

OBP-ko adibidea

# SUPERMERKATUA

# Enuntziatua

Supermerkatu baten izakinen eguneraketa prozesua automatizatu nahi da. Supermerkatuak 40 terminal ditu kutxa erregistratzaile moduan lan egiten. Bakoitzak, egunean zehar bere bitartez egindako salmenta guztien informazioa erregistratzen du. Salmenta bakoitzeko hurrengo informazioa erregistratzen da:

## **Artikulu kodea, saldutako unitate kopurua**

Egunaren amaieran 40 kutxetan gordetako informazioa supermerkatuaren izakin globalak eguneratzeko erabiltzen da, saldutako artikuluei dagokionez. Izakin inbentarioak salmentan dagoen artikulu bakoitzeko honako informazioa gordetzen du:

## **Art. kodea, izakinak, unit. prezioa, stock minimoa**

# Booch-en Gida

Ariketaren deskribapenean agertzen diren izenak aztertu:

- Izen **arruntak**: klaseak edo TAD-ak
- Izen **propioak** eta **erreferentzia zuzeneko** izenak, instantzia espezifiko baterako: instantziak edo Singletonak

# Enuntziatua

**Supermerkatu** baten izakinen eguneraketa prozesua automatizatu nahi da. Supermerkatuak 40 **terminal** ditu kutxa erregistratzaile moduan lan egiten. Bakoitzak, egunean zehar bere bitartez **egindako salmenta** guztien informazioa erregistratzen du. **Salmenta** bakoitzeko hurrengo informazioa erregistratzen da:

## **Artikulu kodea, saldutako unitate kopurua**

Egunaren amaieran 40 kutxetan gordetako informazioa supermerkatuaren izakin globalak eguneratzeko erabiltzen da, saldutako **artikuluei** dagokionez. Izakin **inbentarioak** salmentan dagoen artikulu bakoitzeko honako informazioa gordetzen du:

## **Art. kodea, izakinak, unit. prezioa, stock minimoa**

# Booch-en Gida

Ariketaren deskribapenean agertzen diren izenak aztertu:

- TAD-ak:

  - Terminala, Salmenta, Artikulua, SalmentaZerrenda

- Singleton-ak:

  - Supermerkatua, Inbentarioa

# Booch-en Gida

Supermerkatua (Singleton)

terminalZer: Zerrenda

Terminala (TAD)

identifikadorea: String

salmentaZer: Zerrenda

SalmentaZerrenda (TAD)

salmZer: Zerrenda

Inbentarioa (Singleton)

artikuluZer: Zerrenda

Artikulua (TAD)

artKodea: int

izakinak: int

unitPrezioa: float

stockMin: int

Salmenta (TAD)

artKodea: int

unitateak: int

# Ellis-en Gida

izakinak Eguneratu metodoaren algoritmoa:

- ⦿ Supermerkatuko terminal bakoitzarentzako:
- ⦿ Salmenta Zerrenda zeharkatu Inbentarioa eguneratzeko.
  - ⦿ Zerrendako Salmenta bakoitzeko:
    - ⦿ Inbentarioa eguneratu
    - ⦿ Artikuluaren unitateak eguneratu

# Ellis-en Gida

Supermerkatua

izakinak Eguneratu metodoaren algoritmoa:

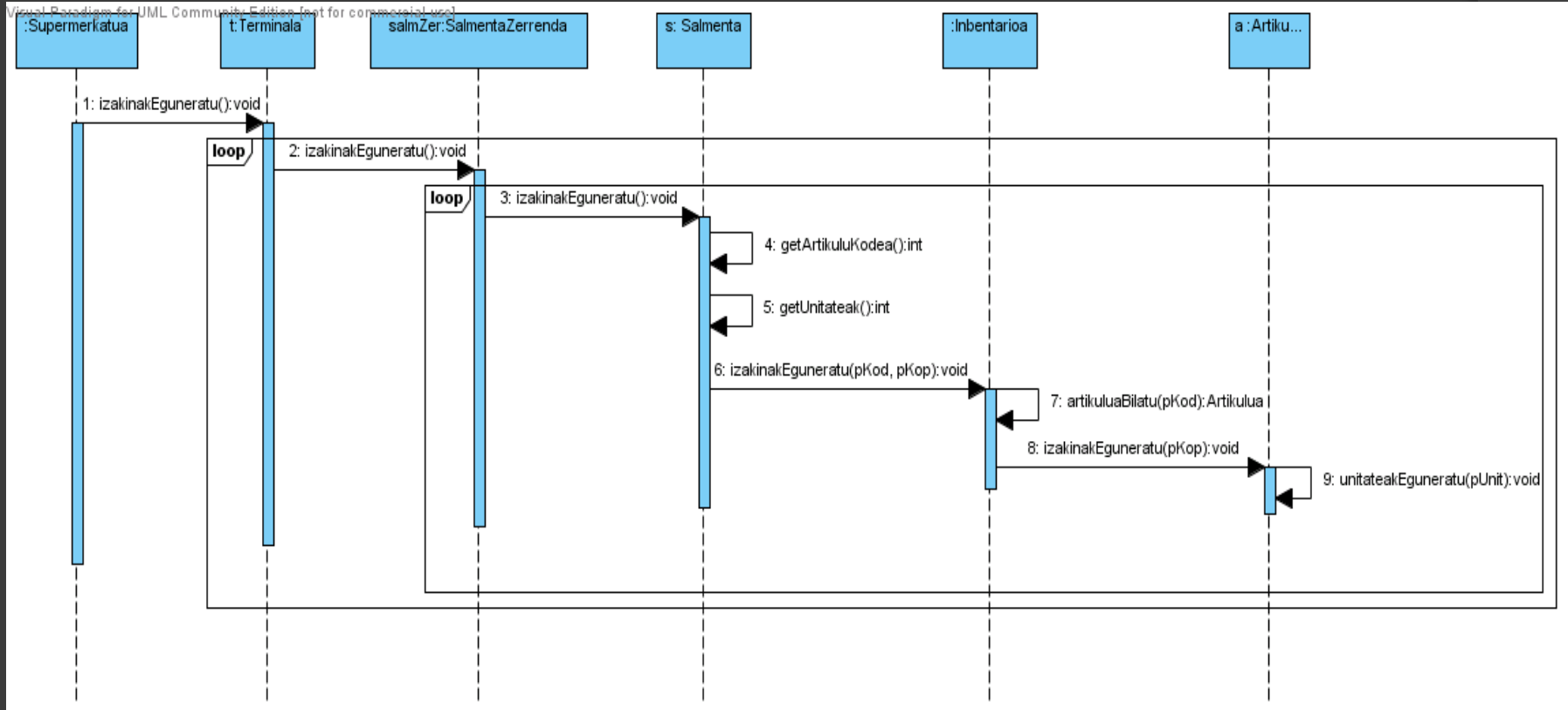
- ◉ Supermerkatuko terminal bakoitzarentzako: Terminala
- ◉ Salmenta Zerrenda zeharkatu Inbentarioa eguneratzeko Salmenta Zerrenda
  - ◉ Zerrendako Salmenta bakoitzeko: Salmenta
    - ◉ Inbentarioa eguneratu Inbentarioa
    - ◉ Artikuluaren unitateak eguneratu Artikulua



# Ellis-en Gida

- ◉ Supermerkatua: izakinakEguneratu():void
- ◉ Terminala: izakinakEguneratu(): void
- ◉ SalmentaZerrenda: izakinakEguneratu(): void
- ◉ Salmenta:       izakinakEguneratu(): void  
                  getArtikuluKodea(): int  
                  getUnitateak(): int
- ◉ Inbentarioa: izakinakEguneratu(pKod:int, pKop:int):  
              void  
                              artikuluaBilatu(pKod: int): Artikulua
- ◉ Artikulua: unitateakEguneratu(pUnit: int): void

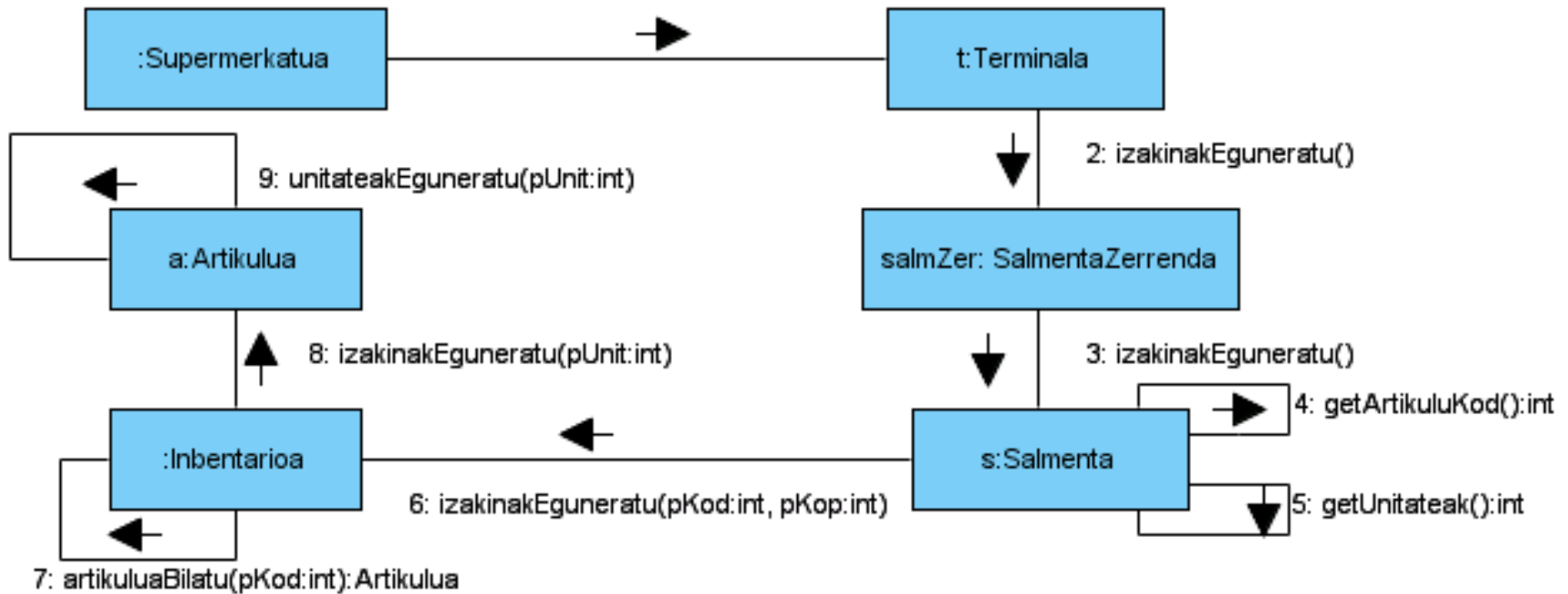
# Sekuentzia Diagrama



# Komunikazio Diagrama

Visual Paradigm for UML Community Edition [not for commercial use]

1: izakinakEguneratu()



# Klaseak

Visual Paradigm for UML Community Edition

## Supermerkatua

-mSupermerkatua : Supermerkatua  
-terminalZer : Zerrenda  
-Supermerkatua() : Supermerkatua  
+getSupermerkatua() : Supermerkatua  
+izakinakEguneratu() : void  
-getIteradorea() : Iterator

Visual Paradigm for UML Community Edition

## Terminala

-identifikadorea : String  
-salmZer : SalmentaZerrenda  
+Terminala(pKod : String) : Terminala  
+izakinakEguneratu() : void

Visual Paradigm for UML Community Edition [not for c

## Inbentarioa

-mInbentarioa : Inbentarioa  
-artikuluZer : Artikulua  
-Inbentarioa() : Inbentarioa  
+getInbentarioa() : Inbentarioa  
+izakinakEguneratu(pKod : int, pKop : int) : void  
-artikuluaBilatu(pKop : int) : Artikulua

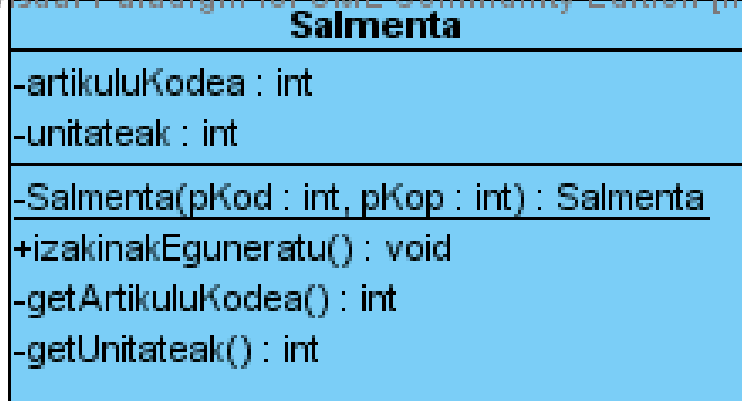
Visual Paradigm for UML Community Edition [not

## SalmentaZerrenda

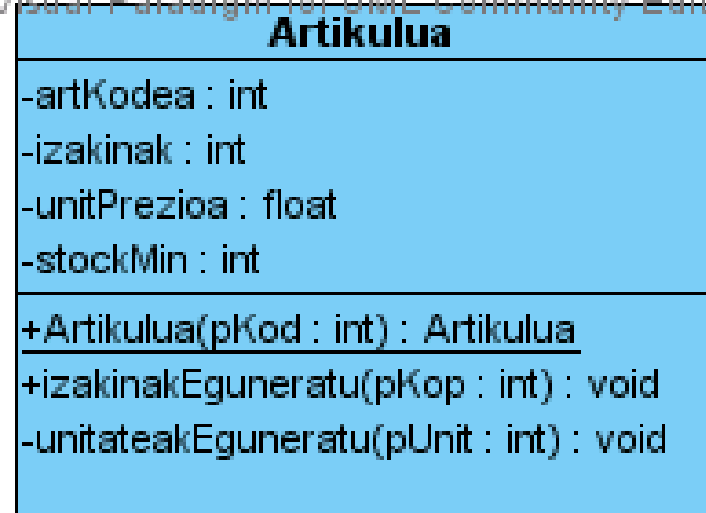
-salmZerr = Zerrenda  
+SalmentaZerrenda() : SalmentaZerrenda  
+izakinakEguneratu() : void  
-getIteradorea() : Iterator

# Klaseak

Visual Paradigm for UML Community Edition [not

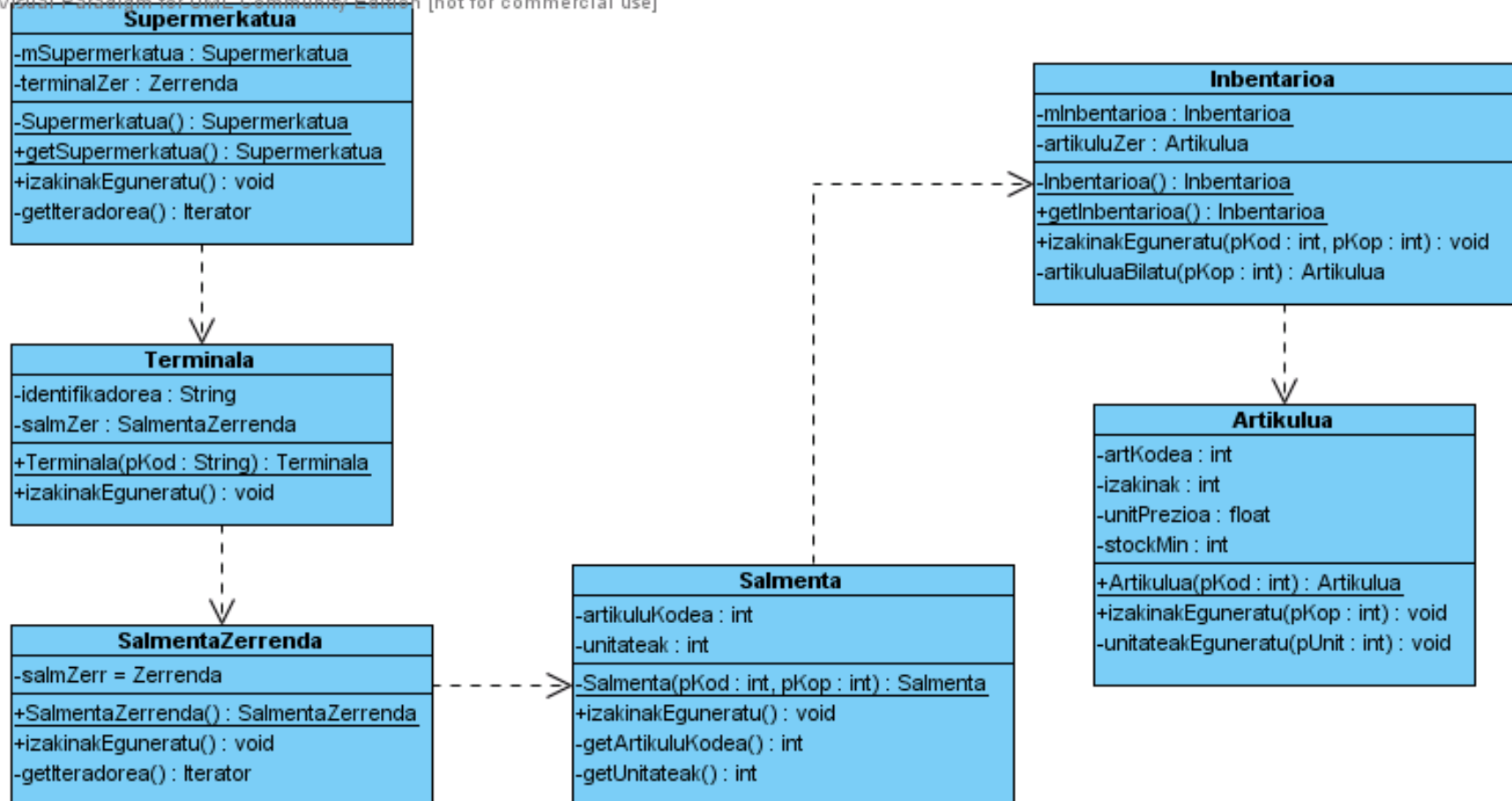


Visual Paradigm for UML Community Editio



# UML Diagrama

Visual Paradigm for UML Community Edition [not for commercial use]



# Supermerkatua Klasea

```
public class Supermerkatua{  
  
    //Atributoak  
    private Vector terminalZerr;  
  
    //Konstruktorea:  
    private Supermerkatua(){  
        terminalZerr= new Vector();  
    }  
  
    //Singletonaren instantzia:  
    private static Supermerkatua  
    mSupermerkatua = new  
        Supermerkatua();  
  
    //instantzia hartzeko metodoa:  
    public Supermerkatua  
        getSupermerkatua(){  
        return mSupermerkatua;  
    }  
  
    //Metodoak:  
    private Iterator getIteradorea(){  
        return terminalZerr.iterator();  
    }  
  
    public void izakinakEguneratu(){  
        Iterator it = getIteradorea();  
        Terminala t;  
        while (it.hasNext()){  
            t= it.next();  
            t.izakinakEguneratu();  
        }  
    }  
} // Supermerkatua klasearen  
    amaiera
```

# Terminala Klasea

```
public class Terminala{
```

```
//Atributoak:
```

```
private String identifikadorea;  
private SalmentaZerrenda salmZer;
```

```
//Konstruktorea:
```

```
public Terminala(String pIdent){  
    identifikadorea = pIdent;  
}
```

```
//Metodoak:
```

```
public void izakinakEguneratu(){  
    salmZer.izakinakEguneratu()  
}
```

```
} // Terminala klasearen amaiera
```



# SalmentaZerrenda Klasea

```
public class SalmentaZerrenda{
```

```
    //Atributoak:
```

```
    private salmZer Vector;
```

```
    //Konstruktorea:
```

```
    public SalmentaZerrenda(){  
        salmZer = new Vector();  
    }
```

```
    //Metodoak:
```

```
    private Iterator getIteradorea(){  
        return salmZer.Iterator();  
    }
```

```
    public void izakinakEguneratu(){  
        Iterator it = getIteradorea();  
        Salmenta s;  
        while (it.hasNext()){  
            s = it.next();  
            s.izakinakEguneratu();  
        }  
    }
```

```
} // SalmentaZerrenda klasearen  
    amaiera
```

# Salmenta Klasea

```
public class Salmenta{
```

```
//Atributoak:
```

```
private int artikuluKodea;  
private int unitateak;
```

```
//Konstruktorea:
```

```
public Salmenta(int pKod, int  
    pUnit){  
    artikuluKodea = pKod;  
    unitateak = pUnit;  
}
```

```
//Metodoak:
```

```
public void izakinakEguneratu(){  
    Inbentarioa nireInbentarioa =  
        Inbentarioa.getInbentarioa();  
    nireInbentarioa.izakinakEguner  
        atu(getArtikuluKodea(),  
            getUnitateak());  
}
```

```
private int getArtikuluKodea(){  
    return artikuluKodea;  
}
```

```
private int getUnitateak(){  
    return unitateak;  
}
```

```
} // Salmenta klasearen amaiera
```

# Inbentarioa Klasea

```
public class Inbentarioa{
```

```
//Atributoak:
```

```
private Vector artikuluZer;
```

```
//Konstruktorea:
```

```
private Inbentarioa(){  
    artikuluZer = new Vector();
```

```
//Singletonaren instantzia:
```

```
private static Inbentarioa nireInbentarioa = new  
    Inbentarioa();
```

```
// instantzia hartzeko metodoa:
```

```
public Inbentarioa getInbentarioa() {  
    return nireInbentarioa;  
}
```

```
//Metodoak:
```

```
public void izakinakEguneratu(pKod: int, pKant: int){  
    Artikulua a = artikuluBilatu(pKod);  
    a.IzakinakEguneratu(pKant);  
}
```

```
private Artikulua artikuluBilatu(pKod: int){
```

```
    Iterator it = getIteradorea()
```

```
    Artikulua a;
```

```
    boolean encontrado = false;
```

```
    while (!encontrado && it.hasNext()) {
```

```
        a = it.next();
```

```
        if (a.berdina(pKod))
```

```
            aurkitua = true;
```

```
    }
```

```
    if (aurkitua)
```

```
        return a;
```

```
    else
```

```
        return null;
```

```
}
```

```
private Iterator getIteradorea(){
```

```
    return artikuluZer.iterator();
```

```
}
```

```
} // Inbentarioa klasearen inbentarioa
```

# Artikulua Klasea

```
public class Artikulua{
```

```
//Atributoak:
```

```
private int artikuluKodea;  
private int izakinak;  
private float unitPrezioa;  
private int stockMin;
```

```
//Konstruktorea:
```

```
public Artikulua(pKod: int){  
    artikuluKodea = pKod;  
}
```

```
//Metodoak:
```

```
public void izakinakEguneratu(int  
    pUnit){  
    izakinak =  
    izakinakEguneratu(izakinak –  
    pUnit);  
}
```

```
public void unitateakEguneratu (int  
    pUnit) {  
    izakinak = pUnit;  
}
```

```
public boolean berdina (int pKod) {  
    if (artikuluKodea == pKod)  
        return true;  
    else  
        return false;  
}
```

```
} // Artikulua klasearen amaiera
```