Simulation du flux de données

Créez une classe DataEmitter qui émet un flux continu de données (par exemple des nombres entiers aléatoires ou des objets SensorData).

Implémentez un Publisher (selon l'API Flow.Publisher) qui envoie les données à des Subscribers.

Traitement réactif

Implémentez un Subscriber qui :

Reçoit les données du flux,

Applique un traitement (ex. : filtrage, transformation, agrégation),

Affiche les résultats.

Testez le comportement avec un SubmissionPublisher standard.

Ajout des Virtual Threads

Remplacez les exécuteurs classiques par des Virtual Threads (Executors.newVirtualThreadPerTaskExecutor()).

Observez comment les tâches sont exécutées simultanément sans surcharge importante.

Comparaison avec les Streams classiques

Réécrivez le même traitement avec les Java Streams :

Une version séquentielle,

Une version parallèle (parallelStream()).

Comparez les performances avec la version réactive + virtual threads.

Analyse et synthèse

Mesurez le temps d'exécution et la consommation mémoire.

Observez la différence entre :

Le modèle bloquant (threads classiques),

Le modèle réactif (Reactive Streams),

Le modèle massivement concurrent (Virtual Threads).