**Sujet : en s’inspirant d’une application ou un site connu, réaliser en Java l’implémentation “serveur” de cette dernière (sans interface graphique)**

Par groupes de 1 à 3 personnes les élèves doivent trouver un sujet qui respecte les différents critères ci-dessous et le faire valider par l’intervenant référent.

Les livrables attendus sont :

* Le code source du projet (git apprécié, github ou gitea)
  + doit être lisible, clair, commenté, compiler et s’exécuter (Java 21)
  + doit contenir la liste des personnes ayant participé au projet
  + doit contenir un fichier README qui décrit votre projet, comment l’installer, quels sont les prérequis techniques et comment l’utiliser globalement
* Une présentation orale
  + accompagnée d’un support de présentation (PowerPoint, PDF, Canva, etc.)
    - Numéros de pages sur les diaporamas
    - Sommaire
    - Quantité de texte limitée
    - Pas de code à l’écran
  + d’une durée de 10mn (max) avec 5mn de questions
    - présenter le projet (quelle est l’application/site choisi et pourquoi)
    - quels sont les différents choix qui ont été faits (choix techniques, choix d’organisation, choix d’architecture, choix de librairies, etc.)
    - comment le travail a-t-il été réparti entre les membres du groupe
    - quelles sont les fonctionnalités développées
    - démonstration de quelques fonctionnalités principales
    - qu’est-ce qu’il manque à votre projet, quelles sont les difficultés rencontrées, quelles sont les pistes d’amélioration

Les différents critères sont :

* visibilité des classes, attributs et méthodes justifiée
* les exceptions sont maitrisées, des exceptions personnalisées sont utilisées
* abstraction, static et autres modifiers utilisés à bon escient
* organisation du code
  + en plusieurs packages
  + le nom des classes est compréhensible et en anglais
  + le nom des variables est clair et en anglais
  + le code respecte le même style dans l’ensemble du projet (camelCase, PascalCase, snake\_case, etc.)
* le code comporte des commentaires sur les fonctionnalités/méthodes principales, avec le plus de Javadoc possible
* un maximum de ce qui a été vu en cours et/ou avec la piscine doit être utilisé
  + polymorphisme
  + héritage (extends, implements)
  + collections
  + généricité, etc.
* la performance du code doit avoir été prise en compte (choix des types de boucles, choix des objets, choix des librairies, etc.)

Critères additionnels (non obligatoires) :

* lecture/écriture de fichiers (journalisation/logs, sauvegarde/historique, rapports, etc.)
* dynamisme de l’application (saisir via le terminal des commandes ou choisir des options au travers d’un menu pour utiliser l’application, ScannerIO)
* interface graphique (awt, swing, javafx, etc.)

Le but du projet étant d’aller le plus loin possible aucune librairie n’est interdite, aucun concept Java non plus. Les groupes sont libres d’utiliser tout ce qu’ils ont découverts ou appris par eux-mêmes.

Organisation du code :

* src
  + main
    - Java
      * domain.name.applicationName <= package principal
        + package1
        + package2

Exemple:

* src/main/java
  + com.iobeya.api <= package principal
    - configuration
    - controllers
    - services
    - etc.