Trimestre Hiver, 2023

Mohamed Lokbani

IFT1155 – Examen Intra –

Inscrivez tout de suite : votre nom et prénom et signer le document.

Nom : _____| Prénom(s) : _____|

Signature : _____| **Ne pas répondre au verso des feuilles d'examen**

Date : mercredi 22 février 2023

Durée : 2 heures (de 16h30 à 18h30)

Local : Z-255, Pavillon Claire McNicoll

Directives :

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

1._____/20

2._____/30

3._____/24

4._____/26

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) répondez par « vrai » ou « faux » **en y incluant une courte explication**.

1.0 [~~VRAI~~ | FAUX] Le sigle du cours est : **IFT1169**.

Le sigle du cours est IFT1155.

1.1 [VRAI | FAUX] Le code source des systèmes libres n'est accessible que par les concepteurs de ces systèmes.

1.2 [VRAI | FAUX] Nous avons besoin d'installer sur son ordinateur le JDK de Java avant de pouvoir programmer des applications Android (avec Java) dans Android Studio.

1.3 [VRAI | FAUX] Une fois l'émulateur est créé, il n'est plus possible de changer ses paramètres.

1.4 [VRAI | FAUX] « Android studio » intègre un programme de gestion de versions.

1.5 [VRAI | FAUX] La dernière version d'Android Studio est « Arctic Fox » (Renard polaire).

1.6 [VRAI | FAUX] Si plusieurs appareils (ou émulateurs) sont en cours d'exécution, vous devez spécifier l'appareil cible lorsque vous utilisez la commande « adb ».

1.7 [VRAI | FAUX] Il n'est pas possible de mettre à jour les paquetages installés dans Android Studio (SDK, outils, etc.) en ligne de commandes (dans un un terminal).

1.8 [VRAI | FAUX] Android exécute les applications dans le même environnement d'exécution, leur permettant de communiquer aisément entre elles.

1.9 [VRAI | FAUX] Un niveau d'API est unique pour chaque version d'Android.

1.10 [VRAI | FAUX] L'activité « A » lance l'activité « B ». « A » est empilée sur « B ».

Exercice 2 (30 points)

2.1 Expliquer ce qui suit: « Si on veut cibler le maximum d'appareils, nous devons développer l'application pour une taille d'écran normal. »

2.2 Expliquer les 8 lignes ci-dessous extraites du fichier « AndroidManifest.xml » d'une application lambda.

```
<activity
    android:name=".MainActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

2.3 Expliquer l'intérêt d'accompagner votre projet d'un rapport conséquent.

2.4 Les chiffres indiqués sur cette capture d'écran sont trompeurs. Si on veut cibler +80% des appareils, il nous faudra développer l'application avec l'API 28. Sauf que Google est d'un autre avis. Expliquer cela.

Plateforme		API	Cumul
8.0	Oreo	26	90.7%
8.1	Oreo	27	88.1%
9.0	Pie	28	81.2%
10.	Q	29	68.0%
11.	R	30	48.5%
12.	S	31	24.1%
13.	T	33	5.2%

2.5 Si on utilise un « intent » implicite, expliquer pourquoi il est souhaitable lorsque le composant n'est pas standard de s'assurer que ce composant est bien disponible sur l'appareil.

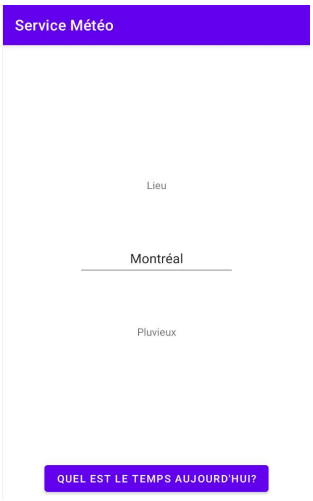
2.6 À travers un exemple simple comme le téléchargement d'une application Android quelconque du « Google Play Store », expliquer le fonctionnement et l'interaction entre elles, des composantes suivantes: « Activity », « Service » et « Brodcast Reciever ».

Exercice 3 (24 points) dessiner l'interface graphique à partir du fichier XML ci-dessous dont la syntaxe est correcte :

01	<androidx.constraintlayout.widget.ConstraintLayout	Couleur et « String »
02	xmlns:android="http://schemas.android.com/apk/res/android"	
03	xmlns:app="http://schemas.android.com/apk/res-auto"	
04	xmlns:tools="http://schemas.android.com/tools"	
05	android:layout_width="match_parent"	
06	android:layout_height="match_parent"	
07	tools:context=".MainActivity">	
08	<TextView	
09	android:id="@+id/view1"	
10	android:layout_width="match_parent"	
11	android:layout_height="150dp"	
12	android:background="#0000ff"	"#0000ff" : bleu
13	android:text="@string/Vue1"	Vue1 : Vue1
14	android:textColor="@color/white"	
15	android:textSize="34sp"	
16	app:layout_constraintLeft_toLeftOf="parent"	
17	app:layout_constraintTop_toTopOf="parent" />	
18	<TextView	
19	android:id="@+id/view2"	
20	android:layout_width="200dp"	
21	android:layout_height="0dp"	
22	android:layout_marginBottom="431dp"	
23	android:background="#ff0000"	"#ff0000" : rouge
24	android:text="@string/Vue2"	Vue2 : Vue2
25	android:textColor="@color/black"	
26	android:textSize="34sp"	
27	app:layout_constraintBottom_toBottomOf="parent"	
28	app:layout_constraintLeft_toLeftOf="parent"	
29	app:layout_constraintTop_toBottomOf="@+id/view1" />	
30	<TextView	
31	android:id="@+id/view3"	
32	android:layout_width="0dp"	
33	android:layout_height="150dp"	
34	android:background="#ffffff"	"#ffffff" : blanc
35	android:text="@string/Vue3"	Vue3 : Vue3
36	android:textColor="@color/black"	
37	android:textSize="34sp"	
38	app:layout_constraintLeft_toRightOf="@id/view2"	
39	app:layout_constraintRight_toRightOf="parent"	
40	app:layout_constraintTop_toBottomOf="@id/view1" />	
41	<TextView	
42	android:id="@+id/view4"	
43	android:layout_width="150dp"	
44	android:layout_height="0dp"	
45	android:background="#00ff00"	"#00ff00" : vert
46	android:text="@string/Vue4"	Vue4 : Vue4
47	android:textColor="@color/black"	
48	android:textSize="34sp"	
49	app:layout_constraintBottom_toBottomOf="parent"	
50	app:layout_constraintLeft_toLeftOf="parent"	
51	app:layout_constraintTop_toBottomOf="@id/view2" />	
52	<TextView	
53	android:id="@+id/view5"	
54	android:layout_width="0dp"	
55	android:layout_height="0dp"	
56	android:background="#ffff00"	"#ffff00" : jaune
57	android:text="@string/Vue5"	Vue5 : Vue5
58	android:textColor="@color/black"	
59	android:textSize="34sp"	
60	app:layout_constraintBottom_toBottomOf="parent"	
61	app:layout_constraintLeft_toRightOf="@id/view4"	
62	app:layout_constraintRight_toRightOf="parent"	
62	app:layout_constraintTop_toBottomOf="@id/view2" />	
64		
65	</androidx.constraintlayout.widget.ConstraintLayout>	

Dessin avec vos explications

Exercice 4 (26 points) soit une application Android composé des deux classes « MainActivity » et « ServiceMeteo ». Cette application affiche une courte description météo de la ville de Montréal. En cliquant sur le bouton « Quel est le temps aujourd'hui », un service va se charger de transmettre la météo du jour.



4.1 Est-il ce que nous avons besoin de déclarer le service dans le fichier manifeste de l'application? (**développer votre réponse**).

4.2 Soit un extrait de la classe « MainActivity » :

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

Compléter les méthodes « **onStart** » (ligne « **27** ») et « **onStop** » (« ligne **31** ») de la classe « **MainActivity** ».

4.3 Soit un extrait de la classe « MeteoService » :

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

Compléter la classe interne « **LocalMeteoBinder** » (ligne « **7** »)

4.4 À travers des exemples concrets de l'utilisation de l'application « Service Météo », expliquer dans quelles situations les quatre méthodes « onBind », « onUnbind », « onRebind » et « onDestroy », de la classe « ServiceMeteo », sont appelées. L'application a été installée sur l'émulateur. L'appel à ces quatre méthodes va dépendre de l'utilisation que vous faites de 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.