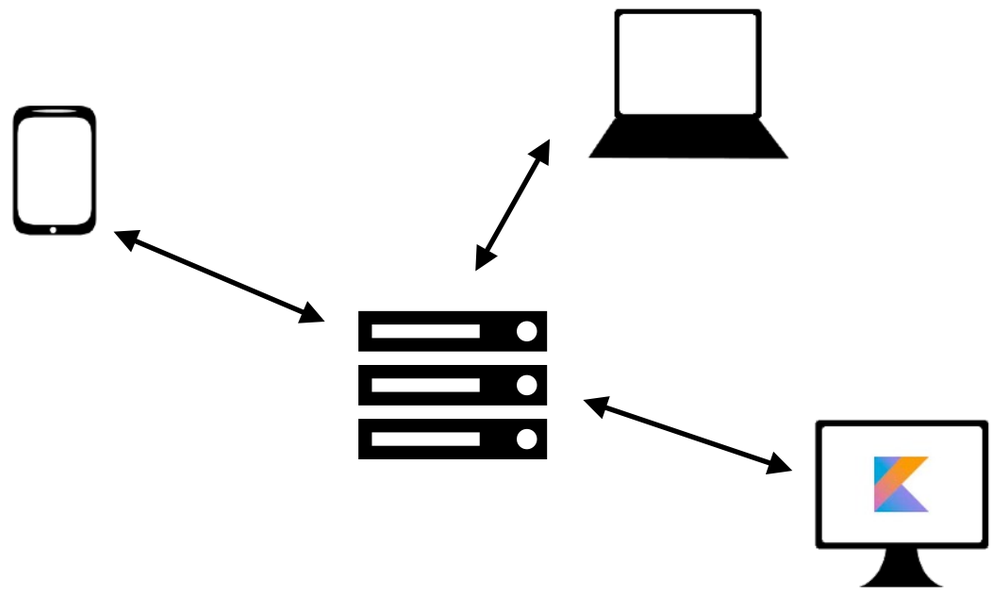

Kotlin

Macha DA COSTA

Programmation Asynchrone

- Problème de la programmation asynchrone
- *Coroutines* en Kotlin
- Implémentation des *coroutines*
- *Async* et *Await* en Kotlin
- *Yield* en Kotlin
- Extension réactive en Kotlin

Programmation Asynchrone

Problème de la programmation asynchrone

- Principe : exécuter plusieurs portions de code en tâche de fond, en parallèle
- Mécanisme : *Thread* en arrière plan
- Problème : bloquer l'app., provoquer des goulots d'étranglement empêchant l'app. de passer à l'échelle supérieur

Référence [] : kotlinlang.org: async-programming.html

Programmation Asynchrone

Coroutines en Kotlin

- Principe : une unité de traitement qui s'apparente à une routine (entité informatique en capsulant une portion de code)

Sortie d'une routine : fin à la routine

Sortie de la coroutine : suspension du traitement (avant signalement de la reprise) accompagné d'une transmission de données

Référence [] : kotlinlang.org: coroutines-overview.html

Programmation Asynchrone

Implémentation des *coroutines*

```
fun main() = runBlocking { // this: CoroutineScope
    launch { // launch a new coroutine and continue
        delay(1000L) // non-blocking delay for 1 second (default time unit is ms)
        println("World!") // print after delay
    }
    println("Hello") // main coroutine continues while a previous one is delayed
}
```


Référence [] : kotlinlang.org: coroutines-basics.html

Programmation Asynchrone

Async et *Await* en *Kotlin*

- `Async` : Job asynchrone, peut bloquer d’autres exécutions
- `Await` : à combiner avec `Async` afin que l’exécution du code soit normal
- Principe : `Await` garantit que l'exécution ira jusqu'à ce que la fonction soit complètement exécutée

Référence [] : kotlinlang.org: coroutine async await

Programmation Asynchrone

Yield en *Kotlin*

```
abstract suspend fun yield(value: T)
```

- Principe : Donne une valeur à l'itérateur en cours de construction et s'interrompt jusqu'à ce que la valeur suivante soit demandée.

Référence [] : kotlinlang.org: sequence scope yield.html

Programmation Asynchrone

Extensions réactives en *Kotlin* 📎 *reactive extension*

- Principe : ReactiveX est un ensemble d'outils permettant aux langages de programmation impératifs de fonctionner sur des séquences de données, que les données soient synchrones ou asynchrones.
- Mécanisme : Il fournit un ensemble d'opérateurs de séquence qui opèrent sur chaque élément de la séquence. Il s'agit d'une implémentation de la programmation réactive et fournit un plan pour les outils à implémenter dans plusieurs langages de programmation.

Référence [] : https://en.wikipedia.org/wiki/ReactiveX

Conclusion

- Problème de la programmation asynchrone
- *Coroutines* en *Kotlin*
- Implémentation des *coroutines*
- *Async* et *Await* en *Kotlin*
- *Yield* en *Kotlin*
- Extension réactive en *Kotlin*

THE EXPERT
AT ANYTHING
WAS ONCE
A BEGINNER.

_ Helen Hayes

Références

Kotlin for Android

- TRY Kotlin
- Kotlin Workshop on Github: Slides and Questions
- https://antonioleiva.com/free-kotlin-android-course/
- ChillCoding.com : Introduction à Kotlin
- ChillCoding.com : Configurer Kotlin dans un projet Android Studio

Library

- ChillCoding.com : Utiliser des bibliothèques graphiques Kotlin dans un projet Android

Fonction d’extension

- Odelia Technologies : Les fonctions d'extension de Kotlin

Kotlin in videos

- Jake Wharton and Kotlin (DEC 2015)
- Tue Dao & Christina Lee on The Road to Kotlintown (KotlinConf 2017)
- Introduction to Kotlin Google I/O '17