Université El Arbi Ben M'Hidi Oum El Bouaghi
Faculté de sciences exactes et Sciences de la nature et de la vie
Département de Mathématique et informatique

Niveau : 3éme années licence SI — Dimanche 12 mai 2024
Module : Les applications mobiles — Durée 1h 30 m

Examen Final

Exercices 01 (4 points) : recopier les bonnes réponses

1. Dans Android, une application partage ses données avec d'autres applications via :
   A. Un fournisseur de contenue. B. Une Activité. **1pts**
   C. Un Service. D. Un Intent-Filter.
2. Toutes les ressources sont déclarées dans le fichier :
   A. R.java B. strings.xml **1pts**
   C. AndroidManifest.xml D. values.xml
3. Lequel des énoncés suivants est false à propos de runtime ART ?
   A. Il a remplacé le moteur d'exécution (Dalvik) dans les versions ultérieures d'Android.
   B. Il utilise une compilation Ahead of time (AOT) pour exécuter le code. **1pts**
   C. Il compile le code source Java directement en code machine.
4. Sous Android Studio, les intents permettent :
   A. De lancer une nouvelle activité. B. La communication entre applications.
   C. De gérer les autorisations et les restrictions d'accès aux ressources de l'appareil
   D. De lire des données à partir d'une base de données **1pts**

Exercice 02 : (5.5 points)

- Dans l'architecture Android, quel est le rôle de la couche des bibliothèques C/C++ ? Citez trois bibliothèques de cette couche et décrivez leur fonction.
  Cette couche fournit des fonctionnalités natives optimisées pour des tâches critiques (graphismes, médias, accès matériel) et sert d'interface entre le noyau Linux et les couches Java/Kotlin. **1pts**
  - OpenSSL : OpenSSL est une bibliothèque de cryptographie qui fournit divers protocoles, algorithmes et outils pour les communications sécurisées, tels que SSL et TLS.
  - zlib : zlib est une bibliothèque de compression qui fournit divers algorithmes et outils pour la compression et la décompression de données, tels que gzip et zip. **0.5 *3 pts**
  - SQLite : SQLite est une bibliothèque de base de données qui fournit divers outils pour créer et gérer des bases de données SQL légères, telles que celles utilisées dans les applications mobiles.
- Quelle est la différence principale entre un services premier plan et les services d'arrière-plan ?

  Service Premier Plan exécute une tâche visible pour l'utilisateur (afficher une notification), Service d'Arrière-plan exécute des tâches non visibles (ex : synchronisation de données). **1pts**
- Quelle est le rôle de ficher AndroidManifest.xml ?
  - Déclarer les composants : Activités, Services, BroadcastReceivers, ContentProviders.
  - Définir les permissions : Accès à la caméra, Internet, etc. **1pts**
  - Configurer l'application : Version, SDK minimum, orientation d'écrans, Intent-Filters (ex: définir une activité comme écran de lancement).
- Design la structure de la classe R.

  La classe R est générée automatiquement et contient des IDs uniques pour toutes les ressources de l'application.

```
public final class R {
    public static final class layout {
        public static final int activity_main = 0x7f0a0010;
    }
    public static final class drawable {
        public static final int icon = 0x7f0b0001;
    }
    public static final class string {
        public static final int app_name = 0x7f0c0002;
    }
    // ... Autres types (id, mipmap, color, etc.)
```

1pts

Exercice 03 : (10.5 points)

Vous êtes chargé de développer une application nommée FootballWorld, conçue pour centraliser les informations sur les clubs de football à travers le monde. Lors du lancement de l'application, une liste complète des clubs disponibles s'affiche. Lorsqu'un utilisateur sélectionne un club spécifique, une nouvelle interface s'ouvre, présentant la liste des joueurs appartenant à ce club. Chaque élément contient : une image du joueur (la même image pour tous les joueurs), son prénom, son nom, sa nationalité et sa position sur le terrain.

1. Quelle est le nom de l'élément utiliser pour afficher les listes de clubs et de joueurs ? **1 pts**
   a. Recycle View
2. Quelle méthode du cycle de vie utiliseriez-vous pour charger la liste des joueurs ?
   a. OnStart ou OnCreate **0.5 pts**
3. Quelle est la configuration nécessaire au fonctionnement de ce composant ?

   Pour fonctionner, un RecyclerView nécessite trois éléments clés :

   a. Modèle de données : Une classe représentant les attributs des éléments de la liste. **0.25* 3 pts**
   b. Adaptateur : Une classe qui Gère la liste et la liaison entre les données et les vues.
   c. Layout d'un élément : Un fichier XML définissant le design d'un élément de la liste

4. Modifiez le code XML fourni (Figure 1) pour (indiquer la ligne de la modification) :
   a. Attribuer 50% de la hauteur (height) de l'élément à l'image du joueur.
   b. Répartir les 50% restants entre deux LinearLayout (25% chacun).
   c. Diviser équitablement la largeur (width) des TextView dans les LinearLayout.

   Réponse figure 1. **Chaque modification 0.25 *14**

5. Dans le code Java de l'adaptateur (Figure 2) :
   a. À quoi sert l'annotation @Override ? Indique que la méthode redéfinit une méthode de la classe parente. **0.5 pts**
   b. Quel est le rôle de la classe « Player » (ligne 12). Modèle de données contenant les attributs d'un joueur. **0.5 pts**
   c. Expliquez le rôle des trois fonctions soulignées.
      i. onCreateViewHolder() : crée et initialise le ViewHolder pour un élément de la liste.
      ii. onBindViewHolder() : Lie les données d'un joueur aux vues. **0.5 * 3 pts**

iii. getItemCount() : Retourne le nombre total de joueurs dans la liste.

d. Compléter le code des fonctions onBindViewHolder, getItemCount, et du constructeur PlayerHolder. Réponse figure 2

Chaque modification 0.25 * 9 pts

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android=
        "http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  android:layout_width="match_parent"
  android:layout_height="wrap_content">
  <androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    app:cardCornerRadius="5dp"
    app:cardElevation="10dp">
    <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:orientation="vertical"
      android:padding="10dp">
      <ImageView
        android:id="@+id/imgplayer"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="5"
        app:srcCompat="@android:drawable/player" />
      <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="2.5"
        android:orientation="horizontal">
        <TextView
          android:id="@+id/textView"
          android:layout_width="0dp"
          android:layout_height="match_parent"
          android:layout_weight="1"
          android:minHeight="100dp"
          android:text="TextView"
          android:textAlignment="center"
          android:textSize="24sp" />
        <TextView
          android:id="@+id/textView2"
          android:layout_width="0dp"
          android:layout_height="match_parent"
          android:layout_weight="1"
          android:text="TextView"
          android:textAlignment="center"
          android:textSize="24sp" />
      </LinearLayout>
      <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="2.5"
        android:orientation="horizontal">
        <TextView
          android:id="@+id/textView3"
          android:layout_width="0dp"
          android:layout_height="match_parent"
          android:layout_weight="1"
          android:text="TextView"
          android:textAlignment="center"
          android:textSize="24sp" />
        <TextView
          android:id="@+id/textView4"
          android:layout_width="0dp"
          android:layout_height="match_parent"
          android:layout_weight="1"
          android:text="TextView"
          android:textAlignment="center"
          android:textSize="24sp" />
      </LinearLayout>
    </LinearLayout>
  </androidx.cardview.widget.CardView>
</LinearLayout>
```

Figure 1: XML Code

```
10  public class PlayerAdapter extends RecyclerView.
11          Adapter<PlayerAdapter.PlayerHolder> {
12    List<Player> players;
13    PlayerAdapter (List<Player> players) {
14      this.players = players;
15    }
16    @Override
17    public PlayerAdapter.PlayerHolder onCreateViewHolder
                (ViewGroup parent, int viewType) {
18      View view = LayoutInflater.from(parent.getContext()).
19            inflate(R.layout.item,parent,false);
20      return new PlayerHolder(view);
21    }
22    @Override
23    public void onBindViewHolder(PlayerAdapter.PlayerHolder
                holder, int position) {
24      holder.name.setText(players.get(position).getName());
25      holder.lastName.setText(players.get(position).getLastName());
26      holder.nat.setText(players.get(position).getNationality());
27      holder.pos.setText(players.get(position).getPosition());
28    }
29    @Override
30    public int getItemCount() {
31      return players.size();
32    }
33    public static class PlayerHolder extends RecyclerView.
                                ViewHolder {
34      TextView name,lastName, nat, pos;
35      public PlayerHolder(View itemView) {
36        super(itemView);
37        name = itemView.findViewById(R.id.name);
38        lastName = itemView.findViewById(R.id.lastName);
39        nat = itemView.findViewById(R.id.nationality);
40        pos = itemView.findViewById(R.id.position);
41      }
42    }
43  }
```

Figure 2: Adapter Code (JAVA)