



DEVOIR SURVEILLE

Classe : SEM21	Matière : Développement d'Applications Mobiles Natives	Nb pages : 4
Enseignant : Souissi Hafedh		
Documents Non Autorisés	Barème : 20 = 6 + 14	Durée : 1 heure

Questions de cours

- 1- Donner deux caractéristiques d'un écran mobile et deux exemples de chaque caractéristique.
- 2- Quelle est la valeur moyenne de la capacité de la batterie d'un smartphone ? quelle est son unité ?
- 3- Quels sont les trois processeurs standards vendus par ARM ? quel est le processeur le plus performant et quel est le processeur qui consomme moins d'énergie que les autres ?
- 4- Qu'est-ce qu'une application mobile native ?

Problème

"VenteSmartphone" est une application android qui permet de gérer la vente des smartphones. Un Smartphone est défini par une **marque**, un **prix**, un **processeur** ("A7" ou "A9" ou "A15"), un **nombre de caméras**, un booléen appelé **noir** (true : noir ou false : blanc), un booléen appelé **ecouteur** (true : avec écouteur ou false : sans écouteur) et un booléen appelé **cage** (true : avec cage ou false : sans cage). De plus Smartphone possède les méthodes getMarque(), getPrix() et getProcesseur(). Le classe Smartphone est donnée dans la page3.

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 graphiques, donner seulement :<ul style="list-style-type: none">o l'initialisation de edMo l'initialisation de lstS- Initialise adpS et le lie à lstS- Appelle effacer() et ajouterEcouteurs()
ajouterEcouteurs()	<ul style="list-style-type: none">- Définit tous les événements de l'activité : donner seulement les deux événements :<ul style="list-style-type: none">o Lorsque l'utilisateur clique sur btnC appeler classifier()o Lorsque l'utilisateur clique sur un item de lstS appeler afficher(position)
classifier()	<ul style="list-style-type: none">- Si edP n'est pas vides elle affiche toast2 qui donne une classification du Smartphone par rapport à son prix (prix<500 : Entrée de gamme, 500<prix<1000 : Milieu de gamme, prix>1000 : Haut de gamme)- sinon elle affiche le message d'erreur toast1.
ajouter()	<ul style="list-style-type: none">- Si edM et edP ne sont pas vides elle instancie un Smartphone avec les données choisies, l'ajoute à adpS et appelle effacer()- sinon elle affiche le message d'erreur toast3.
effacer()	<ul style="list-style-type: none">- Efface edM et edP- Sélectionne le premier élément dans spP- Met le nombre de caméras à 1- Coche rdN (Noir)- Décoche chE et chC- Met le curseur dans edM

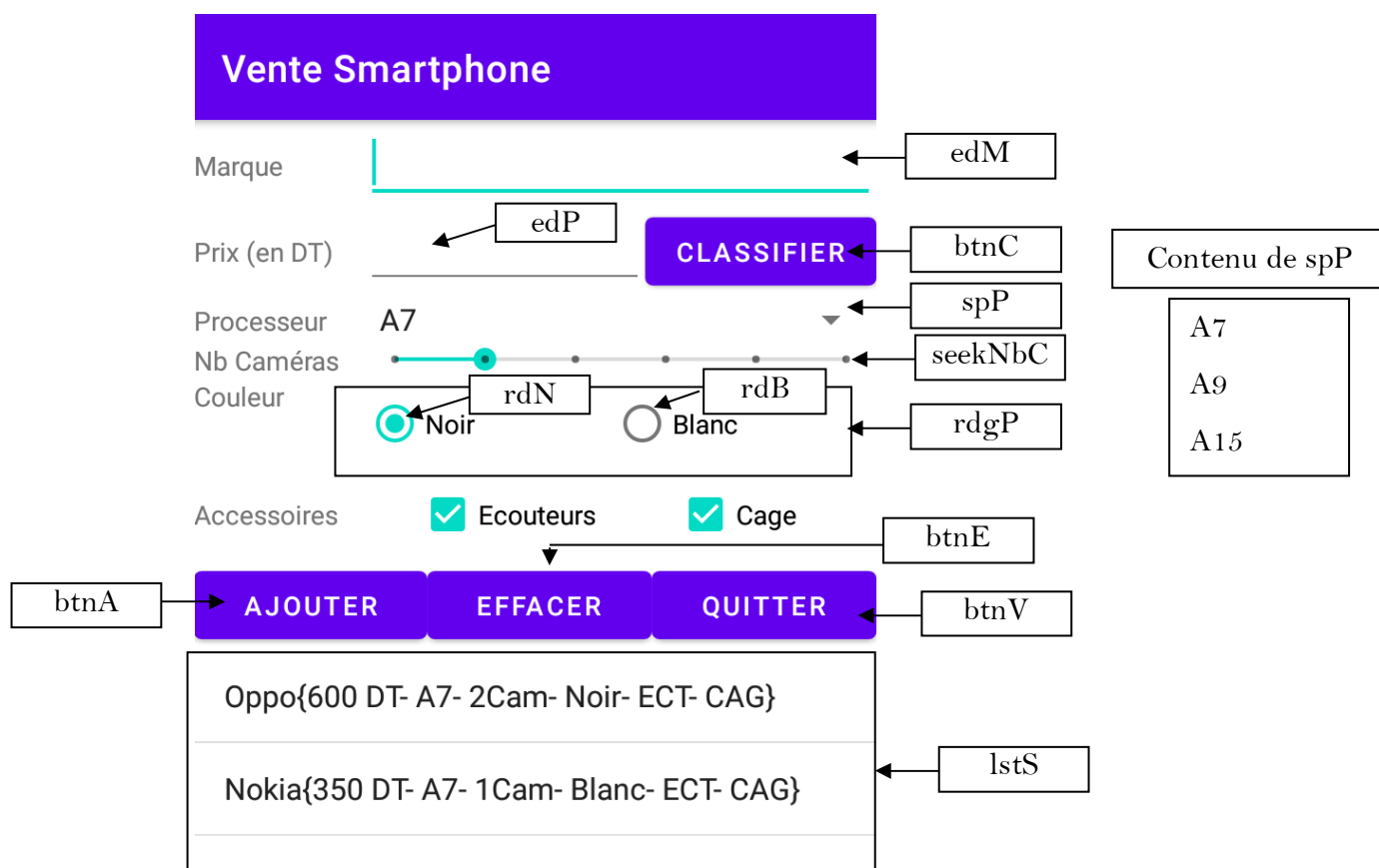
quitter()	- Ferme l'application
afficher(int position)	- Affiche toast4 qui contient la marque, le prix et le processeur du Smartphone d'indice position dans lstS

Travail demandé

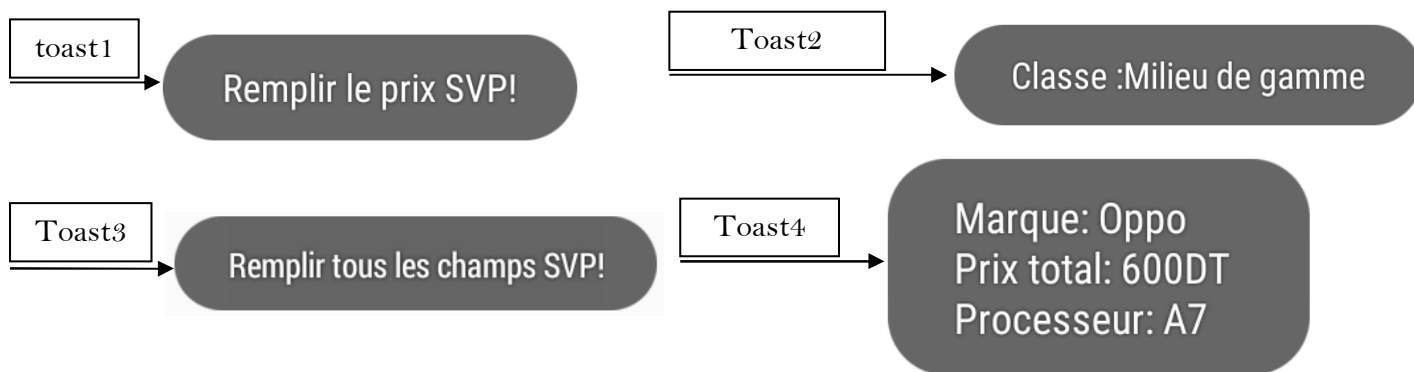
En utilisant les indications, Donner le code des méthodes `init()`, `ajouterEcouleurs()`, `classifier()`, `ajouter()`, `effacer()`, `quitter()` et `afficher(int position)`.

Indications

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



Les quatre Toasts de l'activité sont les suivants :



Le code de la classe Smartphone est le suivant :

```
public class Smartphone {
    private String marque;
    private int prix;
    private String processeur; // Cortex A7-Cortex A9-Cortex A15
    private int nbCamera;
    private boolean estNoir;    // true: Noir    false:Blanc
    private boolean ecouteur;   // true: avec écouteurs false: sans écouteurs
    private boolean cage;       // true: avec cage false: sans cage

    public Smartphone(String marque, int prix, String processeur, int nbCamera,
        boolean estNoir, boolean ecouteur, boolean cage) {
        this.marque = marque;
        this.prix = prix;
        this.processeur = processeur;
        this.nbCamera = nbCamera;
        this.estNoir = estNoir;
        this.ecouteur = ecouteur;
        this.cage = cage;
    }
    public String getMarque() {
        return marque;
    }
    public int getPrix() {
        return prix;
    }
    public String getProcesseur() {
        return processeur;
    }
    public String strCouleur() {
        return (estNoir) ? "Noir" : "Blanc";
    }
    public String strEcouteurs() {
        return (ecouteur) ? "ECT" : "";
    }
    public String strCage() {
        return (cage) ? "CAG" : "";
    }
    @Override
    public String toString() {
        return marque + "{" +
            prix + " DT" +
            "- " + processeur +
            "- " + nbCamera + "Cam" +
            "- " + strCouleur() +
            "- " + strEcouteurs() +
            "- " + strCage() +
            '}'';
    }
}
```

Le code de la classe MainActivity est le suivant :

```
public class MainActivity extends AppCompatActivity {  
    private EditText edM;  
    private EditText edP;  
    private Button btnC;  
    private Spinner spP;  
    private SeekBar seekNbC;  
    private RadioGroup rdgC;  
    private RadioButton rdN;  
    private CheckBox chE;  
    private CheckBox chC;  
    private Button btnA;  
    private Button btnE;  
    private Button btnQ;  
    private ListView lstS;  
    private ArrayAdapter<Smartphone> adpS;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        init();  
    }  
  
    private void init() { ... }  
    private void ajouterEcouteurs() { ... }  
    private void classifieur() { ... }  
    private void ajouter() { ... }  
    private void effacer() { ... }  
    private void quitter() { ... }  
    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)  
    { }  
});
```