

# Rapport – Projet 3 : Système de Gestion Bancaire

## 1. Introduction

Ce projet consiste à développer une application de gestion bancaire en Java. L'objectif principal est de mettre en pratique les concepts de la programmation orientée objet, notamment l'abstraction, le polymorphisme, les interfaces, la gestion des exceptions et l'utilisation de collections.

## 2. Objectifs pédagogiques

- Maîtriser l'abstraction et le polymorphisme
- Implémenter des interfaces pour définir des comportements
- Gérer des transactions et leur historique
- Appliquer des patterns de conception (Factory, Service)
- Gérer des exceptions métier personnalisées

## 3. Architecture générale

L'application est organisée en plusieurs packages afin de respecter une architecture claire : model, service, factory, exception et strategy. Chaque package a une responsabilité bien définie.

## 4. Description des classes principales

Compte (abstraite) : représente un compte bancaire générique.  
CompteCourant : autorise un découvert limité.  
CompteEpargne : permet le calcul et l'application des intérêts.  
CompteEntreprise : possède un plafond de transaction.  
Client : représente un client bancaire pouvant posséder plusieurs comptes.  
Transaction : enregistre chaque opération avec date, type et montant.

## 5. Gestion des opérations bancaires

Les opérations de dépôt et de retrait sont gérées directement par les comptes. Les virements sont gérés par la classe BanqueService afin de respecter le principe de responsabilité unique.

## 6. Gestion des exceptions

Des exceptions personnalisées sont utilisées pour gérer les erreurs métier :  
MontantInvalideException  
SoldeInsuffisantException  
LimiteDepasseeException

## 7. Diagramme UML

Un diagramme UML de classes a été réalisé pour représenter les relations entre les différentes classes du système.

## **8. Guide utilisateur**

L'utilisateur peut créer un client, ajouter des comptes, effectuer des dépôts, des retraits, des virements et consulter l'historique des transactions via un menu interactif.

## **9. Scénarios de test**

Des scénarios de test ont été définis pour vérifier le bon fonctionnement des opérations normales et la gestion des cas exceptionnels.

## **10. Conclusion**

Ce projet a permis de consolider les bases de la programmation orientée objet en Java et d'appliquer des bonnes pratiques de conception logicielle.