

**PROGRAMME DÉTAILLÉ**

**Code : 6990**

Les bonnes pratiques Java

## **Le cycle de vie d’un logiciel : la qualité du code source**

## Le mode projet : construire du logiciel

## Le mode maintenance : faire vivre le logiciel, les différents types d’intervention

## Prérequis aux interventions : gestion de version, Nécessité de tests unitaires automatisés, présentation de JUNIT

## Conclusion

## **Bonnes pratiques générales : les grands principes**

## Les grands principes SOLID

## Les différentes familles de pattern

## Le pattern singleton : la donnée unique accédée par tous

## **Bonnes pratiques générales : zoom sur le couplage faible**

## Découpage d’une application en couches. Vérifier son découpage

## Le modèle MVC

## La couche physique et les dao : un exemple d’utilisation d’interface java

## Comment casser l’héritage

## **Organiser et rendre son code lisible**

## Gestion des packages

## Règles de nommage

## Normes de codage

## Formatage du code. Visibilité des attributs et méthodes

## Documenter son code : la Javadoc

## **Bien choisir ses classes**

## Principes généraux

## Les énumérations java

## Zoom sur la classe String

## Le choix des collections : ArrayList, HashSet, HashMap. Utilisation de leurs interfaces

## Redéfinir Equals() et hashcode()

## **Bien écrire ses méthodes**

## Rappel du passage par valeur et de son impact en Java

## Complexité cyclomatique

## Une bonne gestion des erreurs : créer ses classes d’exception

**Bonnes pratiques EcoDev**

## EcoDev ?

## Constats

## Utiliser GRAAL VM

## Utiliser Creedengo

## Utiliser les caches

## Fermer les ressources

## Dimensionner correctement

## Utiliser Visual VM

## **Bonnes pratiques Lambdas**

## Historique des lambdas expressions

## Constats et bonnes pratiques

## **Bonnes pratiques sur le theading**

## Virtual Threads : un peu d’histoire...

## Virtual Threads vs Threads classics

## Virtual Threads : Programmation Concurrente et Asynchrone

## Solutions ?

## Optimisation JVM pour le theading