

Examen National de Fin de Formation
Session de Juin 2025

Examen de Fin de Formation (Epreuve de synthèse)

Éléments de correction

Secteur :	Digital et Intelligence Artificielle	Niveau :	Technicien Spécialisé
Filière :	Développement Digital option Applications Mobiles		
Variante	V2	Durée :	4H00
		Barème	100

Consignes et Précisions aux correcteurs :

Veuillez respecter impérativement les consignes suivantes :

- Le corrigé est élaboré à titre indicatif,
- Eviter de sanctionner doublement le stagiaire sur les questions liées,
- Pour toutes les questions de synthèse et de compréhension le correcteur s'attachera à évaluer la crédibilité et la pertinence de la réponse du stagiaire. Et à apprécier toute réponse cohérente du stagiaire,
- Le stagiaire n'est pas tenu de fournir des réponses aussi détaillées que celles mentionnées dans le corrigé,
- Pour les exercices de calcul :
 - Prendre en considération la méthode de calcul correcte (formule et relation de calcul correcte) même si le résultat final de calcul est faux
 - Le résultat final correct non justifié ne doit pas avoir la totalité de la note.
- En cas de suspicion d'erreur au niveau du corrigé, prière de contacter la Division de Conception des Examens.

Détail du Barème :

N° Des Dossiers	Travaux à réaliser	Barème
Partie Théorique		
1	Généralités sur Android	15 points
2	Théorie Kotlin	25 points
Total Partie Théorique		/40points
Partie Pratique		
3	Développement Android	45 points
4	Développement Cross-plateforme	15 points
Total Partie Pratique		/60points
Total Général		/100points

Dossier 1 : Généralités sur Android (15 pts)

1. Mécanisme afin de détecter des événements comme la Mise en Mode avion ?
broadcastReceiver (b)
2. menu à gauche/SideBar
NavigationDrawer (c)
3. l'autorisation d'accès aux SMS ?
Manifest.xml (c)
4. Dans Hilt, pour définir une durée de vie dépendance équivalente à l'application entière ?
@Singleton (b)
5. composant **Jetpack** Compose pour afficher une **image**
Image (b)

Dossier 2 : Théorie Kotlin (25 pts)

1. Créer la classe **Tablette** avec ses propriétés. (5 pts)

```
//classe Tablette sur 5 pts
class Tablette (val id:Int,
                val marque:String,
                val modele:String,
                val prix:Double,
                var quantite:Int){}
```

2. Créer la classe **BureauVente** avec ses **propriétés** et **3 méthodes** mentionnées ci-dessus. (20 pts)

NB : Notez que dans cet exercice, on NE VOUS demande AUCUNE interface graphique

```
//classe BureauVente, signature sur 5 pts et chaque méthode sur 5 pts
class BureauVente(val nom:String, val adresse:String, val
                  listTablettes:ArrayList<Tablette>) {
    fun ajouterTablette(tablette: Tablette) {
        listTablettes.add(tablette)
    }
    fun rechercherTablette(id: Int): Tablette? {
        if (listTablettes.filter { it.id == id }.size == 1)
            return listTablettes.filter { it.id == id }[0]
        return null
    }
    fun supprimerTablette(id: Int) {
        listTablettes.removeIf { it.id == id }
    }
}
```

Filière	DDOAM	Variante	V2	Page	Page 2 sur 6
Corrigé	Examen Fin de Formation	Session	Juin 2025		

Dossier 3 : Développement Android (45 pts)

- ✓ NB. Des classes, interfaces et fonctions sont supposées déjà existantes, donc il n'EST PAS demandé de les implémenter. Se référer à l'énoncé pour cela

1- **Bouton Add** (9 pts)

```

val name = findViewById<EditText>(R.id.editTextName).text.toString().trim()
val priceStr = findViewById<EditText>(R.id.editTextPrice).text.toString().trim()
val image = findViewById<EditText>(R.id.editTextImage).text.toString().trim()
val isWR = findViewById<CheckBox>(R.id.isWaterResistant).isChecked
val price = priceStr.toDouble()
val smartWatch = SmartWatch(0, name, price, isWR, image)
// NB : Abréviation retrofit = Retrofit.baseUrl(...) acceptée
val retrofit = Retrofit.Builder()
    .baseUrl("https://watchapi.com/")
    .addConverterFactory(GsonConverterFactory.create())
    .build()
val apiService = retrofit.create(ApiService::class.java)
apiService.addSmartWatch(smartWatch).enqueue(object : Callback<AddResponse> {
    override fun onResponse(call: Call<AddResponse>, response:
        Response<AddResponse>) {
        // Code isSuccessful et Toast non obligatoire
        if (response.isSuccessful) {
            Toast.makeText(this@MainActivity, response.body()?.response.message?: "Erreur
            d'ajout", Toast.LENGTH_LONG).show()}}})

```

2- **Liste des smartWatches**

a- (9 pts)

NB : le candidat a la possibilité d'utiliser des abréviations pour les balises et attributs xml, aussi il n'est pas obligé d'inclure les namespaces et très longues balises comme xmlns...

NB : D'autres alternatives d'alignement existe aussi comme **ConstraintLayout** et **RelativeLayout**

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">

    <LinearLayout
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/pimage"
            android:layout_width="85dp"
            android:layout_height="85dp"
            android:layout_margin="8dp" />

        <Switch
            android:id="@+id/isWT"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:clickable="false"
            android:text="Water Resist"
            android:layout_margin="5dp" />

    </LinearLayout>

```

Filière	DDOAM	Variante	V2	Page	Page 3 sur 6
Corrigé	Examen Fin de Formation	Session	Juin 2025		

```

<LinearLayout
    android:layout_marginTop="30dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:orientation="vertical">

    <TextView
        android:id="@+id/pname"
        style="@style/watchStyle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        />

    <TextView
        android:id="@+id/pprice"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="17sp" />

</LinearLayout>
</LinearLayout>

```

b- (9 pts)

```

class MyAdapter(context: Context, private val smartWatchList: List<SmartWatch>) :
    ArrayAdapter<SmartWatch>(context, R.layout.item_watch, smartWatchList) {

    override fun getView(position: Int, convertView: View?, parent: ViewGroup): View {
        val view: View = convertView ?: LayoutInflater.from(context)
            .inflate(R.layout.item_watch, parent, attachToRoot false)

        val imageView = view.findViewById<ImageView>(R.id.pimage)
        val nameTextView = view.findViewById<TextView>(R.id.pname)
        val priceTextView = view.findViewById<TextView>(R.id.pprice)
        val isWR = view.findViewById<Switch>(R.id.isWT)
        val currentSW = getItem(position)

        currentSW?.let {
            nameTextView.text = it.name
            priceTextView.text = "${it.price} dh"
            isWR.isChecked = it.isWaterResistant
            Glide.with(context)
                .load(it.imageUrl)
                .into(imageView)
        }

        return view
    }
}

```

c- (9 pts)

NB : Abréviation **Retrofit.baseUrl(.....)** acceptée

Filière	DDOAM	Variante	V2	Page	Page 4 sur 6
Corrigé	Examen Fin de Formation	Session	Juin 2025		

```

var swartwatches: List<SmartWatch>
val listView = findViewById<ListView>(R.id.listView)
val retrofit = Retrofit.Builder()
    .baseUrl("https://watchapi.com/")
    .addConverterFactory(GsonConverterFactory.create())
    .build()
val apiService = retrofit.create(ApiService::class.java)

apiService.getAll().enqueue(object : Callback<List<SmartWatch>> {
    override fun onResponse(call: Call<List<SmartWatch>>, response: Response<List<SmartWatch>>) {
        if (response.isSuccessful) {
            swartwatches = response.body()?.reversed() ?: emptyList()
            listView.adapter = MyAdapter(context = this@MainActivity, swartwatches)
        }
    }
})

```

3- switch « Dark Mode »

//Code db = et themeDao = ... à mettre ou bien dans a/ ou bien b/ pas obligatoirement les deux
a- (4.5 pts)

```

val db = AppDatabase.getDatabase(applicationContext)
val themeDao = db.themePreferenceDao()
val themeCheckBox = findViewById<CheckBox>(R.id.mode)
themeCheckBox.setOnCheckedChangeListener { _, isChecked ->
    themeDao.setTheme(UserPrefs(isDarkMode = isChecked))
    changerTheme(isChecked)
}

```

b- (4.5 pts)

```

val db = AppDatabase.getDatabase(applicationContext)
val themeDao = db.themePreferenceDao()
val isDarkMode = themeDao.getSavedTheme()?.isDarkMode ?: false
changerTheme(isDarkMode)

```

Dossier 4 : Développement cross-platform (15 pts)

Notez que le stagiaire a aussi la possibilité d'implémenter la logique de calcul dans une fonction indépendante, veuillez l'accepter

Le bouton "Calculer" affiche le total de la commande

- Création de l'interface : **5 pts**
- Gestion des boutons radio : **5 pts**
- Calcul et affichage du total à payer : **5 pts**

```
enum Taille { taille9, taille10, taille11 }
```

```
class _MyHomePageState extends State<MyHomePage> {
    final _quantite = TextEditingController();
    Taille? _taille = Taille.taille9;
    String _resultat = '';

```

```

@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            backgroundColor: Theme.of(context).colorScheme.inversePrimary,
            title: const Text(
                'Commander Tablette',
            ),),
        body: Center(
            child: Column(
                children: [

```

Filière	DDOAM	Variante	V2	Page	Page 5 sur 6
Corrigé	Examen Fin de Formation	Session	Juin 2025		

```

Padding(
  padding: const EdgeInsets.only(left: 30.0, top: 30, right: 30, bottom:15),
  child: Row(
    children: [
      const Text('Taille écran'),
      const SizedBox(
        width: 30,),
      Radio<Taille>(
        value: Taille.taille9,
        groupValue: _taille,
        onChanged: (value){
          setState(() {
            _taille = value;
          }); },),
      const Text('9"'),
      Radio<Taille>(
        value: Taille.taille10,
        groupValue: _taille,
        onChanged: (value){
          setState(() {
            _taille = value;
          }); }, ),
      const Text('10"'),
      Radio<Taille>(
        value: Taille.taille11,
        groupValue: _taille,
        onChanged: (value){
          setState(() {
            _taille = value;
          }); }, ),
      const Text('11"'),
    ], ),),
  Padding(
    padding: const EdgeInsets.only(left: 30.0, top: 5, right: 30, bottom:15),
    child: Row(
      children: [
        const Text('Quantité : '),
        SizedBox(
          width: 300,
          child: TextField(
            controller: _quantite,
          ),), ],),),
  Padding(
    padding: const EdgeInsets.all(5),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        ElevatedButton(
          onPressed: (){
            int quantite = int.parse(_quantite.text);
            double prix = 3000;
            if(_taille == Taille.taille10)
              prix = 4000;
            else if(_taille == Taille.taille11)
              prix = 5000;
            double resultat = quantite * prix;
            setState(() {
              _resultat = 'Total à payer ${resultat.toString()} DH';
            });
          },
          child: const Text('CALCULER')
        ), ],),),
  Padding(
    padding: const EdgeInsets.all(10),
    child: Text(_resultat,),
  ],),),);}}

```

Filière	DDOAM	Variante	V2	Page	Page 6 sur 6
Corrigé	Examen Fin de Formation	Session	Juin 2025		