```
interface VendableParKilogramme {
    double vendre(double quantite);
}

interface VendableParPiece {
    double vendre(int quantite);
}

interface SusceptibleDeSolde {
    void lancerSolde(double pourcentage);
    void terminerSolde(double pourcentage);
}

interface Descriptible {
    void afficherDescription();
}

interface CalculateurDeRendement {
    double calculerRendement();
}

abstract class Article implements CalculateurDeRendement, Descriptible {
    protected double prixAchat;
    protected double prixVente;
    protected String nom;
    protected String fournisseur;

    public Article(double prixAchat, double prixVente, String nom, String fournisseur) {
        this.prixAchat = prixAchat;
        this.prixVente = prixVente;
        this.nom = nom;
```

```
        this.fournisseur = fournisseur;
    }

    @Override
    public double calculerRendement() {
        return (prixVente - prixAchat) / prixAchat;
    }

    @Override
    public void afficherDescription() {
        System.out.println("Nom: " + nom);
        System.out.println("Fournisseur: " + fournisseur);
        System.out.println("Prix d'achat: " + prixAchat);
        System.out.println("Prix de vente: " + prixVente);
        System.out.println("Rendement: " + calculerRendement());
    }
}

class ArticleElectromenager extends Article implements VendableParPiece, SusceptibleDeSolde {
    private int stock;

    public ArticleElectromenager(double prixAchat, double prixVente, String nom, String fournisseur) {
        super(prixAchat, prixVente, nom, fournisseur);
        this.stock = 0;
    }

    public double remplirStock(int quantite) {
        stock += quantite;
        return quantite * prixAchat;
    }
```

```
    @Override
    public double vendre(int quantite) {
        if (quantite > stock) {
            System.out.println("Stock insuffisant !");
            return 0;
        }
        stock -= quantite;
        return quantite * prixVente;
    }

    @Override
    public void lancerSolde(double pourcentage) {
        prixVente -= prixVente * pourcentage / 100;
    }

    @Override
    public void terminerSolde(double pourcentage) {
        prixVente += prixVente * pourcentage / 100;
    }
}

class ArticlePrimeur extends Article implements VendableParKilogramme {
    private double stock;

    public ArticlePrimeur(double prixAchat, double prixVente, String nom, String fournisseur) {
        super(prixAchat, prixVente, nom, fournisseur);
        this.stock = 0;
    }

    public double remplirStock(double quantite) {
        stock += quantite;
```

```
        return quantite * prixAchat;
    }

    @Override
    public double vendre(double quantite) {
        if (quantite > stock) {
            System.out.println("Stock insuffisant !");
            return 0;
        }
        stock -= quantite;
        return quantite * prixVente;
    }

    @Override
    public void afficherDescription() {
        super.afficherDescription();
        System.out.println("Stock (kg): " + stock);
    }
}

class Magasin {
    private double depenses;
    private double revenus;
    private Article[] electromenagers;
    private Article[] primeurs;
    private int indexElectromenagers;
    private int indexPrimeurs;

    public Magasin() {
        this.electromenagers = new Article[10];
        this.primeurs = new Article[10];
```

```
    this.indexElectromenagers = 0;
    this.indexPrimeurs = 0;
    this.depenses = 0;
    this.revenus = 0;
}

public void ajouterElectromenager(ArticleElectromenager article) {
    if (indexElectromenagers < 10) {
        electromenagers[indexElectromenagers++] = article;
    }
}

public void ajouterPrimeur(ArticlePrimeur article) {
    if (indexPrimeurs < 10) {
        primeurs[indexPrimeurs++] = article;
    }
}

public void description() {
    System.out.println("Dépenses: " + depenses);
    System.out.println("Revenus: " + revenus);
    for (int i = 0; i < indexElectromenagers; i++) {
        electromenagers[i].afficherDescription();
    }
    for (int i = 0; i < indexPrimeurs; i++) {
        primeurs[i].afficherDescription();
    }
}

public double calculerRendement() {
    return (revenus - depenses) / depenses;
```

```
    }

    public void enregistrerDepense(double montant) {
        depenses += montant;
    }

    public void enregistrerRevenu(double montant) {
        revenus += montant;
    }
}

public class Main {
    public static void main(String[] args) {
        Magasin magasin = new Magasin();
        ArticleElectromenager frigo = new ArticleElectromenager(500, 700, "Frigo", "LG");
        ArticlePrimeur pomme = new ArticlePrimeur(2, 3, "Pomme", "FermeBio");

        magasin.ajouterElectromenager(frigo);
        magasin.ajouterPrimeur(pomme);

        double depenseFrigo = frigo.remplirStock(5);
        magasin.enregistrerDepense(depenseFrigo);

        double depensePomme = pomme.remplirStock(50);
        magasin.enregistrerDepense(depensePomme);

        double revenuFrigo = frigo.vendre(2);
        magasin.enregistrerRevenu(revenuFrigo);

        double revenuPomme = pomme.vendre(10);
        magasin.enregistrerRevenu(revenuPomme);
```

```
        magasin.description();
        System.out.println("Rendement: " + magasin.calculerRendement());
    }
}
```