

Partie 1 : QCM 0.5pt/question

Question1

1. Parmi les assertions suivantes, lesquelles sont fausses

- ☒ a. Une classe peut hériter de plusieurs classes
- ☐ b. Une classe peut implémenter plusieurs interfaces
- ☒ c. Une classe est final par défaut
- ☒ d. Une interface est final par défaut

Explication :

- a. En Java, une classe ne peut pas hériter de plusieurs classes (faux). Cela diffère de l'héritage multiple en interfaces.
- b. Une classe peut implémenter plusieurs interfaces (vrai).
- c. Une classe n'est pas **final** par défaut. Si elle l'était, elle ne pourrait pas être étendue (faux).
- d. Une interface n'est pas **final** par défaut. Les interfaces sont destinées à être implémentées (faux).

Question2

1. Comment corriger les erreurs dans le code suivant ?

```
1 val myString: String? = "my super string"
2 println(myString.length)
```

- ☒ a. myString est nullable, faut mettre ? avant d'appeler la fonction length
- ☐ b. Il n'y a pas d'erreurs, relance Android Studio
- ☐ c. La variable est immutable et elle a déjà une valeur, le ? est inutile
- ☐ d. Ajouter !! pour forcer le compilateur à considérer myString comme non nulle

Explication :

- a. En Java, une classe ne peut pas hériter de plusieurs classes (faux). Cela diffère de l'héritage multiple en interfaces.
- b. Une classe peut implémenter plusieurs interfaces (vrai).

- c. Une classe n'est pas `final` par défaut. Si elle l'était, elle ne pourrait pas être étendue (faux).
- d. Une interface n'est pas `final` par défaut. Les interfaces sont destinées à être implémentées (faux).

Question3

1. En Kotlin, une variable immutable

- ☐ a. Est une variable dont sa valeur peut changer
- ☒ b. Est une variable dont sa valeur ne peut pas changer
- ☐ c. Peut avoir des valeurs de types différents
- ☐ d. Peut avoir plusieurs valeurs mais toujours du même type

Explication :

- En Kotlin, une variable déclarée avec `val` est immutable, ce qui signifie que sa valeur ne peut pas changer une fois qu'elle est initialisée

Partie 2: Questions à réponses ouvertes

Question1

Compléter la colonne de droite du tableau suivant en ajoutant une définition des termes de la colonne de gauche. (6 pts)

Terme	Permet de:
val	Définir des variables immuables (ne peuvent pas être modifiées après initialisation)
var	Définir des variables mutables (peuvent être modifiées après initialisation)
class	Définir une structure pour créer des objets (modèles pour les objets)
interface	Définir un contrat que d'autres classes peuvent implémenter (méthodes sans corps)
open	Permettre qu'une classe ou méthode soit surclassable (override) dans les classes enfants
fun	Définir une fonction ou une méthode
adapter	Connecter et convertir des données à une vue (dans les listes ou recycler views)
Intent	Permettre la communication entre les composants de l'application (démarrer une activité, envoyer des données)
Manifest	Déclarer les composants de l'application et les permissions requises
startActivity	Démarrer une nouvelle activité
setContentView	Définir la vue (layout) pour une activité
findViewById	Trouver une vue dans le layout à l'aide de son ID

Question2

Quelle est la différence entre une classe et une interface ? (2 pt)

- La différence principale entre une classe et une interface est que la classe est un modèle complet pour créer des objets, avec des implémentations complètes de méthodes, tandis qu'
- une interface est un contrat qui définit des méthodes sans corps.
- Les classes peuvent hériter d'une seule autre classe mais peuvent implémenter plusieurs interfaces.

Question3

À quoi sert le mot clé `lateinit` en Kotlin, quand peut-on l'utiliser et quelle précaution doit-on prendre en l'utilisant ? (2pts)

- Le mot clé `lateinit` est utilisé pour déclarer qu'une variable sera initialisée plus tard. Il ne peut être utilisé qu'avec des types non primitifs et non nullables. Il faut s'assurer que **la variable est initialisée avant son utilisation**, sinon une exception sera levée.

Question4

Quelle est la différence entre une **activité** et un **fragment** ? (2 pt)

- Une activité est un composant d'application qui fournit un écran avec une interface utilisateur pour que l'utilisateur interagisse.
- Un fragment, en revanche, est une portion modulaire d'une activité, qui a son propre cycle de vie, reçoit ses propres événements d'entrée et peut être ajouté ou retiré lors de l'exécution de l'activité.

Question5

À quoi sert un adaptateur ? (1 pt)

- Un adaptateur en Android sert de pont entre une vue (comme `ListView` ou `RecyclerView`) et les données pour cette vue. Il convertit les données en vues ou en éléments de la liste que l'utilisateur peut voir et avec lesquels il peut interagir.

Question6 ???

Identifiez et commentez les erreurs du programme suivant (3pts):

[Le texte ici est partiellement obscurci et illisible en raison du pli sur le papier. Ce qui est visible et lisible est transcrit ci-dessous.]

```
1 fun myVar(a: Int)
2 class A(val a: Int)
3 class B(b: String) : A(b)
```

- ☐ 1. [illisible]
- ☒ 2. Variable ou déjà définie et finira revient le String et définir retour que int
- ☐ 3. [illisible]
- ☒ 4. La fonction return String et doit retourner un int a doit revenir du int (?)

