

TP - Création d'un composant logiciel en Java - CORRIGÉ

Classe COMPTEUR

```
package sio.seg.metier;

/**
 * Classe Compteur
 * @author sophie
 */
public class Compteur {
    private int numero;
    private char typeCompteur; // A pour abonné, V pour Vanne
    private int indexPrecedent; // Index avant le relevé
    private int indexNouveau; // Index après le relevé

    /**
     * Constructeur de la classe Compteur
     * @param numero      numéro du compteur
     * @param typeCompteur type du compteur
     */
    public Compteur(int numero, char typeCompteur) {
        this.numero = numero;
        this.typeCompteur = typeCompteur;
        this.indexPrecedent=0;
        this.indexNouveau=0;
    }

    /**
     * @return numero du compteur
     */
    public int getNumero() {
        return this.numero;
    }

    /**
     * @return type du compteur
     */
    public char getTypeCompteur() {
        return this.typeCompteur;
    }

    /**
     * Met à jour les index avant et après le relevé
     * @param indexReleve Index relevé sur le compteur
     */
    public void releve(int indexReleve) {
        this.indexPrecedent = this.indexNouveau;
        this.indexNouveau = indexReleve;
    }

    /**
     * Calcule et retourne la consommation en m3 pour le compteur
     * @return consommation en m3
     */
    public int conso() {
        return this.indexNouveau-this.indexPrecedent;
    }
}
```

Classe SECTEUR

```
package sio.seg.metier;
import java.util.ArrayList;
/**
 * Classe Secteur (un secteur géographique correspond à un quartier)
 * @author sophie
 */
public class Secteur {
    private int numSecteur;
    private String nomSecteur; // nom du quartier
    private boolean espaceVert; // indique la présence ou non dans le quartier d'un espace
                                // vert municipal à arroser

    private Commune laCommune;
    private ArrayList<Compteur> lesCompteurs; // liste des compteurs du secteur

    /**
     * Constructeur de la classe Secteur
     * @param numSecteur numéro du secteur
     * @param nomSecteur nom du quartier
     * @param espaceVert indique espace vert ou non
     * @param laCommune commune à laquelle est rattachée le secteur
     */
    public Secteur(int numSecteur, String nomSecteur, boolean espaceVert, Commune laCommune) {
        this.numSecteur = numSecteur;
        this.nomSecteur = nomSecteur;
        this.espaceVert = espaceVert;
        this.laCommune = laCommune;
        this.lesCompteurs = new ArrayList<Compteur>();
    }

    /**
     * @return numéro du secteur
     */
    public int getNumSecteur() {
        return this.numSecteur;
    }

    /**
     * @return nom du secteur
     */
    public String getNomSecteur() {
        return this.nomSecteur;
    }

    /**
     * @return booléen qui indique la présence ou non d'un espace vert
     */
    public boolean isEspaceVert() {
        return this.espaceVert;
    }

    /**
     * @return liste des compteurs du secteur
     */
    public ArrayList<Compteur> getLesCompteurs() {
        return this.lesCompteurs;
    }

    /**
     * Ajoute un nouveau compteur au secteur
     * @param compteur Compteur à ajouter au secteur
     */
    public void ajouterCompteur(Compteur compteur) {
        this.lesCompteurs.add(compteur);
    }
}
```

Classe COMMUNE

```
package sio.seg.metier;
import java.util.ArrayList;

/**
 * Classe Commune
 */
public class Commune {
    private int numCom;
    private String nomCom;
    private ArrayList<Secteur> lesSecteurs;

    /**
     * Constructeur de la classe Commune
     * @param numCom      numéro de la commune
     * @param nomCom      nom de la commune
     */
    public Commune(int numCom, String nomCom) {
        this.numCom = numCom;
        this.nomCom = nomCom;
        this.lesSecteurs = new ArrayList<Secteur>();
    }

    /**
     * Crée le secteur et l'ajoute à la commune
     * @param numeroSecteur  numéro du secteur
     * @param nomSecteur    nom du secteur
     * @param espaceVert    booléen indiquant si le secteur possède un espace vert ou non
     */
    public void ajouteSecteur(int numeroSecteur, String nomSecteur, boolean espaceVert) {
        Secteur leSecteur = new Secteur(numeroSecteur, nomSecteur, espaceVert, this);
        this.lesSecteurs.add(leSecteur);
    }

    /**
     * Retourne le secteur dont le numero est passé en paramètre ou null si le numéro du secteur
     * n'existe pas
     * @param numero      numéro du secteur
     * @return            objet secteur ou null
     */
    public Secteur getSecteurByNumero(int numero) {
        Secteur leSecteur = null;
        for (Secteur s : this.lesSecteurs)
            if (s.getNumSecteur()==numero)
                leSecteur = s;
        return leSecteur;
    }

    /**
     * Retourne la liste des secteurs de la commune qui possèdent un espace vert
     * @return            Liste des secteurs qui possèdent un espace vert
     */
    public ArrayList<Secteur> secteursEVerts() {
        ArrayList<Secteur> lesSecteursVerts = new ArrayList<Secteur>();
        for (Secteur s : this.lesSecteurs) {
            if (s.isEspaceVert())
                lesSecteursVerts.add(s);
        }
        return lesSecteursVerts;
    }
}
```

```

/**
 * Retourne le volume distribué par les compteurs de la commune
 * @param unTypeCompteur    type de compteur V pour Vanne ou A pour Abonne
 * @return    volume en m3
 */
public int volumeDistribue(char unTypeCompteur) {
    int volume = 0;
    for (Secteur s : this.lesSecteurs) {
        for (Compteur c : s.getLesCompteurs()) {
            if (c.getTypeCompteur()== unTypeCompteur)
                volume = volume + c.conso();
        }
    }
    return volume;
}

/**
 * Retourne la différence entre le volume total distribué par les vannes et la consommation
   des abonnés
 * @return    Volume de perte en m3
 */
public int perte() {
    int volAbonnes = this.volumeDistribue('A');
    int volVannes = this.volumeDistribue('V');
    return volVannes - volAbonnes;
}

/**
 * Calcule le pourcentage des pertes par rapport au volume distribué par les vannes et
   retourne 1, 2 ou 3
 * @return    1 --> pourcentage < à 10%
 *            2 --> pourcentage entre 10 et 15% inclus
 *            3 --> pourcentage > à 15%
 */
public int anomalie() {
    int indicateur = 0;
    double pourcentage = (double) this.perte() / (double) this.volumeDistribue('V');
    if (pourcentage <= 0.1)
        indicateur = 1;
    else
        if (pourcentage <= 0.15)
            indicateur = 2;
        else
            indicateur = 3;
    return indicateur;
}
}

```

Classe TestProgram

```
package sio.seg.metier;
import java.util.ArrayList;

public class TestProgram {
    public static void main(String[] args) {
        Commune vienne = new Commune(1, "Vienne");
        vienne.ajouteSecteur(1, "Centre", true);
        vienne.ajouteSecteur(2, "L'Isle", false);
        vienne.ajouteSecteur(3, "Saint-Martin", true);

        Compteur c1 = new Compteur(1, 'V');
        Compteur c2 = new Compteur(2, 'A');
        Compteur c3 = new Compteur(3, 'A');
        Compteur c4 = new Compteur(4, 'A');

        c1.releve(2500);
        c2.releve(1000);
        c3.releve(400);
        c4.releve(770);

        vienne.getSecteurByNumero(1).ajouterCompteur(c1);
        vienne.getSecteurByNumero(1).ajouterCompteur(c2);
        vienne.getSecteurByNumero(1).ajouterCompteur(c3);
        vienne.getSecteurByNumero(1).ajouterCompteur(c4);

        ArrayList<Secteur> lesSecteursV = new ArrayList<Secteur>();
        lesSecteursV = vienne.secteursEverts();
        System.out.println(lesSecteursV.size());

        System.out.println("Volume vannes = " + vienne.volumeDistribue('V'));
        System.out.println("Volume perte = " + vienne.perte());
        System.out.println("Anomalie = " + vienne.anomalie());
    }
}
```