

TD1 [Applications Mobiles]

Année Universitaire : 2020/2021

Niveau d'Etude : 3^{ème} année Licence.

Exercice 1 : Vous désirez réaliser une application avec Android en utilisant l'environnement Eclipse :

1. Citer l'ensemble des éléments que vous devez installer.
 - *JDK : Java Development Kit (pour la compilation Java)*
 - *Eclipse + ADT (Android Development Tools) : IDE pour le développement Java + Plugin Android*
 - *Android SDK (Software Development Kit) : Pour la compilation Android et l'émulateur*

OU BIEN :

- *JDK+ANDROID STUDIO*

2. Que permet de faire le fichier AndroidManifest.xml ?

Le Manifest permet de stocker les informations sur l'application, comme les différentes activités, la version du SDK.

3. Dès le démarrage de notre activité, un message d'erreur apparaît. Donner deux endroits possibles (méthodes ou fichiers) où peut se trouver l'erreur.

- *la méthode onCreate*
- *le manifest*

4. Nous avons corrigé cette erreur, mais maintenant, le message d'erreur apparaît quand on clique sur le bouton. Donner un endroit où peut se trouver l'erreur.

Dans la méthode qui décrit le comportement du bouton.

Exercice 2 : Le diagramme suivant représente le cycle de vie d'une activité :

1. Donner les noms des méthodes manquantes (1), (2), (3) et (4).

(1) onCreate()

(2) onStart()

(3) onPause()

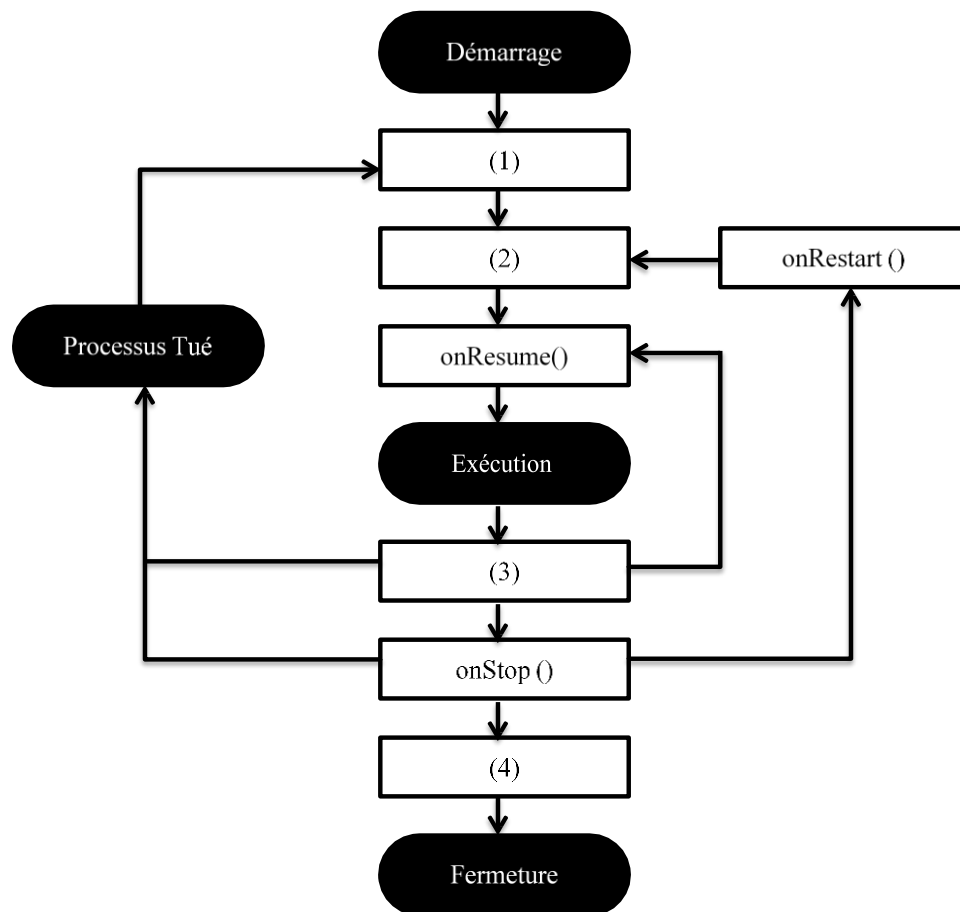
(4) onDestroy()

2. Quelle est la différence entre la méthode *onResume()* et la méthode numéro (2) ?

onStart() : Cette méthode est appelée quand l'application est démarrée.

onResume() : Cette méthode est appelée quand l'application passe (ou repasse) en avant-plan.

3. Quelle est la méthode qui doit être **obligatoirement** implémentée ?
onCreate()



TD2 [Applications Mobiles]

Niveau d'Etude : 3^{ème} année Licence.

Gestion de l'interface:

1. Dans quel répertoire doit-on créer le fichier qui représente l'interface ?

layout

2. Considérons le code xml suivant :

```
<Button
    android:id="@+id/bRecherche"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Recherche"
    android:onClick="recherche"/>
```

a. Donner l'entête de la méthode qui définit le comportement de ce bouton.

public void recherche(View v)

b. Soit l'instruction suivante:

```
Toast.makeText(MonActivity.this, "long clic!", 1000).show();
```

Nous souhaitons que ce Toast apparaisse quand on fait un **clic long** sur le bouton *b1*.
Quelle est l'instruction qui permet de faire cela?

```
b1.setOnLongClickListener(new OnLongClickListener(){
    public boolean onLongClick(View v){
        Toast.makeText(MonActivity.this, "long clic!", 1000).show();
    }
})
```

3. On aimerait créer un élément EditText.

a. Que permet de faire cet élément?

Un élément EditText permet à l'utilisateur de saisir des entrées.

b. Ecrire le code xml de cet élément, sachant qu'on désire lui donner l'identifiant "ed" et la même taille de son conteneur.

```
<EditText
    android:id="@+id/ed"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
```

c. On aimerait changer la couleur du texte de cet élément. Quel est l'attribut qu'on doit ajouter?

- colortext
- android:colorText
- android:textColor

4. Soit l'instruction suivante dans le code d'une activité :

```
pb = (ProgressBar) findViewById(R.id.progressBar) ;
```

a. Que permet de faire cette instruction ?

Cette instruction permet d'associer l'élément graphique progressBar à la variable Java pb

b. Quel est l'identifiant de cet élément graphique dans le fichier xml ?

progressBar

c. Quel est le type de cet élément graphique ?

ProgressBar

5. Quelle est la signification du signe « + » dans la déclaration android:id="@+id/un_id".

6. Préciser l'importance de fixer les permissions lors du développement de votre application?

Une permission Android autorise une activité d'accéder à certaines ressources matérielles, mais permet aussi à l'utilisateur de l'application de voir quelles permissions demande une application avant d'être installée.

TD3+4 [Applications Mobiles]

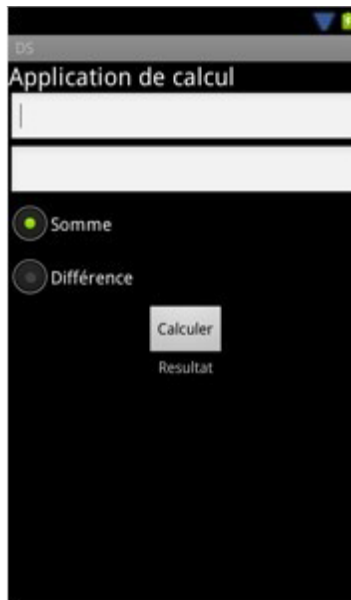
Niveau d'Etude : 3^{ème} année Licence.

Exercice1 Vous avez écrit le code suivant:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout (...)>
    <TextView
        android:id="@+id/titre"
        android:text="Application de calcul"/>
    <EditText
        android:id="@+id/vall"
        android:inputType="number"/>
    <EditText
        android:id="@+id/val2"
        android:inputType="number"/>
    <RadioGroup
        android:id="@+id/operations">
        <RadioButton
            android:id="@+id/somme"
            android:checked="true"
            android:text="Somme"/>
        <RadioButton
            android:id="@+id/diff"
            android:text="Différence"/>
    </RadioGroup>
    <Button
        android:id="@+id/calculer"
        android:layout_gravity="center"
        android:text="Calculer"
        android:onClick="calculer"/>
    <TextView
        android:id="@+id/res"
        android:layout_gravity="center"
        android:text="Resultat"/>
</LinearLayout>
```

Q1: Dessiner l'interface que vous obtenez avec ce code (l'emplacement des éléments et leurs étiquettes sont pris en considération).

R1:



Q2: A quoi sert l'élément **RadioGroup**?

R2 : RadioGroup permet de rassembler les RadioButtons dans un groupe, pour les associer et permettre qu'un seul bouton radio soit coché à la fois.

Q3: A quoi sert l'attribut **android:layout_gravity="center"**?

R3 : Cet attribut permet de centrer l'élément graphique auquel il est associé.

Q4 : Donner le code JAVA permettant de récupérer les valeurs saisies par l'utilisateur, qui calcule leur somme et l'affiche comme résultat.

Corrigé TD5

a. Soit la méthode suivante :

```
public void onCreateContextMenu(ContextMenu menu, View v,
    ContextMenuInfo menuInfo) {

    super.onCreateContextMenu(menu, v, menuInfo);
    menu.add(1, 1, 3, "Ajouter");
    menu.add(1, 2, 2, "Supprimer");
    menu.add(1, 3, 0, "Modifier");
    menu.add(1, 4, 1, "Quitter");
}
```

1. Définir l'élément graphique auquel elle est associée.

Cette méthode est associée à un menu contextuel.

Un menu contextuel est un menu flottant, associé à un événement sur un élément graphique de l'interface.

2. Que permet de faire cette méthode ? Dessiner son résultat.

Cette méthode permet d'ajouter quatre éléments au menu : Ajouter, supprimer, modifier et quitter.

Modifier
Quitter
Supprimer
Ajouter

b. Soit la méthode suivante :

```
public boolean onContextItemSelected(MenuItem item) {  
    ...  
    return super.onContextItemSelected(item);  
}
```

1. Que permet de faire cette méthode ?

Elle permet d'associer un comportement à chaque élément du menu.

2. On voudrait faire appel respectivement aux méthodes *ajouter()*, *supprimer()* et *modifier()* si on clique sur *Ajouter*, *Supprimer* ou *Modifier*, et fermer l'activité si on clique sur *Quitter*. Donner le code nécessaire pour faire cela.

```
switch (item.getItemId()){  
    case 1:  
        ajouter();  
        break;  
    case 2:  
        supprimer();  
        break;  
    case 3:  
        modifier();  
        break;  
    case 4:  
        finish();  
}
```

c. Quelle est la méthode qu'on doit surcharger si on veut définir le **contenu** du menu affiché quand on clique sur le bouton *Menu* du téléphone ? Donnez le code nécessaire pour faire cela.

onCreateOptionsMenu

```
public boolean onCreateOptionsMenu(Menu menu) {  
  
    menu.add(1, 1, 3, "Ajouter");  
    menu.add(1, 2, 2, "Supprimer");  
    menu.add(1, 3, 0, "Modifier");  
    menu.add(1, 4, 1, "Quitter");  
    return true;  
}
```


Corrigé TD6

Vous allez modifier le fichier *CalculActivity.java*

Q1 : Qu'est-ce qu'une activité en Android ? De quelle classe doit-elle hériter ?

Une activité est la composante principale pour une application Android. Elle représente l'implémentation métier dans une application Android, permettant de gérer l'ensemble des vues et ressources.

Elle hérite de la classe Activity.

Q2 : Quelle est la différence entre une activité et une application ? (0,5pt)

Une application peut contenir plusieurs activités.

Votre application va avoir le comportement suivant :

- L'utilisateur saisit deux entiers dans les champs d'édition *val1* et *val2*
- Il choisit l'opération à faire : *somme* ou *différence* en cliquant sur le bouton radio associé
- Il clique sur le bouton *calculer*. Le résultat est affiché dans le champs de texte *res*.

Q3 : Donner la ligne de code nécessaire pour associer l'interface définie avec l'activité *CalculActivity*.

setContentView(R.layout.choixOperation)

Vous avez créé une méthode dans le code pour définir le comportement de votre bouton.

Q4 : Donner la signature de cette méthode (son modificateur de visibilité, son type de retour, son nom et des paramètres).

public void calculer(View v){...}

public void onClick(View v) est acceptée aussi.

Vous avez créé deux attributs représentant les deux boutons radios :

```
private RadioButton b1;  
private RadioButton b2;
```

Vous voulez que *b1* soit associé au bouton radio *somme* et *b2* soit associé à *diff*.

Q5 : Dans quelle méthode proposez-vous de faire ces initialisations ? pourquoi ?

Dans la méthode onCreate

Q6 : Donner les lignes de code nécessaires pour faire cela.

```
b1 = (RadioButton) findViewById(R.id.somme) ;  
b2 = (RadioButton) findViewById(R.id.diff) ;
```

Passage d'une activité à une autre

Vous désirez créer une deuxième activité, appelée *ResultActivity*, pour y afficher le résultat.

Q7 : Donner les étapes nécessaires pour créer une autre activité.

- *Créer une nouvelle classe Java pour cette activité (elle doit hériter de Activity et contenir une méthode onCreate)*
- *Créer une nouvelle interface dans le répertoire layout*
- *Associer l'interface avec l'activité (grâce à setContentView)*
- *Ajouter l'activité dans le fichier Manifest.*

Le but est d'afficher une nouvelle interface en cliquant sur le bouton. Pour cela, nous avons besoin d'un *Intent*.

Q8 : Qu'est-ce qu'un Intent ?

Un Intent est un conteneur d'information. Il permet de passer des messages entre deux activités. L'activité appelante pourra ainsi transmettre des informations à l'application appelée, ainsi qu'au système Android.

Pour créer un nouvel Intent, nous avons besoin de la ligne de code suivante :

```
Intent i = new Intent(<contexte>, <cible>);
```

Q9: Quelle est la valeur de *contexte* et de *cible* dans notre cas ?

Context = CalculActivity.this

Cible = ResultActivity.class

Q10 : Donner la ligne de code nécessaire pour lancer la deuxième activité.

startActivity(i) ;

Dans notre application, nous avons calculé le résultat à afficher et l'avons stocké dans la variable *resultat*. Nous désirons l'envoyer à l'activité suivante.

Q11 : Donner la ligne de code nécessaire pour envoyer la variable *resultat* dans un message qu'on nommera *mess*.

i.putExtra("mess", resultat) ;