

## Examen Du Module Applications Mobiles

**Exercice 1:** Choisissez la bonne réponse (Il peut y avoir plusieurs bonnes réponses) :

1. Soit «MaVue» une classe qui dérive de «View» et «uneVue» une instance de «MaVue». On veut afficher cette instance («uneVue») quand l'application démarre:

- a. setCurrentView(uneVue) dans la méthode startApp() de la classe principale.
- b. setContentView(uneVue) dans la méthode startApp() de la classe principale.
- c. setContentView(uneVue) dans la méthode onStart() de la classe principale.
- d. setContentView(uneVue) dans la méthode onCreate() de la classe principale.

2. Soit «écran1» l'écran principal de l'application «TestAndroid». Si un autre écran, «écran2», doit être ouvert à partir de «écran1»...

- a. «écran2» doit faire partie de «TestAndroid».
- b. «écran2» peut faire partie de n'importe quelle autre application installée sur l'appareil.
- c. «écran2» sera toujours lancé de manière asynchrone.
- d. «écran2» peut-être lancé de manière synchrone.
- e. «écran2» peut retourner un résultat à «écran1» si la méthode «startActivity» est utilisée pour démarrer l'activité associée à «écran2».
- f. «écran2» peut retourner un résultat à «écran1» si la méthode «startActivityForResult» est utilisée pour démarrer l'activité associée à «écran2».

3. Quelle est la composante qui n'est pas activée par un intent?

- a. Activité
- b. Service
- c. ContentProvider
- d. BroadcastReceiver

4. Lesquelles de ces assertions décrivent l'émulateur AndroidSDK?

- a. Un émulateur n'exécute pas en réalité le système Android. Il s'agit plutôt d'un composant logiciel qui interprète les fonctionnalités et les actions de toute l'API d'Android.
- b. Un seul émulateur peut-être exécuté à la fois.
- c. L'émulateur n'a pas besoin d'être redémarré après chaque déploiement d'une application.

5. A quoi correspond le terme Dalvik :

- a. C'est le nom du JDK utilisé sur les systèmes android.
- b. C'est le nom de la machine virtuelle utilisée sur les systèmes android.
- c. C'est le nom du JRE utilisé sur les systèmes android.



85/100

**Exercice 2:**

1. Quelle est la signification du signe « + » dans la déclaration `android:id="@+id/un_id"`.  
 R1 : le « + » désigne un resource id (une ressource) dans un fichier `res/values/ids.xml`.  
 R2 : l'importance de fixer les permissions lors du développement de votre application.  
 R3 : Avec quel objet on peut démarrer une activité à partir d'une autre (Donner un exemple d'utilisation)  
 R3 : avec un Intent.  
`Intent intent = new Intent(this, SecondA.class);`  
`startActivity(intent);`
2. Préciser l'importance de fixer les permissions lors du développement de votre application?  
 R2 : l'importance de fixer les permissions lors du développement de votre application.
3. Avec quel objet on peut démarrer une activité à partir d'une autre (Donner un exemple d'utilisation)  
 R3 : avec un Intent.  
`Intent intent = new Intent(this, SecondA.class);`  
`startActivity(intent);`

4. Nous souhaitons qu'un Toast apparaisse quand on fait un clic sur le bouton 'btn\_Afficher', indiquant le contenu d'un "EditText" nommé 'etxt1'. Complétez avec les instructions appropriées?

```

btn_Afficher.setOnClickListener(click listener (new View.OnClickListener() {
    @Override
    public void onClick (View v) {
        Toast.makeText(MonActivity.this, etxt1.getText().toString(),
        Toast.LENGTH_SHORT).show();
    }
}));
  
```

5. soit la méthode suivante :

```

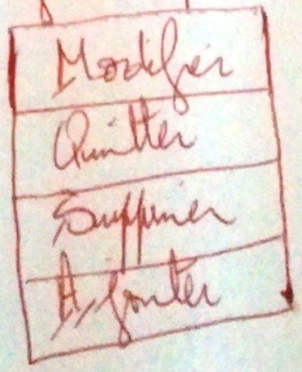
public void onCreateContextMenu(ContextMenu menu, View v,
    ContextMenuInfo menuInfo) {

    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");
  }
  
```

Que permet de faire cette méthode ? Dessiner son résultat.

R5 : Cette méthode permet d'ajouter quatre éléments au menu : Ajouter, Supprimer, Modifier et Quitter.





6. Dessiner l'interface graphique correspondante au fichier XML suivant :

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" >
    <TextView
        android:id="@+id/label"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/Courriel" />

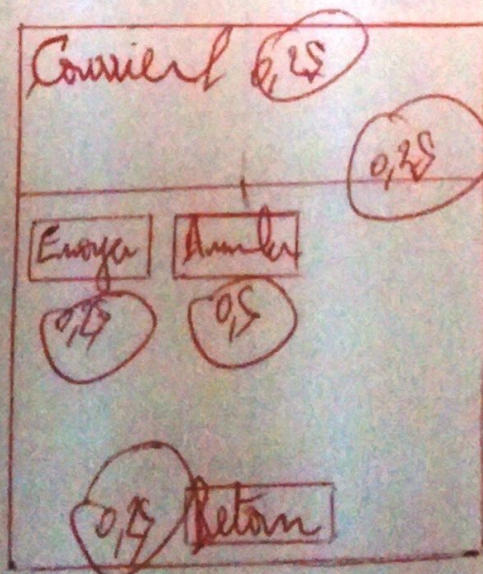
    <EditText
        android:id="@+id/inputEmail"
        android:inputType="textEmailAddress"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/label" />

    <Button
        android:id="@+id/btnLogin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@id/inputEmail"
        android:text="@string/Envoyer" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@id/btnLogin"
        android:layout_toRightOf="@id/btnLogin"
        android:text="@string/Annuler" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:text="@string/Retour" />
</RelativeLayout>
```

R6:





5,50 pts

**Exercice 3 :** Vous avez une activité enfant (activité destination) proposant à l'utilisateur un formulaire de réponse **Oui / Non**.

1-Comment récupérer la valeur saisie par l'utilisateur dans l'activité enfant depuis l'activité principale ?

R1 : En utilisant la méthode `startActivityForResult` dans l'activité principale.

08



2-Ecrire les parties de code des deux activités permettant de faire cela, en affichant dans l'interface de l'activité principale un toast : « Action validée » ou « Action annulée » selon le bouton sélectionné « Oui » ou «Non » respectivement dans l'interface enfant.

R2 : Activité Principale

Activité Destination

`Intent intent = new Intent(this, Enfant.class);  
startActivityForResult(intent, CODE_MEN_ACTIVITE);`

`@Override  
public void onActivityResult (int requestCode, int resultCode, Intent data) {  
 switch (requestCode) {  
 case R.id.button1:`

`@Override  
 protected void onActivityResult (int requestCode, int resultCode, Intent data) {  
 switch (requestCode) {`

`setResult (RESULT_OK);  
 finish();  
 break;`

`case CODE_MEN_ACTIVITE:`

`case R.id.button2:`

`switch (resultCode) {  
 case RESULT_OK:  
 Toast.makeText(this, "Action validée",  
 Toast.LENGTH_LONG).show();  
 return;`

`setResult (RESULT_CANCELED);  
 finish();  
 break;`

`case RESULT_CANCELED:  
 Toast.makeText(this, "Action annulée",  
 Toast.LENGTH_LONG).show();  
 return;`

`default:  
 // Faire quelque chose  
 return;`

`default:  
 // Faire quelque chose  
 return;`