

### Problème

```
public interface Statistique {
    public int getNbIns();
    public float getMoyenneNotes();
    public float getGainVentes();
}

public abstract class AvisUtilisateur {
    public abstract void afficher();
}

public class Commentaire extends AvisUtilisateur {
    private String texte;
    public Commentaire(String texte) {
        this.texte = texte;
    }
    @Override
    public void afficher() {
        System.out.println("Commentaire: "+texte);
    }
}

public class Evaluation extends AvisUtilisateur {
    private float note;
    public Evaluation(float note) {
        super();
        this.note = note;
    }
    @Override
    public void afficher() {
        System.out.println("Note: " + note);
    }
}

public class AppMobile implements Statistique {
    public static final int MAX_AVIS=100;
    private String nom;
    private String theme;
    private String developpeur;
    private String systExp;
    private float prix;
    private int nbIns;
    private int nbAvis;
    private AvisUtilisateur[] tAvis;
    public AppMobile(String nom, String theme, String developpeur, String systExp, float
prix) {
        this.nom = nom;
        this.theme = theme;
        this.developpeur = developpeur;
        this.systExp = systExp;
        this.prix = prix;
        nbIns = 0;
        nbAvis=0;
        tAvis = new AvisUtilisateur[MAX_AVIS];
    }
    public void afficher() {
        System.out.println("Application Mobile:");
        System.out.println("Nom: " + nom);
        System.out.println("Theme: " + theme);
        System.out.println("Developpeur: " + developpeur);
        System.out.println("Système d'exploitation: " + systExp);
        System.out.println("Prix vente: " + prix);
    }
    public void installer() {
        System.out.println("Installation...");
        nbIns++;
    }
    public void ajouter(AvisUtilisateur avis) {
        if(nbAvis<tAvis.length) {
            tAvis[nbAvis]=avis;
            nbAvis++;
        }
    }
    public void afficherCommentaires() {
        boolean aucun = true;
        for (int i = 0; i < nbAvis; i++) {
            if (tAvis[i] instanceof Commentaire) {
                aucun = false;
                tAvis[i].afficher();
            }
        }
        if (aucun)
            System.out.println("Aucun commentaire.");
    }
    @Override
    public int getNbIns() {
        return nbIns;
    }
    @Override
    public float getMoyenneNotes() {
        float somme = 0;
        int nbNote = 0;
        float moyenne = -1;
        for (int i = 0; i < nbAvis; i++) {
            if (tAvis[i] instanceof Evaluation) {
                Evaluation e = (Evaluation) tAvis[i];
                somme += e.getNote();
            }
        }
        if (nbNote > 0)
            moyenne = somme / nbNote;
        return moyenne;
    }
    @Override
    public float getGainVentes() {
        return nbIns * prix;
    }
}
```



```
nbIns = 0;
nbAvis=0;
tAvis = new AvisUtilisateur[MAX_AVIS];
}
public void afficher() {
    System.out.println("Application Mobile:");
    System.out.println("Nom: " + nom);
    System.out.println("Theme: " + theme);
    System.out.println("Developpeur: " + developpeur);
    System.out.println("Système d'exploitation: " + systExp);
    System.out.println("Prix vente: " + prix);
}
public void installer() {
    System.out.println("Installation...");
    nbIns++;
}
public void ajouter(AvisUtilisateur avis) {
    if(nbAvis<tAvis.length) {
        tAvis[nbAvis]=avis;
        nbAvis++;
    }
}
public void afficherCommentaires() {
    boolean aucun = true;
    for (int i = 0; i < nbAvis; i++) {
        if (tAvis[i] instanceof Commentaire) {
            aucun = false;
            tAvis[i].afficher();
        }
    }
    if (aucun)
        System.out.println("Aucun commentaire.");
}
@Override
public int getNbIns() {
    return nbIns;
}
@Override
public float getMoyenneNotes() {
    float somme = 0;
    int nbNote = 0;
    float moyenne = -1;
    for (int i = 0; i < nbAvis; i++) {
        if (tAvis[i] instanceof Evaluation) {
            Evaluation e = (Evaluation) tAvis[i];
            somme += e.getNote();
        }
    }
    if (nbNote > 0)
        moyenne = somme / nbNote;
    return moyenne;
}
@Override
public float getGainVentes() {
    return nbIns * prix;
}
}
```



```

public class Store implements Statistique {
    private String nom;
    private String url;
    private String systExp;
    private Vector<AppMobile> vApp;

    public Store(String nom, String url, String systExp) {
        this.nom = nom;
        this.url = url;
        this.systExp = systExp;
        vApp = new Vector<AppMobile>();
    }

    public void ajouter(AppMobile app) {
        if (app.getSystExp().equals(systExp))
            vApp.addElement(app);
    }

    public int getNbApp() {
        return vApp.size();
    }

    public int getNbApp(String theme) {
        int nbApp = 0;
        for (int i = 0; i < vApp.size(); i++)
            if (vApp.elementAt(i).getTheme().equals(theme))
                nbApp++;
        return nbApp;
    }

    public void supprimer(int indice) {
        if (indice >= 0 && indice < vApp.size())
            vApp.removeElementAt(indice);
    }

    public void supprimer(String nom) {
        for (int i = 0; i < vApp.size(); i++)
            if (vApp.elementAt(i).getNom().equals(nom))
                vApp.removeElementAt(i);
    }

    public void supprimerTout() {
        vApp.removeAllElements();
    }

    public void afficher() {
        System.out.println("Store:");
        System.out.println("Nom: " + nom);
        System.out.println("URL: " + url);
        System.out.println("Système d'exploitation: " + systExp);
    }

    public void afficherApp(String theme) {
        boolean aucun = true;
        for (int i = 0; i < vApp.size(); i++)
            if (vApp.elementAt(i).getTheme().equals(theme)) {
                aucun=false;
                vApp.elementAt(i).afficher();
            }
        if(aucun)
            System.out.println("Aucune application pour ce thème.");
    }

    public void rechercher(String nom) {
        for (int i = 0; i < vApp.size(); i++)
            if (vApp.elementAt(i).getNom().equalsIgnoreCase(nom))
                vApp.elementAt(i).afficher();
    }
}

```



```

}

public void afficherAppPlusChere() {
    if (vApp.size() > 0) {
        AppMobile appPlusChere=vApp.elementAt(0);
        for (int i = 0; i < vApp.size(); i++)
            if (vApp.elementAt(i).getprix()>appPlusChere.getprix())
                appPlusChere=vApp.elementAt(i);

        System.out.println("Une des application les plus chères est:");
        appPlusChere.afficher();
    } else
        System.out.println("Le store ne contient aucune application.");
}

@Override
public int getNbIns() {
    int nb = 0;
    for (int i = 0; i < vApp.size(); i++)
        nb += vApp.elementAt(i).getNbIns();
    return nb;
}

@Override
public float getMoyenneNotes() {
    float somme = 0;
    int nb = 0;
    float moyenne = -1;
    for (int i = 0; i < vApp.size(); i++)
        if (vApp.elementAt(i).getMoyenneNotes() != -1) {
            somme += vApp.elementAt(i).getMoyenneNotes();
            nb++;
        }
    if (nb > 0)
        moyenne = somme / nb;
    return moyenne;
}

@Override
public float getGainVentes() {
    float somme = 0;
    for (int i = 0; i < vApp.size(); i++)
        somme += vApp.elementAt(i).getGainVentes();
    return somme;
}

}

public class TestStore {

    public static void main(String[] args) {
        Store s = new Store("Play Store", "https://play.google.com/store", "Android");
        s.afficher();

        AppMobile a1 = new AppMobile("Taquin", "Jeu", "3D+", "Android", 2);
        AppMobile a2 = new AppMobile("Puzzle", "Jeu", "3D+", "Android", 2);
        s.ajouter(a1);
        s.ajouter(a2);
        System.out.println("Nombre d'application dans le store: " + s.getNbApp());
        for (int i = 0; i < 30; i++)
            a1.installer();
        for (int i = 0; i < 10; i++)
            a2.installer();

        System.out.println("Gain Ventes App: " + a1.getGainVentes());
        System.out.println("Gain Ventes Store: " + s.getGainVentes());
    }

}

```

