



DEVOIR SURVEILLE

Classe : SEM21	Matière : Développement d'Applications Mobiles Natives	Nb pages : 4
Enseignant : Souissi Hamed		
Documents Non Autorisés	Barème : 20 = 5 + 15	Durée : 1 heure

Questions de cours

- 1- Donner deux caractéristiques d'un écran mobile et deux exemples de chaque caractéristique.
- 2- Donner trois noms de marques de processeur utilisés dans les appareils mobile.
- 3- Quels sont les trois types d'application mobile ?
- 4- Qu'est-ce qu'une application mobile native ?
- 5- Donner trois domaines d'applications mobiles.

Problème

"PrixVignette" est une application android qui permet de gérer les prix des vignettes des automobiles. Le prix des vignettes est calculé en fonction du type de l'automobile (Personne physique ou Personne morale), de sa puissance (son nombre de chevaux) et de son carburant (Essence, Diesel, GPL).

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 (edM...lstA), donner seulement l'initialisation de edM et de lstA- Initialise adpA et de le lie à lstA- Appelle effacer() et ajouterEcouleurs()
ajouterEcouleurs()	<ul style="list-style-type: none">- Lorsque l'utilisateur clique sur btnA appeler ajouter()- Lorsque l'utilisateur clique sur btnE appeler effacer()- Lorsque l'utilisateur clique sur btnV appeler vider()- Lorsque l'utilisateur clique sur un item de lstA appeler afficher(position)
ajouter()	<ul style="list-style-type: none">- Si edM n'est pas vide, elle instancie une automobile avec les données choisies, l'ajoute à lstA et appelle effacer(), sinon elle affiche le message d'erreur toast1.
effacer()	<ul style="list-style-type: none">- Efface edM- Sélectionne le premier élément dans spT- Met la valeur 4 dans seekP- Coche rdE- Met le curseur dans edM
vider()	<ul style="list-style-type: none">- Affiche alertDialog1 et si l'utilisateur clique sur OUI elle vide lstA
afficher(int position)	<ul style="list-style-type: none">- Affiche toast2 qui contient la matricule et le prix des vignettes de l'automobile d'indice position dans lstA

Travail demandé

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

Indications

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

The screenshot shows the 'Prix Vignette Automobile' app interface. The title bar is blue with the text 'Prix Vignette Automobile'. Below the title bar, there are four input fields: 'Matricule' (with a red underline), 'Type' (set to 'Personne physique'), 'Puissance' (with a red slider), and 'Carburant' (with three radio buttons: 'Essence' (selected), 'Diesel', and 'GPL'). To the right of these fields are several boxes with arrows pointing to them: 'edM' points to 'Matricule', 'spT' points to 'Type', 'seekP' points to 'Puissance', 'rdgC' points to 'Carburant', 'rdE' points to 'Essence', 'rdD' points to 'Diesel', 'rdG' points to 'GPL', 'btnE' points to the 'AJOUTER' button, and 'btnV' points to the 'VIDER' button. Below the input fields are three buttons: 'AJOUTER', 'EFFACER', and 'VIDER'. To the left of these buttons is a box labeled 'btnA' with an arrow pointing to 'AJOUTER'. To the right of these buttons is a box labeled 'lstA' with an arrow pointing to the list below. The list contains three items: '44 TUN 77{Phy- 4CH- Essence- 60DT}', '45 TUN 77{Phy- 4CH- Diesel- 210DT}', and '46 TUN 77{Phy- 4CH- GPL- 385DT}'. To the right of the list is a box labeled 'Contenu de spT' with an arrow pointing to the list. Below the list is a box labeled 'Personne physique' and 'Personne morale' with an arrow pointing to the list.

Les deux Toasts de l'activité sont les suivants :

The first toast message is labeled 'toast1' and contains the text 'Tapez une matricule SVP!'. The second toast message is labeled 'toast2' and contains the text 'Matricule: 44 TUN 77' and 'Prix vignette: 60'.

Le AlertDialog de l'activité est le suivant :

The AlertDialog dialog box is titled 'Vider?' and contains the text 'Etes vous sûr de vider la liste?'. It has two buttons: 'NON' and 'OUI'.

Le code de la classe Automobile est le suivant :

```
public class Automobile {  
    private static final int[][] TARIF_PHYSIQUE={{60,210,385},{120,270,445}};  
    private static final int[][] TARIF_MORALE={{120,270,445},{240,390,565}};  
    private String matricule;  
    private int type;    // 0 : Personne physique  1 : Personne morale  
    private int puissance;  
    private String carburent;  
  
    public Automobile(String matricule, int type, int puissance, String carburent) {  
        this.matricule = matricule;  
        this.type = type;  
        this.puissance = puissance;  
        this.carburent = carburent;  
    }  
  
    public String getMatricule() {        return matricule;        }  
    public int getType()            {        return type;            }  
    public int getPuissance()       {        return puissance;       }  
    public String getCarburent() {        return carburent;        }  
    public int getPrix() {  
        int iP=0;  
        int iC=0;  
        iP=(puissance<=4)?0:1;  
        if(carburent.equalsIgnoreCase("Essence"))  
            iC=0;  
        else if(carburent.equalsIgnoreCase("Diesel"))  
            iC=1;  
        else  
            iC=2;  
        if (type == 0)  
            return TARIF_PHYSIQUE[iP][iC];  
        else  
            return TARIF_MORALE[iP][iC];  
    }  
    @Override  
    public String toString() {  
        String strType = "";  
        strType=(type==0)?"Phy":"Mor";  
        return matricule + "{" + strType + "- " + puissance + "CH"+  
            "- " + carburent + "- " + getPrix() + "DT"+ "}'";  
    }  
}
```

Le code de la classe MainActivity est le suivant :

```
public class MainActivity extends AppCompatActivity {  
    private EditText edM;  
    private Spinner spT;  
    private SeekBar seekP;  
    private RadioGroup rdgC;  
    private RadioButton rdE;  
    private Button btnA;  
    private Button btnE;  
    private Button btnV;  
    private ListView lstA;  
    private ArrayAdapter<Automobile> adpA;  
  
    @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 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();
```