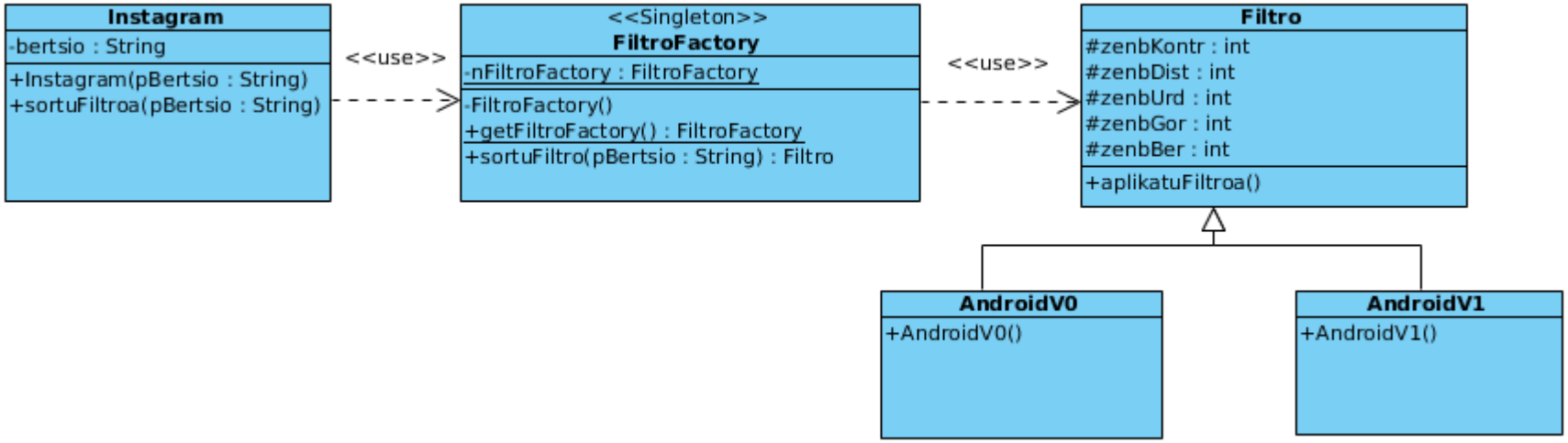
INSTRAGRAM (FACTORY)



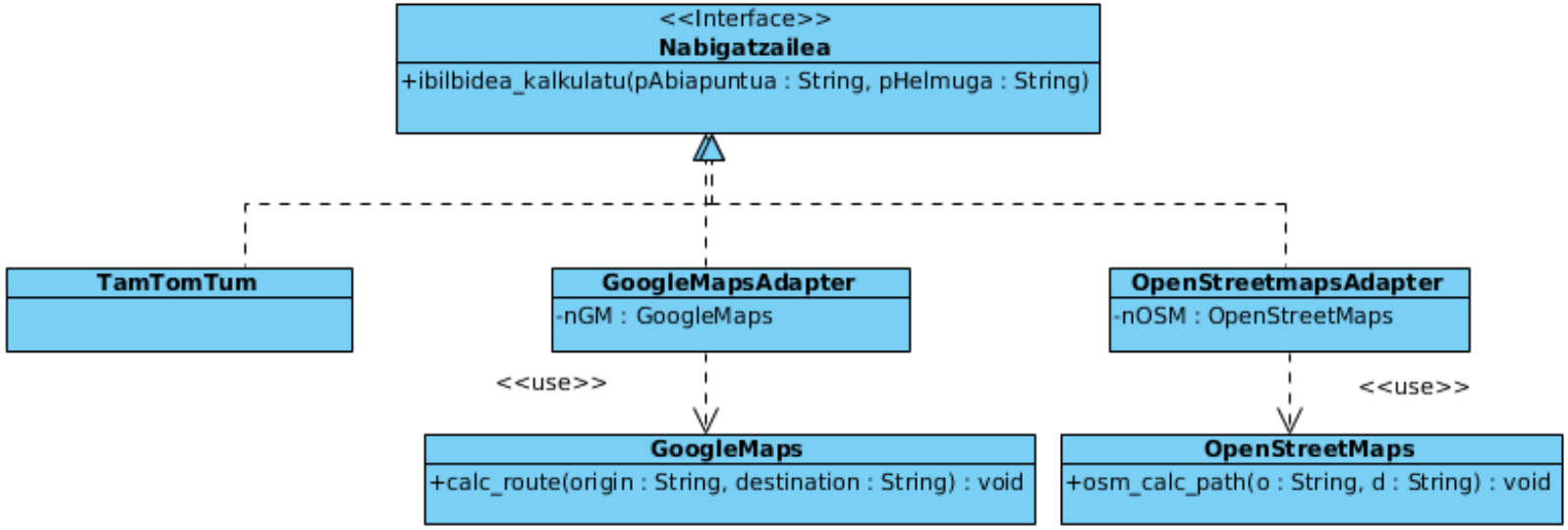
```java
public class Instagram {

    private String bertsio;

    public Instagram() {    }

    public void sortuFiltro() {
        FiltroFactory.getFiltroFactory().sortuFiltro(bertsio);
    }
}
```

```java
public class FiltroFactory {

    private static FiltroFactory nFiltroFactory;
    private FiltroFactory() { }

    public static FiltroFactory getFiltroFactory() {
        if (nFiltroFactory == null) {
            nFiltroFactory = new FiltroFactory();
        }
        return nFiltroFactory;
    }

    public Filtro sortuFiltro(String pBertsio) {
        switch (pBertsio) {
            case "v0":
                return new AndroidV0();
            case "v1":
                return new AndroidV1();
            default:
                return new AndroidV0();
        }
    }
}
```

```java
public abstract class Filtro {
    protected int zenbKontr;
    protected int zenbDist;
    protected int zenbUrd;
    protected int zenbGor;
    protected int zenbBer;

    public void aplikatuFiltroa() {
        System.out.println("Filtro abstraktua naiz.");
    }
}

public class AndroidV0 extends Filtro {
    public AndroidV0() {
        zenbKontr = 2;
        zenbDist = 3;
        zenbUrd = 4;
        zenbGor = 5;
        zenbBer = 6;
    }

    public void aplikatuFiltroa() {
        System.out.println("Android v0 filtro konkretua aplikatzen.");
    }
}
```

```java
public class ProbaPA {

    public static void main(String[] args) {
        FiltroFactory ff = FiltroFactory.getFiltroFactory();
        //Filtroak sortu
        AndroidV0 a0 = (AndroidV0)ff.sortuFiltro("v0");
        AndroidV1 a1 = (AndroidV1)ff.sortuFiltro("v1");
        //Filtroak aplikatu, bertsio desberdinetarako
        a0.aplikatuFiltroa();
        a1.aplikatuFiltroa();
    }
}
```

KOTXE NABIGATZAILEA (ADAPTER)



```java
public interface Nabigatzailea {
    void ibilbidea_kalkulatu(String pAbiapuntua, String pHelmuga);
}

public class TamTomTum implements Nabigatzailea {
    public void ibilbidea_kalkulatu(String pAbiapuntua, String pHelmuga) {
        System.out.println(String.format("TTT nabigatzailea erabiltzen-> pAbiapuntu: %s phelmuga %s", pAbiapuntua, pHelmuga));
    }
}
```

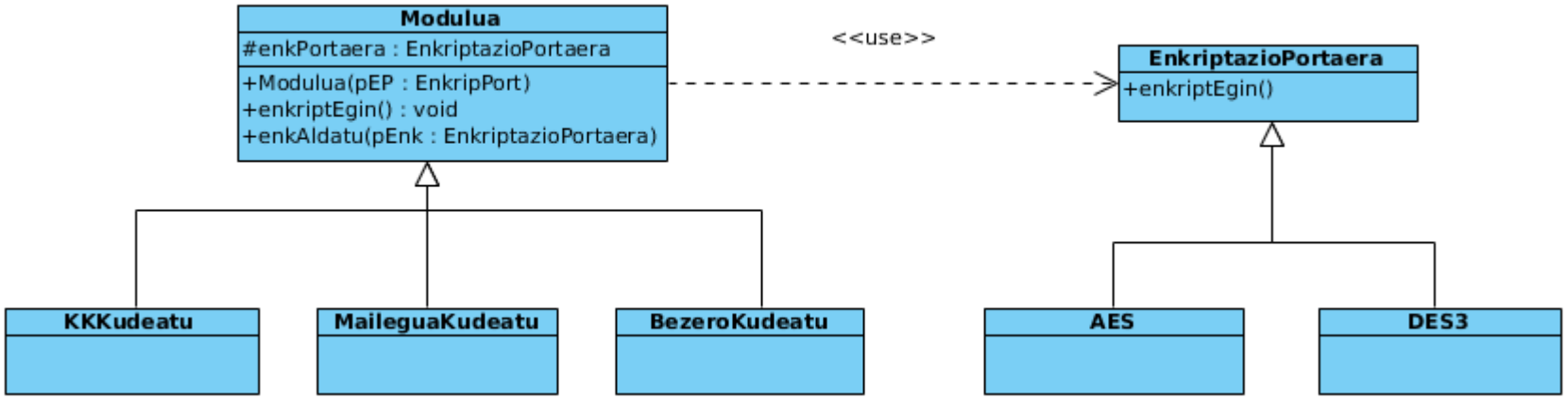
```java
public class OpenStreetmapsAdapter implements Nabigatzailea {
    private OpenStreetMaps nOSM = new OpenStreetMaps();
    public void ibilbidea_kalkulatu(String pAbiapuntua, String pHelmuga) {
        nOSM.osm_calc_path(pAbiapuntua, pHelmuga);
    }
}

public class OpenStreetMaps {
    public void osm_calc_path(String o, String d) {
        System.out.println(String.format("OSM nabigatzailea erabiltzen-> o: %s d: %s", o, d));
    }
}
```

```java
public class ProbaPA {
    public static void main(String[] args) {
        TamTomTum ttt = new TamTomTum();
        ttt.ibilbidea_kalkulatu("Bilbo", "Gasteiz");
        OpenStreetmapsAdapter osm = new OpenStreetmapsAdapter();
        osm.ibilbidea_kalkulatu("Gernika", "Leioa");
        GoogleMapsAdapter gm = new GoogleMapsAdapter();
        gm.ibilbidea_kalkulatu("Irun", "Lesaka");
    }
}
```

EXEKUZIOAN

```
TTT nabigatzailea erabiltzen-> pAbiapuntu: Bilbo phelmuga Gasteiz
OSM nabigatzailea erabiltzen-> o: Gernika d: Leioa
GM nabigatzailea erabiltzen-> Origin: Irun Destination Lesaka
```

ENKRIPTAZIOA (STRATEGY)



```java
public class Modulua {

    protected EnkriptazioPortaera enkPortaera;

    public void enkriptEgin() {
        enkPortaera.enkriptEgin();
    }

    public void enkriptAldatu(EnkriptazioPortaera pEP) {
        enkPortaera = pEP;
        System.out.println("Enkriptazio portaera aldatuko dut");
    }
}

public class MaileguaKudeatu extends Modulua {
    public MaileguaKudeatu() {
        enkPortaera = new DES3();
        System.out.println("Bezeroak kudeatzeko modulua naiz, eta DES3 enkriptazioa daukat defektuz.");
    }
}
```

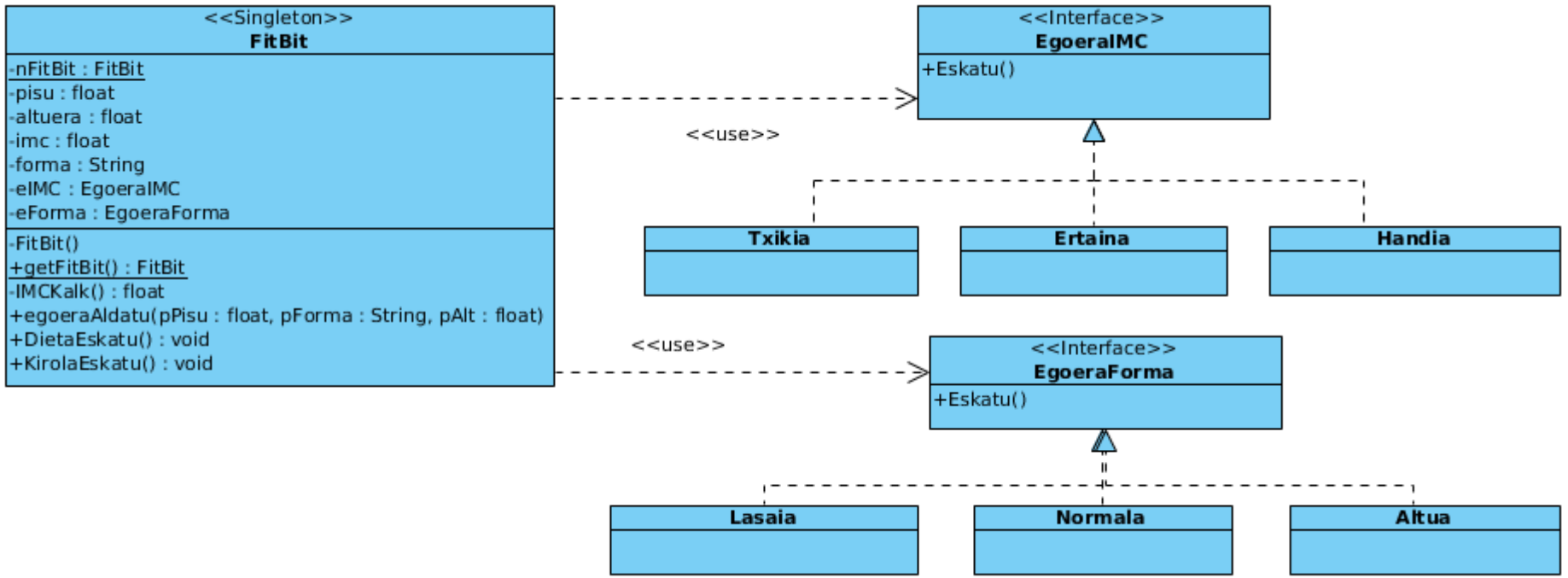
```java
public class EnkriptazioPortaera {
    public void enkriptEgin() { }
}

public class AES extends EnkriptazioPortaera {
    public void enkriptEgin() { System.out.println("AES enkriptazioa egiten nabil");}
}
```

```java
public class PA_Proba {
    public static void main(String[] args) {
        KKKudeatu kk = new KKKudeatu();
        kk.enkriptEgin();
        kk.enkriptAldatu(new DES3());
        kk.enkriptEgin();
        MaileguaKudeatu mk = new MaileguaKudeatu();
        mk.enkriptEgin();
    }
}
```

EXEKUZIOAN

```
Kontu korronteak kudeatzeko modulua naiz, eta AES enkriptazioa daukat defektuz.
AES enkriptazioa egiten nabil
Enkriptazio portaera aldatuko dut
DES3 enkriptazioa egiten nabil
Bezeroak kudeatzeko modulua naiz, eta DES3 enkriptazioa daukat defektuz.
DES3 enkriptazioa egiten nabil
```

FITNESS TRACKER (STATE)



```java
public class FitBit {

    private static FitBit nFitBit;
    private float pisu;
    private float altuera;
    private float imc;
    private String forma;
    private EgoeraIMC eIMC;
    private EgoeraForma eForma;

    private FitBit() {}

    public static FitBit getFitBit() {
        if (nFitBit == null) {
            nFitBit = new FitBit();
        }
        return nFitBit;
    }

    private float IMCKalk() {return pisu/(altuera*altuera);}

    public void egoerAldatu(float pPisu, String pForma, float pAlt) {
        //Atributuak
        pisu = pPisu;
        forma = pForma;
        altuera = pAlt;
        imc = IMCKalk();
        //IMC egoera inferitu
        if (imc<18.5) eIMC = new Txikia();
        else if ((15.5<imc) && (imc<24.5)) eIMC = new Ertaina();
        else eIMC = new Handia();
        //Forma egoer inferitu
        switch(forma) {
            case "lasaia":
                eForma = new Lasaia();
                break;
            case "normala":
                eForma = new Normala();
                break;
            case "altua":
                eForma = new Altua();
                break;
            default:
                eForma = new Normala();
                break;
        }
    }
    public void DietaEskatu() {
        eIMC.Eskatu();
    }
    public void KirolaEskatu() {
        eForma.Eskatu();
    }
}
```

```java
public interface EgoeraForma {
    public void Eskatu();
}

public class Lasaia implements EgoeraForma {
    public void Eskatu() {
        System.out.println("Forma lasaia daukazu-> Ariketa plangintza lasaia");
    }
}
```

```java
public interface EgoeraIMC {
    public void Eskatu();
}

public class Txikia implements EgoeraIMC {
    public void Eskatu() {
        System.out.println("IMC indize txikia -> dieta hiperkalorikoa.");
    }
}
```

```java
public class ProbaPA {
    public static void main(String[] args) {
        FitBit fb = FitBit.getFitBit();
        //Egoera1
        fb.egoerAldatu((float)68.3, "normala", (float)1.68);
        fb.DietaEskatu();
        fb.KirolaEskatu();
        //Egoera2
        fb.egoerAldatu((float)90.3, "lasaia", (float)1.68);
        fb.DietaEskatu();
        fb.KirolaEskatu();
    }
}
```

EXEKUZIOAN

```
Zorionak!!! IMC indize ertaina daukazu!! Ez duzu dietarik behar!!
Forma normala daukazunez, ariketa plangintza normala egokitzen zaizu.
IMC indize handia daukazunez, dieta hipokalorikoa egokitzen zaizu.
Forma lasaia daukazu-> Ariketa plangintza lasaia
```