

Trimestre Hiver, 2022

Mohamed Lokbani

IFT1155 – Examen Intra –

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

Nom : _____ Prénom(s) : _____

Signature : _____

Date : mercredi 23 février 2022

Durée : 2 heures (de 16h30 à 18h30)

Local : Z-337, 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. _____ /24

2. _____ /24

3. _____ /22

4. _____ /30

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 (24 points) compléter par un mot ce qui suit :

Exemple :

1.0 Le sigle du cours est : **IFT1155**.

À vous :

1.1 L'interface permettant de visualiser tous les messages imprimés par les différentes applications est :

1.2 L'outil permettant de spécifier les règles à appliquer sur vos projets Android est :

1.3 L'outil permettant de communiquer avec les appareils Android grâce à des commandes est :

1.4 L'outil de gestion de dépendance et d'automatisation est :

1.5 L'outil de développement d'applications Android est :

1.6 La structure et les métadonnées de votre application, ses composants et ses exigences sont définis dans :

1.7 L'outil permettant de tester vos applications est :

1.8 Un des langages de programmation d'applications Android est :

1.9 On démarre l'activité principale grâce à :

1.10 Les fichiers de mise en page sont stockés dans le dossier :

1.11 Pour disposer des éléments graphiques les uns par rapport aux autres dans une mise en page, on utilise la disposition :

1.12 L'interface utilisateur dans les applications Android est construite à l'aide de :

Exercice 2 (24 points) on vous demande d'expliquer les différents blocs du fichier manifeste ci-dessous :

01	<?xml version="1.0" encoding="utf-8"?>
02	<manifest xmlns:android="http://schemas.android.com/apk/res/android"
03	package="ca.umontreal.iro.ift1155.exo1"
04	android:versionCode="1"
05	android:versionName="1.0" >
06	
07	<uses-sdk
08	android:minSdkVersion="30"
09	android:targetSdkVersion="32" />
10	
11	<application
12	android:allowBackup="false"
13	android:icon="@drawable/icon"
14	android:label="@string/app_name" >
15	
16	<activity
17	android:name="ca.umontreal.iro.ift1155.exo1.EntryPoint"
18	android:allowBackup="false"
19	android:exported="true" >
20	
21	<intent-filter>
22	<action android:name="android.intent.action.MAIN" />
23	<category android:name="android.intent.category.LAUNCHER" />
24	</intent-filter>
25	
26	</activity>
27	
28	<activity
29	android:name="ca.umontreal.iro.ift1155.exo1.GoogleSearch"
30	android:exported="false" >
31	
32	<intent-filter>
33	<action android:name="android.intent.action.DEFAULT" />
34	</intent-filter>
35	
36	</activity>
37	
38	</application>
39	
40	</manifest>

2.1

01	<?xml version="1.0" encoding="utf-8"?>
----	--

2.2

02	<manifest xmlns:android="http://schemas.android.com/apk/res/android"
03	package="ca.umontreal.iro.ift1155.exo1"
04	android:versionCode="1"
05	android:versionName="1.0" >

2.3

07	<uses-sdk
08	android:minSdkVersion="30"
09	android:targetSdkVersion="32" />

2.4

11	<application
12	android:allowBackup="false"
13	android:icon="@drawable/icon"
14	android:label="@string/app_name" >

2.5

16	<activity
17	android:name="ca.umontreal.iro.ift1155.exo1.EntryPoint"
18	android:allowBackup="false"
19	android:exported="true" >

2.6

21	<intent-filter>
22	<action android:name="android.intent.action.MAIN" />
23	<category android:name="android.intent.category.LAUNCHER" />
24	</intent-filter>

2.7

28	<pre><activity android:name="ca.umontreal.iro.ift1155.exo1.GoogleSearch" android:exported="false" ></pre>
29	
30	

2.8

32	<pre><intent-filter> <action android:name="android.intent.action.DEFAULT" /> </intent-filter></pre>
33	
34	

Exercice 3 (22 points) dessiner l'interface graphique à partir du fichier XML ci-dessous dont la syntaxe est correcte :

01	<?xml version="1.0" encoding="utf-8"?>	couleur
02	<RelativeLayout	
03	xmlns:android="http://schemas.android.com/apk/res/android"	
04	android:layout_height="fill_parent"	
05	android:layout_width="fill_parent">	
06		
07	<TextView	
08	android:text="@string/rouge"	
09	android:id="@+id/TextView01"	
10	android:layout_height="wrap_content"	
11	android:background="#f00"	"#f00" : rouge
12	android:gravity="center"	
13	android:textColor="#000"	"#000" : noir
14	android:layout_width="wrap_content"	
15	android:padding="25dp">	
16	</TextView>	
17		
18	<TextView	
19	android:text="@string/orange"	
20	android:layout_height="wrap_content"	
21	android:background="#ffa500"	"#ffa500" : orange
22	android:gravity="center"	
23	android:textColor="#000"	"#000" : noir
24	android:id="@+id/TextView02"	
25	android:layout_width="wrap_content"	
26	android:layout_centerHorizontal="true"	
27	android:padding="25dp">	
28	</TextView>	
29		
30	<TextView	
31	android:text="@string/jaune"	
32	android:layout_height="wrap_content"	
33	android:background="#ffff00"	"#ffff00" : jaune
34	android:gravity="center"	
35	android:textColor="#000"	"#000" : noir
36	android:id="@+id/TextView03"	
37	android:layout_width="wrap_content"	
38	android:layout_alignParentEnd="true"	
39	android:padding="25dp">	
40	</TextView>	
41		
42	<TextView	
43	android:text="@string/vert"	
44	android:layout_height="wrap_content"	
45	android:background="#0f0"	"#0f0" : vert
46	android:gravity="center"	
47	android:textColor="#000"	"#000" : noir
48	android:id="@+id/TextView04"	
49	android:layout_width="wrap_content"	
50	android:layout_toStartOf="@+id/TextView05"	
51	android:padding="25dp"	
52	android:layout_centerVertical="true">	
53	</TextView>	
54	<TextView	
55	android:text="@string/bleu"	
56	android:layout_height="wrap_content"	
57	android:background="#00f"	"#00f" : bleu
58	android:gravity="center"	
59	android:textColor="#fff"	"#fff" : blanc
60	android:id="@+id/TextView05"	
61	android:layout_width="wrap_content"	
62	android:layout_centerInParent="true"	
62	android:layout_margin="10dp"	
64	android:padding="25dp">	
65	</TextView>	

66	<TextView	
67	android:text="@string/indigo"	
68	android:layout_height="wrap_content"	
69	android:gravity="center"	"#fff" : blanc
70	android:textColor="#fff"	
71	android:id="@+id/TextView06"	
72	android:layout_width="wrap_content"	
73	android:layout_toEndOf="@+id/TextView05"	
74	android:background="#4b0082"	"#4b0082" : indigo
75	android:padding="25dp"	
76	android:layout_centerVertical="true">	
77	</TextView>	
78		
79	<TextView	
80	android:text="@string/violet"	
81	android:layout_height="wrap_content"	
82	android:background="#ee82ee"	#ee82ee" : violet
83	android:gravity="center"	
84	android:textColor="#000"	"#000" : noir
85	android:id="@+id/TextView07"	
86	android:layout_alignParentBottom="true"	
87	android:layout_width="fill_parent"	
88	android:padding="25dp">	
89	</TextView>	
90		
91	</RelativeLayout>	

Dessin :

Explications :

Exercice 4 (30 points) l'application contient un seul bouton. Dès que l'utilisateur clique sur ce bouton, deux services sont lancés l'un après l'autre par l'activité principale dans cet ordre: ServiceS1 puis ServiceS2.

ServiceS1 envoie le message « Ping » à ServiceS2.
ServiceS2 envoie le message « Pong » à ServiceS1.

La classe ServiceS1 est presque une copie de la classe ServiceS2 moyennant quelques différences pour permettre un envoi efficace des messages entre les deux services.

Les deux services doivent enregistrer un récepteur d'événement, respectivement, S1Receiver et S2Receiver. Les deux services doivent implémenter dans ce cas la méthode onReceive correspondante.

Extrait du log après l'exécution de la méthode onReceive du service ServiceS1:

```
D/Log2TAG: onReceive S1!  
D/Log2TAG: pong
```

Extrait du log après l'exécution de la méthode onReceive du service ServiceS2:

```
D/Log2TAG: onReceive S2!  
D/Log2TAG: ping
```

Extrait du log après l'exécution de l'application

```
D/Log2TAG: Démarrage de l'application deuxservices ...  
D/Log2TAG: onStartCommand S1  
D/Log2TAG: Arguments: id: 1 , intent: Intent { cmp=ca.iro.umontreal.ift1155.deuxservices/.ServiceS1 }  
D/Log2TAG: onStartCommand S2  
D/Log2TAG: Arguments: id: 1 , intent: Intent { cmp=ca.iro.umontreal.ift1155.deuxservices/.ServiceS2 }  
D/Log2TAG: onReceive S1!  
D/Log2TAG: pong  
D/Log2TAG: onReceive S2!  
D/Log2TAG: ping  
D/Log2TAG: onDestroy S1  
D/Log2TAG: onDestroy S2  
D/Log2TAG: L'application deuxservices est détruite!
```

Dans les différentes méthodes que vous allez développer ci-dessous, il vous est demandé d'inclure le code nécessaire permettant d'obtenir en sortie l'extrait du log précédemment mentionné.

4.1 Écrire les instructions permettant d'enregistrer ServiceS1 et ServiceS2 dans le fichier manifeste de l'application.

4.2 Compléter la méthode onClick définie dans la méthode onCreate de la classe MainActivity (activité principale) et associée à l'écouteur du bouton et permettant de démarrer ServiceS1 et ServiceS2.

4.3 Écrire la méthode onStop de MainActivity.

4.4 Écrire la méthode onDestroy de MainActivity.

4.5 Écrire la méthode onStartCommand du ServiceS1 définie dans la classe ServiceS1.

4.6 Écrire la méthode onStartCommand du ServiceS2 définie dans la classe ServiceS2.

4.7 Écrire le récepteur d'événement S1Receiver défini dans la classe ServiceS1.

4.8 Écrire le récepteur d'événement S1Receiver défini dans la classe ServiceS2.

4.9 Écrire la méthode onDestroy du service ServiceS1 définie dans la classe ServiceS1.

4.10 Écrire la méthode onDestroy du service ServiceS2 définie dans la classe ServiceS2.

4.11 Comment êtes-vous assuré du bon enchaînement des échanges de messages entre les deux services?

D/Log2TAG: onReceive S1!

D/Log2TAG: pong

D/Log2TAG: onReceive S2!

D/Log2TAG: ping