

## Module : Les applications mobiles

### Examen Final - corrigé type -

#### Exercices 01 (4 points) : recopier les bonnes réponses

- Quels sont les composants fondamentaux d'une application Android ?
  - Activités, layouts, widgets.
  - Les intentions, les filtres, les ressources.
  - Services, fournisseurs, récepteurs.
  - Tous ces éléments.
- Le fichier AndroidManifest.xml est :
  - Un fichier qui contient les permissions nécessaires à l'exécution de l'application.
  - Un fichier de configuration d'une application Android.
  - Une application mobile
  - Le point d'entrée de l'application, qui indique quel code à exécuter au démarrage de l'app.
- Lequel des énoncés suivants est vrai à propos de runtime Dalvik ?
  - Il utilise une compilation Just-In-Time (JIT) pour exécuter le code.
  - Il compile le code source Java directement en code machine.
  - Il a été remplacé par l'Android Runtime (ART) dans les versions Android ultérieures.
- Lorsque la fonction `getReadableDatabase ()` est appelée sur une base de données SQLite ?
  - La base de données sera ouverte en mode lecture, ce qui empêcherait toute modification.
  - Le fichier de base de données existant est supprimé et un nouveau fichier est créé en mode lecture.
  - La base de données sera ouverte et rendue disponible pour les opérations de lecture/écriture.

#### Exercice 02 : (4 points)

Au début d'Android, le moteur d'exécution principal était Dalvik, qui compile le bytecode en code machine au moment de l'exécution, selon les besoins. L'un des principaux inconvénients de Dalvik est la longueur du temps d'exécution. Pour résoudre ce problème, ART a été introduit.

- Quelle est la première version d'Android qui utilise l'ART ? (1 point)  
Android KitKat 4.4
- Quelle est la différence fondamentale entre l'ART et Dalvik ? (1 point)  
Art utilise la compilation à l'avance (Ahead-of-time) qui compile les fichiers d'application lors de l'installation de l'application. En revanche, le runtime Dalvik utilise la compilation juste à temps (Just-In-Time) qui compile l'application étape par étape selon les besoins.
- Quel est le nom de l'approche du runtime ART ? (1 point)  
Ahead of time (AOT)

4. Au début de l'ART, l'utilisateur est confronté à une erreur d'espace mémoire insuffisant même si l'espace libre est grand que la taille de l'application (e.g. espace libre = 60 mb et taille d'application = 55 mb), quelle est la cause principale de cette erreur ? (1 point)

En raison du processus de compilation effectué par le système Android lors de l'installation, l'application nécessite un espace supplémentaire sur le disque dur. Cet espace n'est généralement pas pris en compte dans la taille de l'application.

### Exercice 03 : (6 points)

Vous souhaitez développer une application Android composée de deux activités.

- La première activité « Activity1 » comporte un EditText dans lequel l'utilisateur peut saisir son nom et un bouton qui envoie le nom saisi à la deuxième activité.
- La deuxième activité « Activity2 » contient un TextView qui affiche le nom de l'utilisateur reçu de la première activité et un bouton qui permet à l'utilisateur de revenir à la première activité.

#### Questions :

1. Quel est le nom de chaque fichier XML associé à chaque activité ?

Le fichier XML associé à l'activité 1 sera nommé activity\_1.xml.

Le fichier XML associé à l'activité 2 sera nommé activity\_2.xml.

2. Si vous connaissez les identifiants des éléments de l'activité 1 (EditText, bouton) et de l'activité 2 (TextView, bouton), qui sont respectivement et\_name, bnt1, tv\_welcome et bnt2, comment pouvez-vous obtenir ces éléments dans le code Java ? (1 point)

EditText name = FindViewById(R.id.et\_name);

Button btn = FindViewById(R.id.bnt1);

TextView welcome = FindViewById(R.id.tv\_welcome);

Button brn2 = FindViewById(R.id.bnt2);

3. Comment s'appelle la fonction Android permettant d'associer une action de clic à un bouton ?

setOnClickListener() ; (1 point)

4. Ecrivez le code qui démarre l'activité 2 et transmet la valeur du nom entre l'activité 1 et l'activité 2 (expliquez le code). (1 point)

Intent intent = new Intent (Activity1.this;Activity2.class); créer une intention pour aller à l'activité 2  
Intent.putExtra("name";name.getText().toString()); ajouter le nom inséré dans l'EditText.  
startActivity(intent); démarrer l'activité.

5. Écrivez le code qui vous permet d'obtenir la valeur envoyée à la activité 2 de la question précédente (expliquez le code). (1 point)

Intent intent = getIntent() ; retourne l'intention envoyée à l'activité lors du démarrage.

String name = Intent.getStringExtra("name"); recherche un paramètre dans l'intention avec la clé "name".

welcome.setText(name); afficher le nom dans le TextView

6. Écrivez le code du bouton de l'activité 2 pour revenir à l'activité 1 (expliquez le code). (1 point)

Intent intent = new Intent(Activity2.this;Activity1.class);  
startactivity(intent);

## Exercice 04 : (6 points)

Ce code représente une classe SQLite qui gère les requêtes de base de données. Ce code contient quelques erreurs. Vous devrez :

1. Identifier et corriger les erreurs.

1 → " Text, " (espace avant TEXT).

2 → db.execSQL (.....);

3 → this.getWritableDatabase ();

4 → OnUpgrade (.....);

2. Décrire le fonctionnement de ce code.

Ce code a deux fonctionnalités principales :

- Il crée une table nommée " my\_table " dans la base de données " my\_database.db ".
- Permettre l'insertion de données dans la table créée.

```
public class MyDatabaseHelper extends SQLiteOpenHelper {

    private static final String DB_NAME = "my_database.db";
    private static final int DB_VERSION = 1;
    private static final String TABLE_NAME = "my_table";
    private static final String COL_ID = "id";
    private static final String COL_NAME = "name";
    private static final String COL_AGE = "age";

    private SQLiteDatabase db;

    public MyDatabaseHelper(Context context) {
        super(context, DB_NAME, factory: null, DB_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String createTableQuery = "CREATE TABLE " + TABLE_NAME + " (" +
            COL_ID + " INTEGER PRIMARY KEY, " +
            COL_NAME + " TEXT, " + 1
            COL_AGE + " INTEGER);";
        db.execSQL(createTableQuery); 2
    }

    public void insertData(String name, int age) {
        db = this.getWritableDatabase(); 3
        ContentValues values = new ContentValues();
        values.put(COL_NAME, name);
        values.put(COL_AGE, age);
        db.insert(TABLE_NAME, nullColumnHack: null, values);
    }
}
```

**Bon courage !**