

**Examen National de Fin de Formation  
Session de Juin 2025**

**Examen de Fin de Formation (Epreuve de synthèse)**

**Eléments de correction**

<b>Secteur :</b>	<b>Digital et Intelligence Artificielle</b>	<b>Niveau :</b>	<b>Technicien Spécialisé</b>
<b>Filière :</b>	<b>Développement Digital option Applications Mobiles</b>		
<b>Variante</b>	<b>V1</b>	<b>Durée :</b>	<b>4H00</b>
		<b>Barème</b>	<b>100</b>

**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
<b>Partie Théorique</b>		
<b>1</b>	Généralités sur Android	<b>15 points</b>
<b>2</b>	Théorie Kotlin	<b>25 points</b>
<b>Total partie Théorique</b>		<b>/40points</b>
<b>Partie Pratique</b>		
<b>3</b>	Développement Android	<b>45 points</b>
<b>4</b>	Développement Cross-plateforme	<b>15 points</b>
<b>Total partie Pratique</b>		<b>/60points</b>
<b>Total Général</b>		<b>/100points</b>

**Dossier 1 : Généralités (15 pts) Chaque question sur 3 pts :**

1. Quel fichier utiliser pour définir les **dépendances** du projet en Kotlin DSL ?  
build.gradle.kts (b)
2. Détecter des événements comme l'arrivée d'un **appel téléphonique/niveau de batterie** ?  
BroadcastReceiver (c)
3. **menu en haut** (menu horizontal) avec des éléments cliquables  
ActionBar/Appbar/ Toolbar (d)
4. Dans **Jetpack** Compose, quelle fonction est utilisée pour définir un élément **cliquable** ?  
Modifier.clickable (b)
5. composant utilisé pour créer un conteneur avec un arrière-plan ?  
Surface (C)

**Dossier 2 : Théorie Kotlin (25 pts)**

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

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

2. Créer la classe **Magasin** 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 Magasin, signature sur 5 pts et chaque méthode sur 5 pts
class Magasin(val nom:String, val adresse:String, val listPhones:ArrayList<Smartphone>)
{
    fun ajouterSmartphone(smartphone: Smartphone) {
        listPhones.add(smartphone)
    }
    fun rechercherSmartphone(id: Int): Smartphone? {
        if (listPhones.filter { it.id == id }.size == 1)
            return listPhones.filter { it.id == id }[0]
        return null
    }
    fun supprimerSmartphone(id: Int) {
        listPhones.removeIf { it.id == id }
    }
}
```

Filière	DDOAM	Variante	V1	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 haveFP = findViewById<CheckBox>(R.id.HaveFP).isChecked
val price = priceStr.toDouble()
val pc = PC(0, name, price, haaveFP, image)
// NB : Abréviation retrofit = Retrofit.baseUrl(...) acceptée
val retrofit = Retrofit.Builder()
    .baseUrl("https://myapi.com/")
    .addConverterFactory(GsonConverterFactory.create())
    .build()

val apiService = retrofit.create(ApiService::class.java)
apiService.addData(pc).enqueue(object : Callback<AddResponse> {
    override fun onResponse(call: Call<AddResponse>, response:
        Response<AddResponse>) {
        // Code isSuccessful et Toast NON obligatoire, seulement le Post
        if (response.isSuccessful) {
            Toast.makeText(this@MainActivity,
                response.body()?.response.message?:"Erreur d'ajout",
                Toast.LENGTH_LONG).show()})})

```

**2- Liste des PCs**

## 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 **ConstarintLayout** et **RelativeLayout**

```

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

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <ImageView
            android:id="@+id/pimage"
            android:layout_width="118dp"
            android:layout_height="118dp"
            android:layout_margin="18dp"
        />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_marginLeft="28dp">

```

Filière	DDOAM	Variante	V1	Page	Page 3 sur 6
Corrigé	Examen Fin de Formation	Session	Juin 2025		

```

        <TextView
            android:id="@+id/pname"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            style="@style/nameStyle"
        />
        <TextView
            android:id="@+id/pprice"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="17sp"/>
        <CheckBox
            android:id="@+id/HaveFP"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Have FPS"
            android:clickable="false"/>
    </LinearLayout>
</LinearLayout>
</LinearLayout>

```

b- (9 pts)

```

class Adapter(context: Context, private val pcList: List<Pc>) :
    ArrayAdapter<Pc>(context, R.layout.products_item, pcList) {

    override fun getView(position: Int, convertView: View?, parent: ViewGroup): View {
        val view: View = convertView ?: LayoutInflater.from(context)
            .inflate(R.layout.products_item, 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 haveFpCheckBox = view.findViewById<CheckBox>(R.id.HaveFP)
        val pc = getItem(position)

        pc?.let {
            nameTextView.text = it.name
            priceTextView.text = "${it.price} dh"
            haveFpCheckBox.isChecked = it.haveFPS
            Glide.with(context)
                .load(it.image)
                .into(imageView)
        }

        return view
    }
}

```

c- (9 pts)

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

```

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

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

```

Filière	DDOAM	Variante	V1	Page	Page 4 sur 6
Corrigé	Examen Fin de Formation	Session	Juin 2025		

### 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 themeSwitch = findViewById<Switch>(R.id.modeSwitch)
themeSwitch.setOnCheckedChangeListener { _, isChecked ->
    themeDao.saveThemePreference(ThemePreference(isDarkMode = isChecked))
    setAppTheme(isChecked)
}
```

b- (4.5 pts)

```
val db = AppDatabase.getDatabase(applicationContext)
val themeDao = db.themePreferenceDao()
val isDarkMode = themeDao.getThemePreference()?.isDarkMode ?: false
setAppTheme(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 Stockage { STOCKAGE64, STOCKAGE128, STOCKAGE256 }
```

```
class _MyHomePageState extends State<MyHomePage> {
    final _quantite = TextEditingController();
    Stockage? _stockage = Stockage.stockage64;
    String _resultat = '';

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                backgroundColor: Theme.of(context).colorScheme.inversePrimary,
                title: const Text('Commander Smartphone'),
            ),
            body: Center(
                child: Column(
                    children: [
                        Padding(
                            padding: const EdgeInsets.only(left: 30.0, top: 30, right: 30, bottom:15),
                            child: Row(
                                children: [
                                    const Text('Quantité : '),
                                    SizedBox(
                                        width: 300,
                                        child: TextField(),
                                        controller: _quantite,
                                    ),
                                ],
                            ),
                        ),
                        Padding(
                            padding: const EdgeInsets.only(left: 30.0, top: 0, right: 30, bottom:15),
                            child: Row(
                                children: [
                                    const Text('Stockage'),
                                    const SizedBox(
                                        width: 30,
                                    ),
                                ],
                            ),
                        ),
                    ],
                ),
            ),
        );
    }
}
```

Filière	DDOAM	Variante	V1	Page	Page 5 sur 6
Corrigé	Examen Fin de Formation	Session	Juin 2025		

```

Radio<Stockage>(
  value: Stockage.STOCKAGE64,
  groupValue: _stockage,
  onChanged: (value){
    setState(() {
      _stockage = value;
    });
  },
),
const Text('64G',),
Radio<Stockage>(
  value: Stockage.STOCKAGE128,
  groupValue: _stockage,
  onChanged: (value){
    setState(() {
      _stockage = value;
    });
  },
),
const Text('128G',),
Radio<Stockage>(
  value: Stockage.STOCKAGE256,
  groupValue: _stockage,
  onChanged: (value){
    setState(() {
      _stockage = value;
    });
  },
),
const Text('256G',),
],),),
Padding(
  padding: const EdgeInsets.all(5),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      ElevatedButton(
        onPressed: (){
          int quantite = int.parse(_quantite.text);
          double prix = 5000;
          if(_stockage == Stockage.STOCKAGE128)
            prix = 6000;
          else if(_stockage == Stockage.STOCKAGE256)
            prix = 7000;

          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	V1	Page	Page 6 sur 6
Corrigé	Examen Fin de Formation	Session	Juin 2025		