

TD02

Classes : DSI3

Matière : Développement Mobile

Nb pages : 2

Enseignants : S. Hadhri & M. Hadiji & H. Souissi

Problème

"TestCovid19" est une application android qui permet de gérer les tests Covid19 d'un ensemble de suspects. Un TestCovid19 est défini par un entier **type** (test de référence (0) ou test rapide (1)), une chaîne de caractère **CIN**, un entier **age**, un booléen **immunité faible** (true ou false), un booléen **maladie chronique** (true ou false) et une chaîne de caractère **résultat** ("Covid-" ou "Covid+"). De plus TestCovid19 possède une méthode `getInterprétation()` qui retourne "Sain", "Grave" ou "Très Grave".

La description des méthodes de la classe MainActivity est donnée par le tableau suivant :

Méthode	Description
init()	<ul style="list-style-type: none"> - Initialise tous les attributs graphique (spT...lstT) - Initialise adpT et de le lie à lstT - Appelle <code>effacer()</code> et <code>ajouterEcouteurs</code>
ajouterEcouteurs()	<ul style="list-style-type: none"> - Lorsque l'utilisateur clique sur btnA appeler <code>ajouter()</code> - Lorsque l'utilisateur clique sur btnE appeler <code>effacer()</code> - Lorsque l'utilisateur clique sur btnV appeler <code>vider()</code> - Lorsque l'utilisateur clique sur un item de lstT appeler <code>afficher(position)</code>
ajouter()	- Si edCIN et edAge ne sont pas vides, elle instancie un TestCovid19 avec les données choisies, l'ajoute à lstT et appelle <code>effacer()</code> , sinon elle affiche le message d'erreur toast1.
effacer()	<ul style="list-style-type: none"> - Sélectionne le premier élément dans spT - Efface edCIN - Efface edAge - Déchoche chIF - Déchoche chMC - Coche rdN
vider()	- Affiche <code>alertDialog1</code> et si l'utilisateur clique sur OUI elle vide lstT
afficher(int position)	- Affiche toast2 qui contient le CIN du suspect, le résultat et l'interprétation du TestCovid19 d'indice position dans lstT

Travail demandé

En utilisant les indications, Donner le code des méthodes `init()`, `ajouterEcouteurs()`, `ajouter()`, `effacer()`, `vider()` et `afficher(int position)`.

Indications

L'interface de l'activité MainActivity est la suivante :

The screenshot shows the 'Test Covid 19' app interface. The title bar is blue with the text 'Test Covid 19'. Below it, there is a form with several fields and buttons. Annotations with arrows point to specific elements:

- spT** points to the 'Type' dropdown menu, which currently shows 'Test de Référence'.
- edCIN** points to the 'CIN Suspect' text input field.
- edAge** points to the 'Age Suspect' text input field.
- chIF** points to the 'Immunité Faible' checkbox under 'Remarques'.
- chMC** points to the 'Maladie Chronique' checkbox under 'Remarques'.
- rdN** points to the 'Négatif' radio button under 'Résultat'.
- rdP** points to the 'Positif' radio button under 'Résultat'.
- rdgR** points to the 'Résultat' section.
- btnA** points to the 'AJOUTER' button.
- btnE** points to the 'EFFACER' button.
- btnV** points to the 'VIDER' button.
- lstT** points to the list of test results at the bottom.

On the right side, there is a sidebar with two buttons: 'Contenu de spT' and 'Test de Référence Test Rapide'.

Les deux Toasts de l'activité sont les suivants :

Two toast messages are shown:

- toast1** displays the message: 'Remplir tous les champs SVP!' (Fill in all fields please!).
- toast2** displays the message: 'CIN Suspect : 012345', 'Résultat : Covid19+', and 'Interprétation : Très Grave'.

Le AlertDialog de l'activité est le suivant :

An AlertDialog dialog box is shown with the title 'Vider?' and the message 'Etes vous sûr de vider la liste?'. It has two buttons: 'NON' and 'OUI'.

Le code de la classe TestCovid19 est le suivant :

```
public class TestCovid19 {
    private int type;        // 0 : Test de référence
                             // 1 : Test rapide

    private String cin;
    private int age;
    private boolean imF; // false: immunité normale
                        // true : immunité faible
    private boolean maC; // false: pas de maladie chronique
                        // true : présence d'une maladie chronique
    private String res; // "Covid19-" ou "Covid19+"

    public TestCovid19(int type, String cin, int age, boolean imF, boolean maC,
                       String res) {
        this.type = type;
        this.cin = cin;
        this.age = age;
        this.imF = imF;
        this.maC = maC;
        this.res = res;
    }

    public int getType() {
        return type;
    }

    public String getCin() {
        return cin;
    }

    public int getAge() {
        return age;
    }

    public boolean isImF() {
        return imF;
    }

    public boolean isMaC() {
        return maC;
    }

    public String getRes() {
        return res;
    }

    public String getInterpretation() {
        if (res.equalsIgnoreCase("Covid19+")) {
            if (maC || imF)
                return "Très Grave";
            return "Grave";
        }
        return "Sain";
    }

    @Override
    public String toString() {
        String strType = "";
        if (type == 0) strType = "Référence";
        else strType = "Rapide";
        return strType + ": " + cin + "(" + age + ")" +
            " -> " + getRes();
    }
}
```

Le code de la classe MainActivity est le suivant :

```
public class MainActivity extends AppCompatActivity {  
    private Spinner spT;  
    private EditText edCIN;  
    private EditText edAge;  
    private CheckBox chIF;  
    private CheckBox chMC;  
    private RadioGroup rdgR;  
    private RadioButton rdN;  
    private Button btnA;  
    private Button btnE;  
    private Button btnV;  
    private ListView lstT;  
    private ArrayAdapter<TestCovid19> adpT;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        init();  
    }  
    private void init() { ... }  
    private void ajouterEcouteurs() { ... }  
    private void ajouter() { ... }  
    private void effacer() { ... }  
    private void vider() { ... }  
    private void afficher(int position) { ... }  
}
```

Pour programmer l'évènement click sur un item d'un ListView

```
lst.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> parent, View view, int position, long id)  
    { }  
});
```

Pour afficher un AlertDialog

```
AlertDialog.Builder b = new AlertDialog.Builder(this);  
b.setTitle("Titre");  
b.setMessage("Message");  
b.setPositiveButton("Positif", new DialogInterface.OnClickListener() {  
    @Override  
    public void onClick(DialogInterface arg0, int arg1) { }  
});  
b.setNegativeButton("Négatif", new DialogInterface.OnClickListener() {  
    @Override  
    public void onClick(DialogInterface dialog, int which) {}  
});  
AlertDialog a = b.create();  
a.show();
```