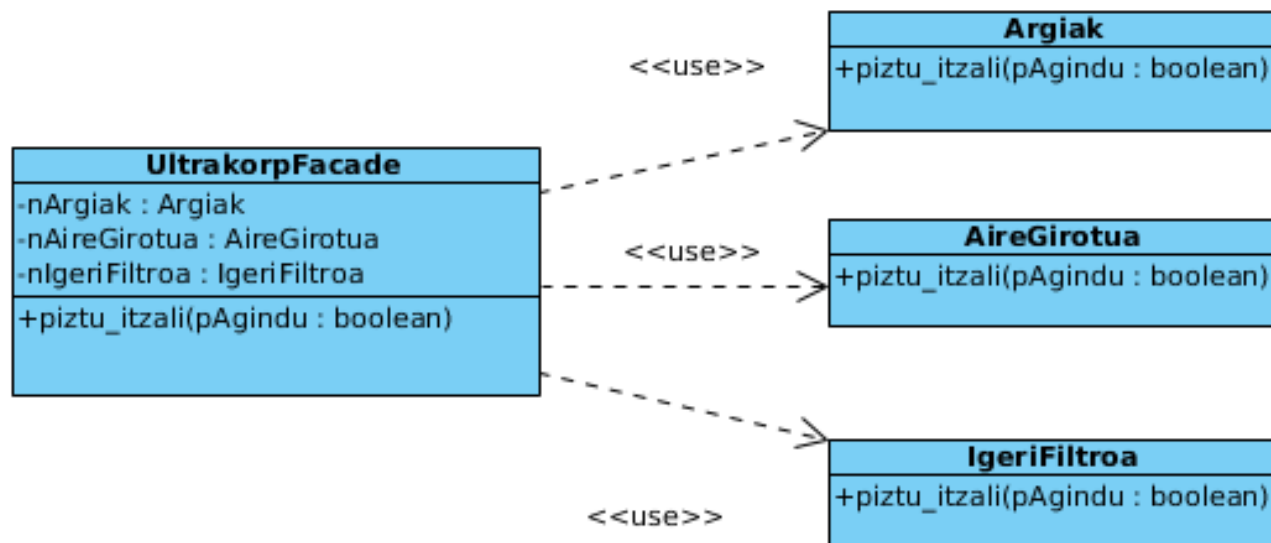


## ERAIKIN ADIMENDUNA (FACADE)



```

public class AireGirotua {
    public void piztu_itzali(boolean pAgindu) {
        if (pAgindu) {
            System.out.println("Aire girotuaren sistema piztuko da.");
        } else {
            System.out.println("Aire girotuaren sistema itzaliko da.");
        }
    }
}

```

```

public class UltrakorpFacade {

    private Argiak nArgiak = new Argiak();
    private AireGirotua nAireGirotua = new AireGirotua();
    private IgerilekuFiltro nIgerilekuFiltro = new IgerilekuFiltro();

    public void piztu_itzali(boolean pAgindu) {
        if(pAgindu) {
            System.out.println("---HIRU SISTEMAK PIZTU---");
            nArgiak.piztu_itzali(true);
            nAireGirotua.piztu_itzali(true);
            nIgerilekuFiltro.piztu_itzali(true);
        } else {
            System.out.println("---HIRU SISTEMAK ITZALI---");
            nArgiak.piztu_itzali(false);
            nAireGirotua.piztu_itzali(false);
            nIgerilekuFiltro.piztu_itzali(false);
        }
    }
}

```

## EXEKUZIOA

```

public class PAProba {
    public static void main(String[] args) {
        //Facade
        UltrakorpFacade ucf = new UltrakorpFacade();
        ucf.piztu_itzali(true); //piztu
        ucf.piztu_itzali(false); //itzali
    }
}

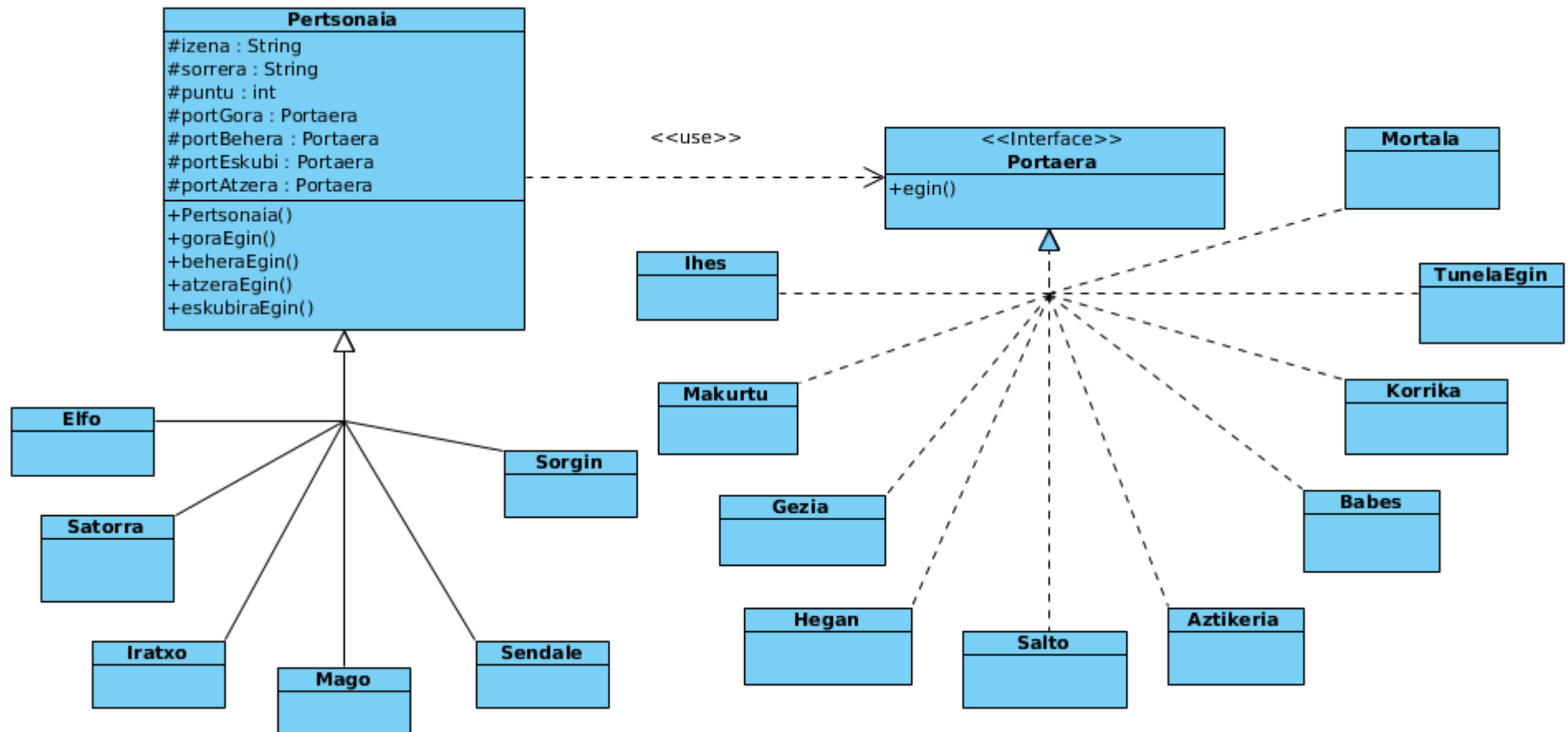
```

```

---HIRU SISTEMAK PIZTU---
Argien sistema piztuko da.
Aire girotuaren sistema piztuko da.
Igerilekuko filtroaren sistema piztuko da.
---HIRU SISTEMAK ITZALI---
Argien sistema itzaliko da.
Aire girotuaren sistema itzaliko da.
Igerilekuko filtroaren sistema itzaliko da.

```

## BIDEOJOKOA (STRATEGY)



## TESTUINGURUAK

```
public abstract class Pertsonaia {
    //Atributuak
    protected String izena;
    protected String sorrera;
    protected int puntuak;
    //Portaerak
    protected Portaera portGora;
    protected Portaera portBehera;
    protected Portaera portEskubi;
    protected Portaera portAtzera;
    //Mugimenduak
    public void goraEgin() {portGora.egin();}
    public void beheraEgin() {portBehera.egin();}
    public void atzeraEgin() {portAtzera.egin();}
    public void eskubiraEgin() {portEskubi.egin();}
}

public class Sorgin extends Pertsonaia{
    public Sorgin() {
        portGora = new Hegan();
        portBehera = new Makurtu();
        portAtzera = new Ihes();
        portEskubi = new Aztikeria();
    }
}

public class Mago extends Pertsonaia{
    public Mago() {
        portGora = new Salto();
        portBehera = new Makurtu();
        portAtzera = new Ihes();
        portEskubi = new Aztikeria();
    }
}
```

## ESTRATEGIA

```
public interface Portaera {
    public void egin();
}

public class Aztikeria implements Portaera{
    public void egin() {System.out.println("Aztikeria botatzen nabil!");}
}

public class Hegan implements Portaera{
    public void egin(){System.out.println("Hegan egiten nabil!");}
}

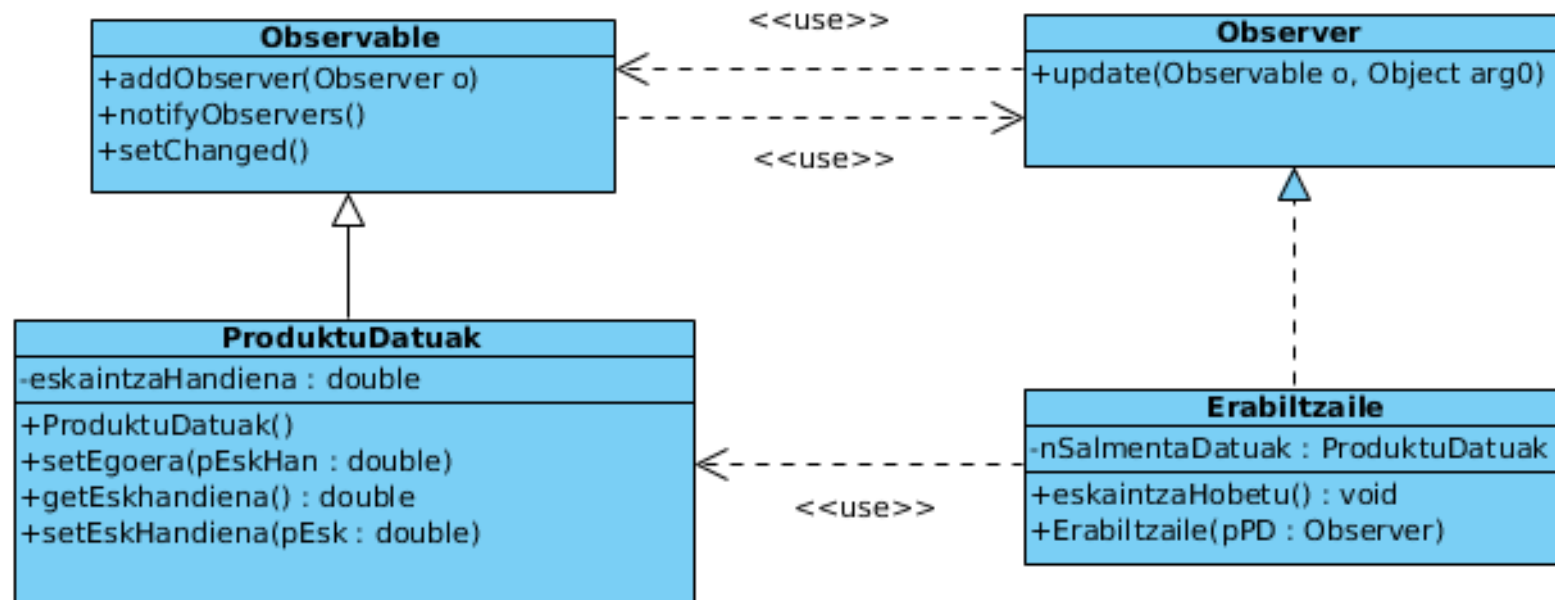
public class Salto implements Portaera{
    public void egin() {System.out.println("Salto egiten nabil!");}
}
```

```
public class PAProba {
    public static void main(String[] args) {
        Sorgin sorgin = new Sorgin();
        Mago mago = new Mago();
        System.out.println("Sorgina=> GORA BOTOIA");
        sorgin.goraEgin();
        System.out.println("Magoa=> GORA BOTOIA");
        mago.goraEgin();
        System.out.println("Sorgina=> BEHERA BOTOIA");
        sorgin.beheraEgin();
        System.out.println("Magoa=> BEHERA BOTOIA");
        mago.beheraEgin();
    }
}
```

## EXEKUZIOA

```
Sorgina=> GORA BOTOIA
Hegan egiten nabil!
Magoa=> GORA BOTOIA
Salto egiten nabil!
Sorgina=> BEHERA BOTOIA
Makurtu egin naiz!
Magoa=> BEHERA BOTOIA
Makurtu egin naiz!
```

## eBAY (OBSERVER)



```

import java.util.Observable;
import java.util.Observer;

public class Erabiltzailea implements Observer{
    Observable pPD;

    public Erabiltzailea (Observable pObs) {
        pPD = pObs;
        pPD.addObserver(this);
    }
    @Override
    public void update(Observable o, Object arg) {
        ((ProduktuDatuk)pPD).setEskHandiena(eskaintzaHobetu());
    }

    public double eskaintzaHobetu () {
        double eskBerri = ((ProduktuDatuk)pPD).getEskHandiena() + 10.0;
        System.out.println("Erabiltzaileak eskaintza hobetuko du, eta " + eskBerri + " eurora igo du.");
        return eskBerri;
    }
}

```

```

import java.util.Observable;
import java.util.Observer;

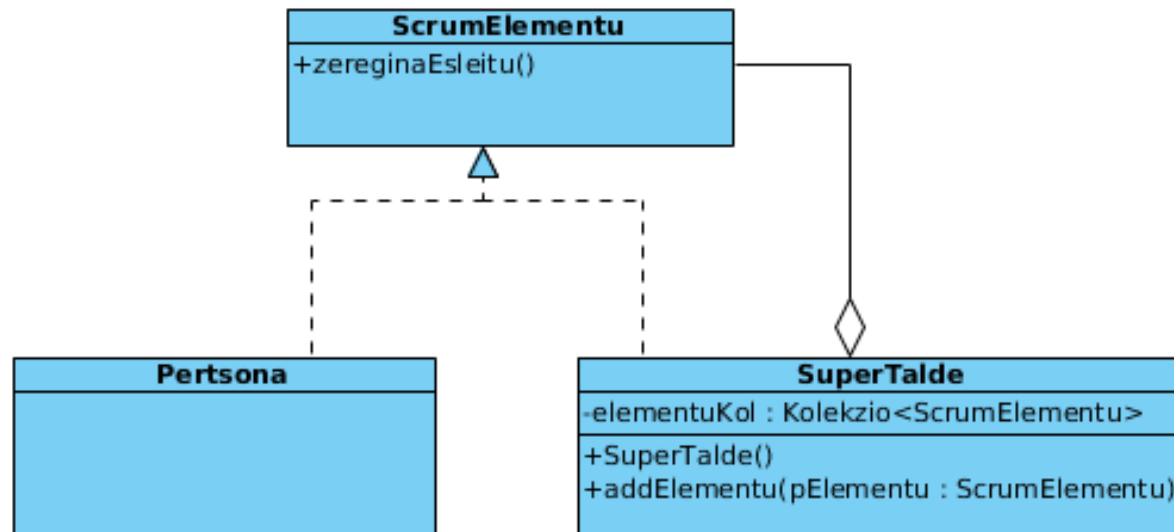
public class ProduktuDatuk extends Observable{

    private double eskaintzaHandiena = 0.0;

    public void setEgoera(double pEskHan) {
        if (eskaintzaHandiena < pEskHan) {
            System.out.println("Kanpotik eskaintza hobetu dute, orain " + pEskHan + " euro da!");
            eskaintzaHandiena = pEskHan;
            setChanged();
            notifyObservers();
        }
    }
    public double getEskHandiena() {return eskaintzaHandiena;}
    public void setEskHandiena(double pEsk) {eskaintzaHandiena = pEsk;}
}

```

## SCRUM (COMPOSITE)



```
public interface ScrumElementu {
    public void zereginaEsleitu();
}
```

```
public class Pertsona implements ScrumElementu{
    @Override
    public void zereginaEsleitu() {
        System.out.println("Pertsonari dagokion zeregina esleitu didate.");
    }
}
```

SOILA

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

public class SuperTalde implements ScrumElementu{
    private List<ScrumElementu> elementuKol = new ArrayList<ScrumElementu>();
    @Override
    public void zereginaEsleitu() {
        System.out.println("SuperTaldeko zereginak esleitu dizkidate.");
        Iterator<ScrumElementu> itSE = getIter();
        while(itSE.hasNext()) {
            ScrumElementu se = itSE.next();
            se.zereginaEsleitu();
        }

        public void addElementu(ScrumElementu pElementu) {
            elementuKol.add(pElementu);
        }

        private Iterator getIter() {
            return elementuKol.iterator();
        }
    }
}
```

KONPOSATUA

```
public class PAProba {

    public static void main(String[] args) {
        //HOSTOAK
        Pertsona p11 = new Pertsona();
        Pertsona p12 = new Pertsona();
        Pertsona p13 = new Pertsona();
        Pertsona p21 = new Pertsona();
        Pertsona p22 = new Pertsona();
        Pertsona p23 = new Pertsona();
        //NODOAK
        SuperTalde sp1 = new SuperTalde();
        SuperTalde sp2 = new SuperTalde();
        SuperTalde sp3 = new SuperTalde();
        //HIERARKIA OSATU
        sp1.addElementu(p11);
        sp1.addElementu(p12);
        sp1.addElementu(p13);
        sp2.addElementu(p21);
        sp2.addElementu(p22);
        sp2.addElementu(p23);
        sp3.addElementu(sp1);
        sp3.addElementu(sp2);
        //ZEREGINAK ESLEITU
        sp3.zereginaEsleitu();
    }
}
```

