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 11 mai 2025
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.
   C. Un Service.
   D. Un Intent-Filter.
2. Toutes les ressources sont déclarées dans le fichier :
   A. R.java
   B. strings.xml
   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.
   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

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.
- Quelle est la différence principale entre un services premier plan et les services d'arrière-plan ?
- Quelle est le rôle de ficher AndroidManifest.xml ?
- Design la structure de la classe R.

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 ?
2. Quelle méthode du cycle de vie utiliseriez-vous pour charger la liste des joueurs ?
3. Quelle est la configuration nécessaire au fonctionnement de ce composant ?
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.
5. Dans le code Java de l'adaptateur (Figure 2) :
   a. À quoi sert l'annotation @Override ?
   b. Quel est le rôle de la classe « Player » (ligne 12).

c. Expliquez le rôle des trois fonctions soulignées.
d. Compléter le code des fonctions onBindViewHolder, getItemCount, et du constructeur PlayerHolder.

```
<?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="match_parent"
    app:srcCompat="@android:drawable/player" />
   <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">
    <TextView
     android:id="@+id/name"
     android:layout_width="match_parent"
     android:layout_height="match_parent"
     android:textSize="24sp" />
    <TextView
     android:id="@+id/lastName"
     android:layout_width="match_parent"
     android:layout_height="match_parent"
     android:textSize="24sp" />
   </LinearLayout>
   <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">
    <TextView
     android:id="@+id/nationality"
     android:layout_width="match_parent"
     android:layout_height="match_parent"
     android:textSize="24sp" />
    <TextView
     android:id="@+id/position"
     android:layout_width="match_parent"
     android:layout_height="match_parent"
     android:textSize="24sp" />
   </LinearLayout>
  </LinearLayout>
 </androidx.cardview.widget.CardView>
</LinearLayout>
```

Figure 1: XML Code

```
public class PlayerAdapter extends RecyclerView.
        Adapter<PlayerAdapter.PlayerHolder> {
  List<Player> players;
  PlayerAdapter (List<Player> players) {
    this.players = players;
  }
  @Override
  public PlayerAdapter.PlayerHolder onCreateViewHolder
            (ViewGroup parent, int viewType) {
    View view = LayoutInflater.from(parent.getContext()).
        inflate(R.layout.item,parent,false);
    return new PlayerHolder(view);
  }
  @Override
  public void onBindViewHolder(PlayerAdapter.PlayerHolder
      holder, int position) {

  }
  @Override
  public int getItemCount() {
    return 0;
  }
  public static class PlayerHolder extends RecyclerView.
            ViewHolder {
    TextView name,lastName, nat, pos;
    public PlayerHolder(View itemView) {
      super(itemView);
    }
  }
}
```

Figure 2: Adapter Code (JAVA)