

```

package com.example.elegestionfamillesmedandroid.modele;

import android.util.Log;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

public class Medicaments {
    // propriétés
    private Integer idMed;
    private String MedDepotLegal;
    private String MedNomCommercial;
    private String MedEffets;
    private String MedConteIndic;
    private Double MedPrixEchantillon;
    private Integer CodeFamille;

    /**
     * Constructeur null
     */
    public Medicaments() {
        this.idMed = null;
        this.MedDepotLegal = null;
        this.MedNomCommercial = null;
        this.MedEffets = null;
        this.MedConteIndic = null;
        this.MedPrixEchantillon = null;
        this.CodeFamille = null;
    }

    /**
     * Constructeur plein
     * @param lIdMedoc Entier
     * @param medDepotLegal chaîne
     * @param medNomCommercial chaîne
     * @param leCodeFamille chaîne
     * @param medPrixEchantillon Réel
     * @param medEffets chaîne
     * @param medConteIndic chaîne
     */
    public Medicaments(Integer lIdMedoc, String medDepotLegal, String
medNomCommercial, Integer leCodeFamille,
                        Double medPrixEchantillon, String medEffets, String
medConteIndic) {
        this.idMed = lIdMedoc;
        this.MedDepotLegal = medDepotLegal;
        this.MedNomCommercial = medNomCommercial;
        this.CodeFamille = leCodeFamille;
        this.MedPrixEchantillon = medPrixEchantillon;
        this.MedEffets = medEffets;
        this.MedConteIndic = medConteIndic;
    }

    /**
     * méthode permettant de vider un objet en mettant tout ses champs à
    null
     */
}

```

```

public void mettreNull() {
    this.idMed = null;
    this.MedDepotLegal = null;
    this.MedNomCommercial = null;
    this.CodeFamille = null;
    this.MedPrixEchantillon = null;
    this.MedEffets = null;
    this.MedConteIndic = null;
}

/**
 * retourne l'id d
 * @return Entier idMed
 */
public Integer getIdMed() {
    return idMed;
}

/**
 * retourne l'id du dépôt légal
 * @param idMed Entier
 */
public void setIdMed(Integer idMed) {
    this.idMed = idMed;
}

/**
 * retourne le Numéro de dépôt légal sélectionné
 * @return Chaîne MedDepotLegal
 */
public String getMedDepotLegal() {
    return this.MedDepotLegal;
}

/**
 * retourne le nom commercial du médicament sélectionné
 * @return Chaîne MedNomCommercial
 */
public String getMedNomCommercial() {
    return this.MedNomCommercial;
}

/**
 * retourne le code de la Famille sélectionné
 * @return Chaîne CodeFamille
 */
public Integer getCodeFamille() {
    return this.CodeFamille;
}

/**
 * retourne le prix de l'échantillon du médicament sélectionné
 * @return Chaîne MedComposition
 */
public Double getMedPrixEchantillon() {
    return this.MedPrixEchantillon;
}

/**
 * retourne les effets du médicament sélectionné
 * @return Chaîne MedEffets

```

```

*/
public String getMedEffets() {
    return this.MedEffets;
}

/**
 * retourne les contre-indication du médicament sélectionné
 * @return Chaîne MedConteIndic
 */
public String getMedConteIndic() {
    return this.MedConteIndic;
}

/**
 * instancier le dépôt légal du médicament sélectionné
 * @param medDepotLegal chaîne
 */
public void setMedDepotLegal(String medDepotLegal) {
    this.MedDepotLegal = medDepotLegal;
}

/**
 * instancier le nom commercial du médicament sélectionné
 * @param medNomCommercial chaîne
 */
public void setMedNomCommercial(String medNomCommercial) {
    this.MedNomCommercial = medNomCommercial;
}

/**
 * instancier le code de la Famille du médicament sélectionné
 * @param codeFamille chaîne
 */
public void setCodeFamille(Integer codeFamille) {
    this.CodeFamille = codeFamille;
}

/**
 * instancier le prix de l'échantillon du médicament sélectionné
 * @param prixEchantillon réel
 */
public void setMedPrixEchantillon(Double prixEchantillon) {
    this.MedPrixEchantillon = prixEchantillon;
}

/**
 * instancier les effets du médicament sélectionné
 * @param medEffets Chaîne
 */
public void setMedEffets(String medEffets) {
    this.MedEffets = medEffets;
}

/**
 * instancier les contre-indications du médicament sélectionné
 * @param medConteIndic chaîne
 */
public void setMedConteIndic(String medConteIndic) {
    this.MedConteIndic = medConteIndic;
}

```

```

/**
 * Conversion du médicament au Format JSON
 * @return
 */
public JSONArray convertToJsonArray() {
    List laListe = new ArrayList();
    laListe.add(idMed); // ajout du code Médicament à la liste
    laListe.add(MedDepotLegal); // ajout du libelle des Médicaments à
la liste
    laListe.add(MedNomCommercial); // ajout du nom commercial des
Médicaments à la liste
    laListe.add(CodeFamille); // ajout du code Famille à la liste
Médicament
    laListe.add(MedPrixEchantillon); // ajout d'un prix d'échantillon à
la liste Médicament
    laListe.add(MedEffets); // ajout de l'effet du Médicament à la
liste
    laListe.add(MedConteIndic); // ajout d'une contre indication au
Médicament à la liste
    return new JSONArray(laListe); // Conversion en JSONArray des
éléments composant l'objet Médicament dans la liste
}

/**
 * Méthode remplissant une ArrayList de Médicaments convertit dans
leurs format à partir d'un tableau JSON
 * @param jsonArray
 * @return lesMédicaments(liste)
 */
public ArrayList<Médicaments> getArrayListMedocFromJSONArray(JSONArray
jsonArray) {
    ArrayList<Médicaments> lesMédicaments = new
ArrayList<Médicaments>();
    try {
        for (int k = 0; k < jsonArray.length(); k++) {
            JSONObject info = new
JSONObject(jsonArray.get(k).toString());
            Integer idMed = info.getInt("idMed");
            String MedDepotLegal = info.getString("MED_DEPOTLEGAL");
            String MedNomCommercial =
info.getString("MED_NOMCOMMERCIAL");
            String MedEffets = info.getString("MED_EFFETS");
            String MedConteIndic =
info.getString("MED_CONTREINDICATION");
            Double MedPrixEchantillon =
info.getDouble("MED_PRIXECHANTILLON");
            Integer CodeFamille = info.getInt("FAM_ID");
            Médicaments medicament = new Médicaments(idMed,
MedDepotLegal, MedNomCommercial,
CodeFamille, MedPrixEchantillon, MedEffets,
MedConteIndic);
            lesMédicaments.add(medicament);
        }
    } catch (JSONException e) {
        Log.d("erreur", "conversion JSON impossible" + e.toString());
    }
    return lesMédicaments;
}
}

```