

Trimestre Hiver, 2019

Mohamed Lokbani

IFT1155 – Examen Intra –

Inscrivez tout de suite : votre nom et le code permanent.

Nom : _____ Prénom(s) : _____

Signature : _____ Login : _____

Date : Vendredi 1 mars 2019

Durée : 2 heures (de 16h30 à 18h30)

Local : Z-350, Pavillon Claire McNicoll

Directives :

- Toute documentation est permise.
- Calculatrice **non** permise.
- Répondre directement sur le questionnaire.
- Les réponses **doivent être brèves, précises, claires et nettement présentées.**

1. _____ /20

2. _____ /20

3. _____ /24

4. _____ /36

Total : _____ /100

Directives officielles

* Interdiction de toute communication verbale pendant l'examen.

* Interdiction de quitter la salle pendant la première heure.

* L'étudiant qui doit s'absenter après la première heure remettra sa carte d'étudiant au surveillant, l'absence ne devant pas dépasser 5 minutes. Un seul étudiant à la fois peut quitter la salle.

* Toute infraction relative à une fraude, un plagiat ou un copiage est signalée par le surveillant au directeur de département ou au professeur qui suspend l'évaluation.

F.A.S

Exercice 1 (20 points) trouvez la meilleure correspondance pour les descriptions ci-dessous en y associant un nombre de 1 à 11. Le nombre ne peut-être utilisé qu'une seule fois. Nous avons complété une description à titre d'exemple.

En développement Android, les _____ servent à organiser les _____ (un bouton, un texte, une image, etc.).

Toutes les vues doivent spécifier ces deux attributs :

_____ largeur de la vue

_____ hauteur de la vue

Ils peuvent valoir :

_____ : la vue est la plus petite possible

_____ : la vue est la plus grande possible

9 : une taille fixe (non recommandée)

Les dp sont une unité de taille indépendante de l'écran. 100dp font 100 _____ sur un écran de 100 _____.

Il est possible de modifier l'espacement des vues avec :

_____ : espace entre le texte et les bords, géré par chaque vue.

_____ : espace autour des bords, géré par les groupes.

dpi	1	
layout	2	
layout_height	3	
layout_width	4	
match_parent	5	
margin	6	
padding	7	
pixel	8	
valeur en dp	9	x
vue	10	
wrap content	11	

Exercice 2 (20 points)

2.1 Expliquer ce qui suit : « pourquoi Android est si populaire? ».

2.2 Expliquer ce qui suit : « une des faiblesses d'Android est son système ouvert (open source). »

2.3 Expliquer ce qui suit : « l'activité « A » lance l'activité « B ». L'activité « B » sera empilée sur l'activité « A », »

2.4 À travers un exemple simple comme le téléchargement d'une application Android quelconque du « Google Play store », expliquez le fonctionnement, et l'interaction entre elles, des 3 composantes suivantes : « Activity », « Service » et « Broadcast Receiver ».

Exercice 3 (24 points)

3.1 dessiner l'interface graphique à partir du fichier XML ci-dessous dont la syntaxe est correcte, en prenant soin d'inclure une courte description de votre démarche.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <Button
        android:id="@+id/bouton1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:layout_weight="35"
        android:text="Bouton 1" />
    <Button
        android:id="@+id/bouton2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_weight="30"
        android:gravity="bottom|right"
        android:text="Bouton 2" />
    <Button
        android:id="@+id/bouton3"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="top"
        android:layout_weight="35"
        android:text="Bouton 3" />
</LinearLayout>
```

--	--

3.2 dessiner l'interface graphique à partir du fichier XML ci-dessous dont la syntaxe est correcte, en prenant soin d'inclure une courte description de votre démarche.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:id="@+id/alpha"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:text="Alpha"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_toLeftOf="@id/alpha"
        android:layout_alignBottom="@id/alpha"
        android:text="2"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_toRightOf="@id/alpha"
        android:layout_alignBottom="@id/alpha"
        android:text="3"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@id/alpha"
        android:layout_alignLeft="@id/alpha"
        android:text="1"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/alpha"
        android:layout_alignLeft="@id/alpha"
        android:text="4"/>
</RelativeLayout>
```

--	--

Exercice 4 (36 points) nous allons examiner la composante « service » en profondeur.

4.1 À quoi servent ces méthodes associées à un service?

Méthode	Utilité
onStartCommand()	
onBind()	
onCreate()	
onDestroy()	

4.2 Est-ce que nous avons besoin de déclarer le service dans le fichier manifeste de l'application? (pourquoi, courte explication)

4.3 Expliquer cet extrait de la documentation de Google relative aux services :

« To ensure that your app is secure, always use an explicit intent when starting a Service and don't declare intent filters for your services. »

« Pour s'assurer que votre application est sécuritaire, utiliser toujours un intent explicite quand vous démarrez un Service et éviter de déclarer des filtres d'intention pour votre service ». (ma traduction)

4.4 L'application est composée des deux classes : « **MainActivity** » et « **WeatherService** ». Pour information, le fichier «activity_main.xml » est disponible en annexe.

```
01 public class MainActivity extends AppCompatActivity {
02     private boolean binded=false;
03     private WeatherService weatherService;
04     private TextView weatherText;
05     private EditText locationText;
06
07     ServiceConnection weatherServiceConnection = new ServiceConnection() {
08         @Override
09         public void onServiceConnected(ComponentName name, IBinder service) {
10             WeatherService.LocalWeatherBinder binder =
11                 (WeatherService.LocalWeatherBinder) service;
12             weatherService = binder.getService();
13             binded = true;
14         }
15         @Override
16         public void onServiceDisconnected(ComponentName name) {
17             binded = false;
18         }
19     };
20     @Override
21     protected void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
23         setContentView(R.layout.activity_main);
24         weatherText = this.findViewById(R.id.text_weather);
25         locationText = this.findViewById(R.id.text_input_location);
26     }
27     @Override
28     protected void onStart() {
29         // à compléter
30     }
31     @Override
32     protected void onStop() {
33         // à compléter
34     }
35     public void showWeather(View view) {
36         String location = locationText.getText().toString();
37         String weather= this.weatherService.getWeatherToday(location);
38         weatherText.setText(weather);
39     }
40 }
```

4.4.1 Compléter les méthodes « **onStart** » (ligne « 28 ») et « **onStop** » (« ligne 32 ») de la classe « **MainActivity** »

```
01 public class WeatherService extends Service {
02     public WeatherService() {
03     }
04     private final IBinder binder = new LocalWeatherBinder();
05
06     class LocalWeatherBinder extends Binder {
07         // à compléter
08     }
09     @Override
10     public IBinder onBind(Intent intent) {
11         return this.binder;
12     }
13     @Override
14     public void onRebind(Intent intent) {
15         super.onRebind(intent);
16     }
17     @Override
18     public boolean onUnbind(Intent intent) {
19         return true;
20     }
21     @Override
22     public void onDestroy() {
23         super.onDestroy();
24     }
25     public String getWeatherToday(String location) {
26         String weather;
27
28         String[] weathers = new String[]{"Pluvieux", "Chaud",
29                                         "Printanier", "Doux" ,"Froid"};
30
31         // Une valeur aléatoire entre 0 et 4
32         int i= new Random().nextInt(5);
33         weather = weathers[i];
34         return weather;
35     }
36 }
```

4.4.2 compléter la classe (interne) « **LocalWeatherBinder** » de la classe « **WeatherService** » (ligne « 7 »).

4.4.3 À travers des exemples concrets de l'utilisation de l'application, expliquer dans quelles situations les trois méthodes « onBind », « onUnbind » et « onDestroy », de la classe « WeatherService », sont appelées.

L'application a été installée sur l'émulateur. Ces 3 méthodes vont être probablement appelées, cela va dépendre comment vous allez manipuler l'application sur l'émulateur. On vous demande de nous donner un exemple pour chaque méthode. Une situation suffit pour expliquer chaque méthode. Il faut juste la décrire brièvement.

Annexe

Fichier « activity_main.xml » de l'exercice « 4 ».

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/text_weather"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:text="Météo"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <TextView
        android:id="@+id/text_location"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:text="Lieu"
        app:layout_constraintBottom_toTopOf="@+id/text_weather"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <Button
        android:id="@+id/button_weather"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:onClick="showWeather"
        android:text="Quel est le temps aujourd'hui?"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/text_weather" />
    <EditText
        android:id="@+id/text_input_location"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:text="Montreal"
        android:textAlignment="center"
        app:layout_constraintBottom_toTopOf="@+id/text_weather"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/text_location" />
</android.support.constraint.ConstraintLayout>
```