Université Abderrahmane Mira de Béjaia **mai 2026**
Faculté des Sciences Exactes
Département d'Informatique
ING4
Architectures Logicielles

Série de TP N° 2

Activité 1

Event-Driven Architecture (EDA)

- Comprendre les principes de l'EDA
- Implémenter des événements, producteurs et consommateurs
- Mettre en place une communication asynchrone et une exécution indéterministe
- Simuler un système basé sur des topics (pub/sub)

Implémentation d'une architecture EDA : Système de gestion de commandes

- Vous devez développer un système simplifié de gestion de commandes (EDA_Order_System) basé sur une architecture événementielle.
- Scénario : Lorsqu'une commande est créée, un événement *OrderCreatedEvent* est publié et plusieurs services réagissent automatiquement :
    - Service de paiement
    - Service de notification
    - Service de livraison
- Le projet EDA_Order_System doit être organisé en packages, chacun ayant une responsabilité précise (respecter le principe de séparation des responsabilités)
- Exemple de résultat d'exécution :

    Commande créée : ord-001
    Paiement en cours : ord-001
    Livraison en préparation : ord-001
    Notification envoyée : ord-001

☞ **Travail noté**

Utilisez le brocker Apache Kafka pour implémenter un système de gestion de commandes basé sur une architecture dirigée par les événements (EDA).

Activité 2

Architecture micro-noyau

- Développer une application Java modulaire basée sur une architecture :
    - micro-noyau
    - extensible par plugins
    - chargement dynamique des modules
- Plugins demandés :
    - Calcul
    - Notification
    - Export PDF

Etapes à suivre :

1. **Créer les projets NetBeans**
    - PluginAPI
    - CoreApplication
    - PluginCalcul
    - PluginNotification
    - PluginExportPDF
2. **Développer l'API des plugins**
    - Dans PluginAPI : Créer l'interface commune :

```
public interface Plugin {
String getName() ;
void execute() ;
}
```

    - Cette interface définit le contrat que tous les plugins doivent respecter.
3. **Ajouter PluginAPI comme dépendance**

    Dans :
    - CoreApplication
    - PluginCalcul
    - PluginNotification
    - PluginExportPDF
4. **Développer le micro-noyau (CoreApplication)**
    - Créer :
        1. PluginLoader
            - parcourir le dossier plugins,
            - détecter les fichiers .jar,
            - charger dynamiquement les classes des plugins,
            - créer leurs objets, les retourner dans une liste.
        2. Main
    - Le noyau doit :
        - parcourir le dossier plugins

- charger dynamiquement les fichiers .jar
- instancier les plugins
- exécuter leurs fonctionnalités

5. Implémenter les plugins

Chaque plugin :

- implémente l'interface Plugin
- fournit sa propre fonctionnalité

6. **Générer les fichiers JAR**

Pour chaque plugin :

- Right Click sur le projet: Clean and Build
- NetBeans génère :
  - dist/PluginCalcul.jar
  - dist/PluginNotification.jar
  - dist/PluginExportPDF.jar

7. Créer le dossier plugins

- Dans : CoreApplication, créer le folder: plugins/

8. Copier les plugins

Copier les fichiers :

- PluginCalcul.jar
- PluginNotification.jar
- PluginExportPDF.jar

dans :

- CoreApplication/plugins

9. Exécuter l'application

L'application :

- détecte automatiquement les plugins
- les charge
- exécute leurs méthodes