

Devoir Surveillé [session principale]

Niveau d'Etude : L2SE

Matière : Programmation Mobile

Enseignants Responsables: Lilia Sfaxi

Année Universitaire : 2011/2012

Semestre : 2^{ème} Semestre

Date : 02/04/2012

Durée : 1h30

Documents : Non autorisés

Nombre de pages : 6

Le barème est donné à titre indicatif. La clarté de la copie rendue sera prise en considération.

Installation

Vous désirez réaliser une application avec Android 2.3.3 en utilisant l'environnement Eclipse.

Q1. Citer l'ensemble des éléments que vous devez installer. (1pt)

- *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*

En créant une nouvelle application Android avec Eclipse, intitulée Calcul, vous obtenez l'arborescence suivante :

```
_ Calcul
  _ (1)
    _ isi.calcul
      _ CalculActivity.java
  _ (2)
    _ isi.calcul
      _ (3)
        _ Android 2.3.3
        _ assets
        _ bin
        _ (4)
          _ drawable-(5)
            _ ic_launcher.png
          _ drawable-ldpi
          _ drawable-mdpi
        _ (6)
```

```
        |_ choixOperation.xml
    |_ (7)
        |_ strings.xml
    |_ (8)
    |_ proguard.cfg
    |_ project.properties
```

Q2 : Donner le nom associé à chacun des numéros. (2pts)

(1) : src

(2) : gen

(3) : R.java

(4) : res

(5) : hdpi

(6) : layout

(7) : values

(8) : AndroidManifest.xml

Q3 : Quel est le rôle de l'élément (3) ? à partir de quel élément est-il généré ? (2pt)

L'élément (3) est un fichier qui représente les ressources de l'application sous forme d'éléments Java, pour qu'ils soient utilisés dans le code Java des activités. Il est généré à partir du répertoire res.

Interface

Dans votre application, vous avez commencé par modifier l'interface.

Q4 : Quel fichier devez-vous modifier pour définir l'interface ? (0,5pt)

choixOperation.xml

Vous avez écrit le code suivant :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout (...) >
    <TextView
        android:id="@+id/titre"
        android:text="Application de calcul"/>

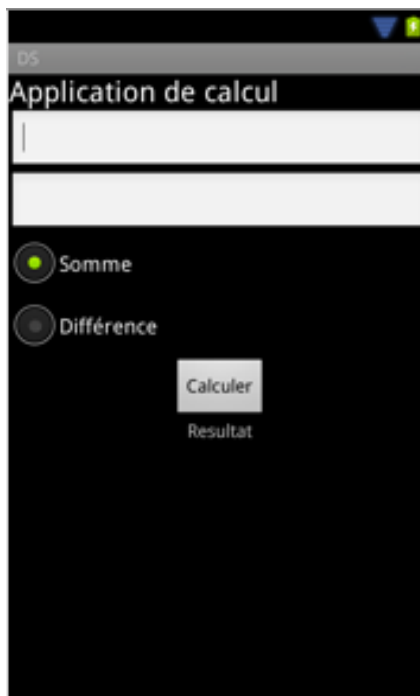
    <EditText
        android:id="@+id/val1"
```

```

        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>

```

Q5 : Dessiner l'interface que vous obtenez avec ce code. (l'emplacement des éléments et leurs étiquettes sont prises en considération). (1pt)



Q6 : A quoi sert l'élément *RadioGroup* ? (0,5pt)

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

Q7 : A quoi sert l'attribut *android:layout_gravity="center"*? (0,5pt)

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

Code

Vous allez modifier le fichier *CalculActivity.java*

Q8 : Qu'est-ce qu'une activité en Android ? De quelle classe doit-elle hériter ? (1pt)

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.

Q9 : 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*.

Q10 : Donner la ligne de code nécessaire pour associer l'interface définie avec l'activité *CalculActivity*. (1pt)

setContentView(R.layout.choixOperation)

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

Q11 : Donner la signature de cette méthode (son modificateur de visibilité, son type de retour, son nom et des paramètres). (2pt)

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*.

Q12 : Dans quelle méthode proposez-vous de faire ces initialisations ? pourquoi ? (1pt)

Dans la méthode onCreate

Q13 : Donner les lignes de code nécessaires pour faire cela. (1pt)

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.

Q14 : Donner les étapes nécessaires pour créer une autre activité. (2pts)

- *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*.

Q15 : Qu'est-ce qu'un Intent ? (1pt)

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>);
```

Q16 : Quelle est la valeur de *contexte* et de *cible* dans notre cas ? (1pt)

Context = CalculActivity.this

Cible = ResultActivity.class

Q17 : Donner la ligne de code nécessaire pour lancer la deuxième activité.

(1pt)

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.

Q18 : Donner la ligne de code nécessaire pour envoyer la variable *resultat* dans un message qu'on nommera *mess*. *(1pt)*

i.putExtra("mess", resultat) ;

BON TRAVAIL.