

```

public class Date {
    Date(int an, int mois, int jour) {
        _an = an;
        _mois = mois;
        _jour = jour;
    }
    int getAn() {
        return _an;
    }
    int getJour() {
        return _jour;
    }
    int getMois() {
        return _mois;
    }
    void setAn(int a) {
        _an = a;
    }
    void setJour(int j) {
        _jour = j;
    }
    void setMois(int m) {
        _mois = m;
    }
    public Object clone() {
        return new Date(_an, _mois, _jour);
    }
    public String toString() {
        StringBuffer str = new StringBuffer();
        str.append(_jour + "/");
        str.append(_mois + "/");
        str.append(_an);
        return str.toString();
    }
    public boolean equals(Object o) {
        if (o instanceof Date) {
            Date d = (Date)o;
            return _jour == d._jour && _mois == d._mois && _an == d.
            _an;
        }
        return false;
    }
    private int _jour, _mois, _an;
}

```

```

public interface Media {
    String getTitre();
    String getAuteur();
    Date getDate();
    Object clone();
    String toString();
    boolean equals(Object o);
}

```

```

public class Disquev1 implements Media {

    public Disquev1(String titre, String auteur, Date date, int nbPistes){
        _titre = titre;
        _auteur = auteur;
        _date = (Date) date.clone();
        _nbPistes = nbPistes;
    }
    public String getTitre() {
        return _titre;
    }
    public String getAuteur() {
        return _auteur;
    }
    public Date getDate() {
        return (Date) _date.clone();
    }
    public int getNbPistes(){
        return _nbPistes;
    }
    public void setTitre(String titre) {
        _titre = titre;
    }
    public void setAuteur(String auteur) {
        _auteur = auteur;
    }
    public void setDate(Date date) {
        _date = (Date) date.clone();
    }
    public void setNbPistes(int nbPistes){
        _nbPistes = nbPistes;
    }
    public String toString(){
        return "Disque [Titre: " + _titre + " -Auteur: " + _auteur +
            " - Date de sortie " + _date+ " - Nb pistes: " + _nbPistes +]";
    }
    public Object clone(){
        return new Disquev1(_titre, _auteur, _date, _nbPistes);
    }
    public boolean equals(Object o){
        if (o instanceof Disquev1){
            Disquev1 d = (Disquev1) o;
            return _titre.equals(d._titre) && _auteur.equals(d._aute
ur) &&
                _date.equals(d._date)&& _nbPistes == d._nbPistes;
        }
        return false;
    }

    private String _titre;
    private String _auteur;
    private Date _date;
    private int _nbPistes;
}

```

```

public abstract class MediaAbstract implements Media {

    public MediaAbstract(String titre, String auteur, Date d){
        _titre = titre;
        _auteur = auteur;
        _date = (Date) d.clone();
    }

    public abstract Object clone();

    public String getTitre() {
        return _titre;
    }

    public String getAuteur() {
        return _auteur;
    }

    public Date getDate() {
        return (Date) _date.clone();
    }

    public void setTitre(String titre) {
        _titre = titre;
    }

    public void setAuteur(String auteur) {
        _auteur = auteur;
    }

    public void setDate(Date date) {
        _date = (Date) date.clone();
    }

    public String toString(){
        StringBuffer str = new StringBuffer();
        str.append("Titre: " + _titre);
        str.append("-Auteur: " + _auteur);
        str.append("- Date de sortie: " + _date);
        return str.toString();
    }

    public boolean equals(Object o){
        if (o instanceof MediaAbstract){
            MediaAbstract m = (MediaAbstract) o;
            return _titre.equals(m._titre) && _auteur.equals(m._aute
ur) && _date.equals(m._date);
        }
        return false;
    }

    private String _titre;
    private String _auteur;
    private Date _date;
}

```

```
public class Disque extends MediaAbstract {

    public Disque(String titre, String auteur, Date date, int nbPistes){
        super(titre, auteur, date);
        _nbPistes = nbPistes;
    }

    public int getNbPistes(){
        return _nbPistes;
    }

    public void setNbPistes(int nbPistes){
        _nbPistes = nbPistes;
    }

    public String toString(){
        StringBuffer str = new StringBuffer();
        str.append("Disque[").append(super.toString());
        str.append(" - Nb pistes: ").append(_nbPistes).append("]");
        return str.toString();
    }

    public Object clone(){
        return new Disque(getTitre(), getAuteur(), getDate(), _nbPistes)
;
    }

    public boolean equals(Object o){
        if (o instanceof Disque){
            Disque d = (Disque) o;
            return super.equals(d) && _nbPistes == d._nbPistes;
        }
        return false;
    }

    private int _nbPistes;
}
```

```
public class Livre extends MediaAbstract {

    public Livre(String titre, String auteur, Date date, int nbPages){
        super(titre, auteur, date);
        _nbPages = nbPages;
    }

    public int getNbPages(){
        return _nbPages;
    }

    public void setNbPages(int nbPages){
        _nbPages = nbPages;
    }

    public String toString(){
        StringBuffer str = new StringBuffer();
        str.append("Livre[").append(super.toString());
        str.append(" - Nb pages: ").append(_nbPages).append("]");
        return str.toString();
    }

    public Object clone(){
        return new Livre(getTitre(), getAuteur(), getDate(), _nbPages);
    }

    public boolean equals(Object o){
        if (o instanceof Livre){
            Livre d = (Livre) o;

            return super.equals(d) && _nbPages == d._nbPages;
        }
        return false;
    }

    private int _nbPages;
}
```

```

public class Mediatheque {
    public Mediatheque(){
        medias = new Media[TAILLE_INITIALE];
        indice = 0;
    }
    public void ajouter (Media m){
        if(tableauPlein())
            etendreTableau();
        medias[indice++] = (Media) m.clone();
    }

    public Media[] rechercherParAuteur(String auteur){
        int nbOccAuteur = 0;
        for(int i = 0; i < indice; ++i)
            if (medias[i].getAuteur().equals(auteur))
                nbOccAuteur ++;

        Media[] res = new Media[nbOccAuteur];
        int k = 0;
        for(int i = 0; i < indice; ++i)
            if (medias[i].getAuteur().equals(auteur))
                res[k++] = (Media) medias[i].clone();

        return res;
    }

    public boolean rechercherMedia (String titre){
        for(int i = 0; i < indice; ++i)
            if (medias[i].getTitre().equals(titre))
                return true;

        return false;
    }

    public String toString(){
        StringBuffer str = new StringBuffer();
        for(int i = 0; i < indice; ++i)
            str.append(medias[i] + "\n");
        return str.toString();
    }
    private boolean tableauPlein(){
        return indice == medias.length;
    }

    private void etendreTableau() {
        int nouvelleTaille = medias.length * 2;
        Media[] nMedias = new Media[nouvelleTaille];
        for(int i=0; i< medias.length; i++)
            nMedias[i] = medias[i];
        medias = nMedias;
    }

    private static final int TAILLE_INITIALE = 10;
    private int indice;
    private Media[] medias;
}

```

```

public class TestMediatheque {
    public static void main (String[] args){
        Date d = new Date(2011, 05, 2);
        Media l1 = new Livre ("La prisonniÃre des Sargasses", "Jean Rhys", d, 239
, 1), 588);
        Media l2 = new Livre ("Fables", "Jean de La Fontaine", new Date(2000, 8
, 1), 588);

        Media d1 = new Disque ("Blabla", "DJ", d, 10);
        System.out.println("Disque 1:" + d1);
        //Disque 1 : Disque [Titre : Blabla - Auteur : DJ - Date de sort
ie : 2/5/2011 - Nb pistes : 10]

        System.out.println("Nombre de Pistes du disque 1 : " + ((Disque)d1).getNb
Pistes());
        //Nombre de Pistes du disque 1 : 10

        System.out.println("Date de sortie du livre 1:" + l1.getDate());
        //Date de sortie du livre 1 : 2/5/2011

        System.out.println(l2.equals(l2.clone())); //true

        Mediatheque md = new Mediatheque();
        md.ajouter(l1);
        md.ajouter(d1);
        md.ajouter(l2);

        System.out.print(md);

        System.out.println("Fables se trouve dans la mÃdiathÃque : " + md.recherch
erMedia("Fables"));
        //Fables se trouve dans la mÃdiathÃque : true

        Media[] res = md.rechercherParAuteur("Jean Rhys");
        for (int i = 0; i < res.length; ++i){
            System.out.println(res[i]);
        }

    }
}

```