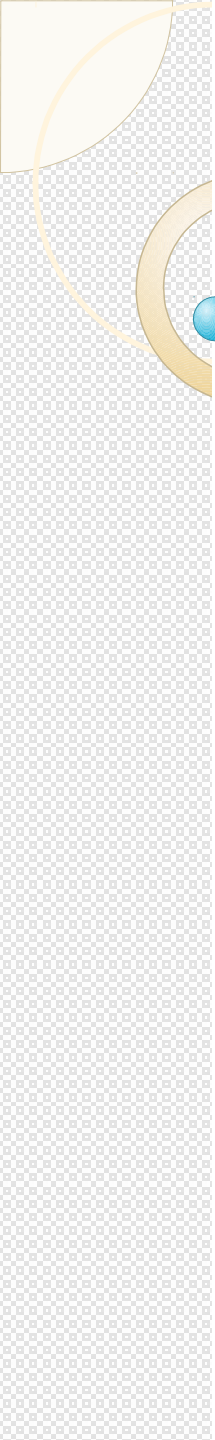


Bigarren azterketa partzialaren soluzioa

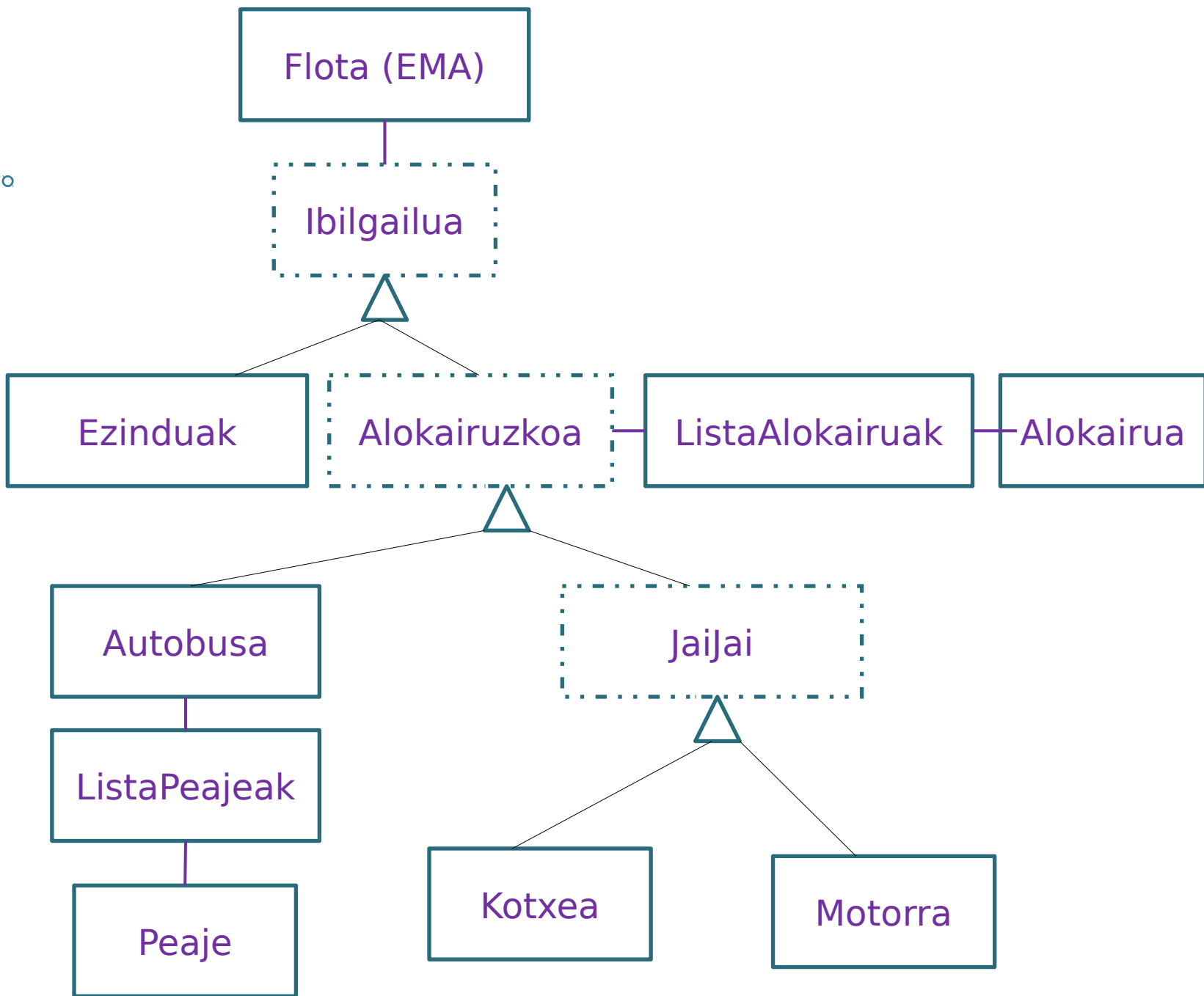


PMOBO 2013



Klase Diagrama

A Bertsioa



Flota (EMA)

- Atributuak

- - ArrayList<Ibilgailua> lista;
- - static Flota nireFlota;

- Metodoak

- - eraikitzailea
- + static getFlota(): Flota
- - getIteradorea(): Iterator<Ibilgailua>
- + inprimatuIrabaziMensualak(): void
- - kalkulatuDiruSarrerak(): double
- - kalkulatuGastuak(): double

Ibilgailua (abstraktua)

- Atributuak
 - - String matrikula;
 - - String modelo;
 - - int numBidaiaariak;
- Metodoak
 - + constructora
 - + abstract kalkulatuDiruSarerrak(): double
 - + abstract kalkulatuGastuak(): double

Ezinduen

- Atributuak
 - - String deskripzioa;
- Metodoak
 - + eraikitzailea
 - + kalkulatuDiruSarreak(): double
 - + kalkulatuGastuak(): double

Alokairuzkoa (abstraktua)

- Atributuak

- - ListaAlokairuak listaAlok;
- - static double gastuaOrduko;

- Metodoak

- + eraikitzailea
- - static getGastuaOrduko(): double
- + kalkulatuGastua(): double

ListaAlokairuak

- Atributuetak

- - ArrayList<Alokairua> lista;

- Metodoak

- + eraikitzailea
 - - getIteradorea(): Iterator<Alokairua>
 - + orduTotalak(): double
 - + kmsTotalak(): double
 - + numAlokairuNormalak(): double
 - + numAlokairuJaijai(): double

Alokairua

- Atributuak
 - - int orduak;
 - - double kmak;
 - - boolean jaijai;
- Metodoak
 - + eraikitzailea
 - + getOrduak(): int
 - + getKmak(): double
 - + getJaijai(): boolean

Jaijai (abstraktua)

- Atributuak
 - - static double OinarrizkoTarifa;
- Metodoak
 - + eraikitzailea
 - - static getOinarrizkoTarifa(): double
 - + kalkulatuDiruSarrerak(): double

Kotxea

- Atributuak
 - - double salneurriKmko;
- Metodoak
 - + eraikitzailea
 - - getSalneurriKmko(): double
 - + kalkulatuDiruSarrerak(): double

Motorra

- Atributuak
 - - static double kuotaOrduko;
- Metodoak
 - + eraikitzailea
 - - static getKuotaOrduko(): double
 - + kalkulatuDiruSarrerak(): double

Autobusa

- Atributuak

- - ListaPeajeak lista;
- - double salneurriOrduko;

- Metodoak

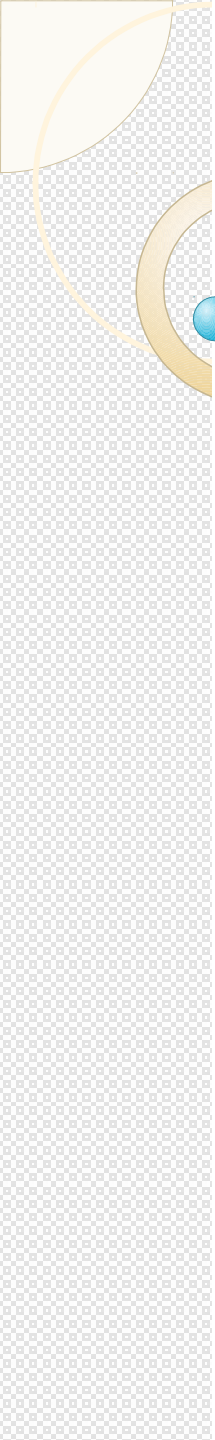
- + eraikitzailea
- - getSalneurriOrduko(): double
- + kalkulatuDiruSarrerak(): double
- + kalkulatuAutopistakoGastuak(): double

ListaPeajeak

- Atributuak
 - - ArrayList<Peajea> lista;
- Metodoak
 - + eraikitzailea
 - - getIteradorea(): Iterator<Peaje>
 - + peajeenGehiketa(): double

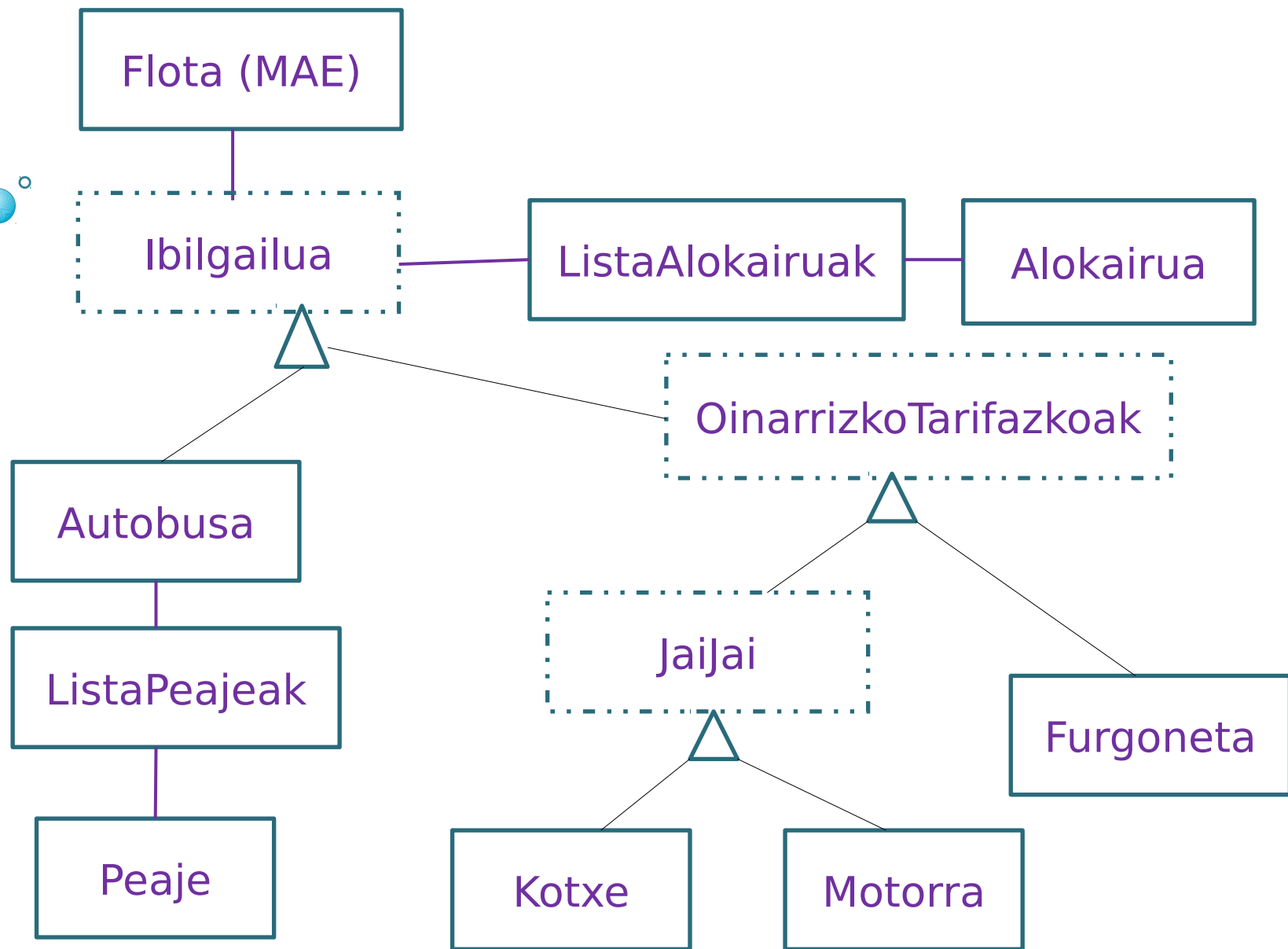
Peajea

- Atributuak
 - - String idAutopista;
 - - double gastua;
- Metodoak
 - + eraikitzailea
 - + getGastua;



Klase Diagrama

B bertsioa





A Bertsioaren pareko klaseak

- Flota
- ListaAlokairuak eta Alokairua
- Autobus, ListaPeajeak eta Peajea
- Kotxe eta Motorra

Ibilgailua (abstraktua)

- Atributuak

- - String matrikula;
- - String modeloa;
- - int numBidaiaariak;
- - ListaAlokairuak: listaAlo;

- Metodoak

- + eraikitzailea
- + abstract kalkulatuDiruSarrerak(): double
- + kalkulatuGastuak(): double



OinarrizkoTarifazkoak (abstraktua)

- Atributuak
 - - static double oinarrizkoTarifa;
- Metodoak
 - + eraikitzailea
 - * static getOinarrizkoTarifa(): double

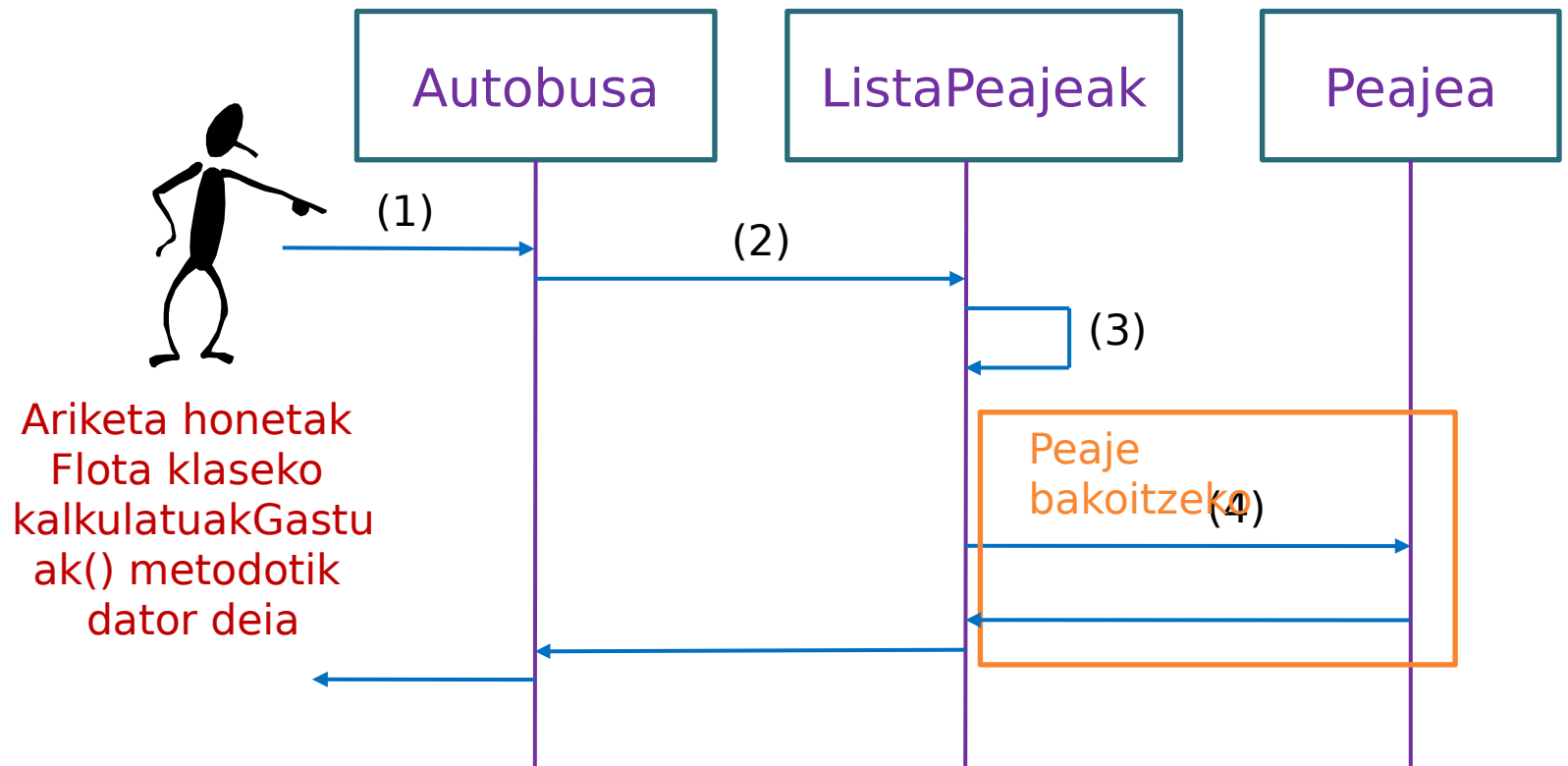
Furgoneta

- Atributuak
 - - int kapazitatea;
- Metodoak
 - + eraikitzailea
 - - getKapazitatea(): int
 - + kalkulatuDiruSarrerak(): double



Sekuentzia Diagrama

A eta B bertsioak



1)kalkulatuAutopistaGastuak():double

2)peajeenGehiketa(): double

3)getIteradorea(): Iterator<Peajea>

4)getGastua(): double



Inplementazioa

A Bertsioa

Flota

```
public void inprimatuIrabaziMensualak()
{
    double irabazia;

    irabazia = this.kalkulatuDiruSarrerak() -
this.kalkulatuGastuak();

    System.out.println("Irabazia: " + irabazia + " euro da");
}
```

Flota

```
public void inprimatuIrabaziMensualak()
{
    double irabazia;

    irabazia = this.kalkulatuDiruSarrerak() -
this.kalkulatuGastuak();

    System.out.println("Irabazia: " + irabazia + " euro da");
}
```

Ibilgailua

```
public abstract double kalkulatuDirusarrerak();
```

Ezezina

```
public double kalkulatuDirusarrerak()  
{  
    return 50.0;  
}
```

Autobusa

```
public double kalkulatuDirusarrerak()  
{  
  
    return this.listaAlo.orduTotalak()*this.getSalneurriOrduko();  
}
```

Jaijai

```
public double kalkulatuDirusarrerak()  
{  
    int normalak = this.listaAlo.numAloNormalak();  
    int deskontuarekin = this.listaAlo.numAlokairuJaijai();  
    double tb = getTarifaBase();  
    return (normales*tb)+(conDescuento*0.9*tb);  
}
```

Kotxea

```
public double  
kalkulatuDirusarrerak()  
{  
    return  
    super.kalkulatuDirusarrerak()  
        +  
        this.listaAlo.kmsTotalak() *  
        this.getSalneurriKmko();  
}
```

Motorra

```
public double  
kalkulatuDirusarrerak()  
{  
    return  
    super.kalkulatuDirusarrerak()  
        +  
        this.listaAlo.orduTotalak() *  
        Moto.getKuotaOrduko();  
}
```

Flota

```
private double kalkulatuGastuak()
{
    Iterator<Ibilgailua> itr = this.getIteradorea();
    double akumPeaj = 0.0; -- Peajeen gehiketa
    double akumGF = 0.0; -- Gastu Finkoen gehiketa

    while (itr.hasNext())
    {
        Ibilgailua iBat = itr.next();
        akumGF = akumGF + iBat.kalkulatuGastuak();
        if (iBat instanceof Autobus)
        {
            if (akumPea < 25)
            {
                akumPea = akumPea +
                (Autobus)iBat.kalkulatuAutopistakoGastuak();
                if (akumPea > 25) { akumPea = 25; }
            }
        }
    }
    return akumGF + akumPea;
}
```



Ibilgailua

```
public abstract double kalkulatuGastuak();
```

Ezezina

```
public double kalkulatuGastuak()  
{  
    return 0.0;  
}
```

Alokairuzkoa

```
public double kalkulatuGastuak()  
{  
    return  
this.listaAlo.orduTotalak()*alokairuzkoa.getGastuaOrduko();  
}
```



Inplementazioa

B Bertsioa



A bertsioarekiko desberdintasunak

- Ibilgailua.kalkulatuGastuak() orain ez da abstraktua izango, eta A bertsioaren Alokairuzkoa inplementatzen den bezala inplementatuko da.

Furgoneta

```
public double kalkulatuDirusarrerak()
{
    double emaitza = Furgoneta.getOinarrizkoTarifa();

    if (this.getKapazitatea() == 7)
    {
        emaitza = emaitza + 5;
    }
    else if (this.getKapazitatea() == 15)
    {
        emaitza = emaitza + 10;
    }

    return emaitza;
}
```