

Guide Utilisateur – Application de Gestion Bancaire

1- Objectif :

L'application permet de gérer des clients, plusieurs comptes bancaires, effectuer des opérations (dépôts, retraits, virements), et appliquer des règles métier comme le découvert, le plafond ou le taux d'intérêt.

2- Lancer l'application :

1. Ouvrir le projet dans votre IDE (IntelliJ ou autre) ou via le terminal.
2. Exécuter la classe Main.java.
3. L'écran de création de client s'affiche :

===== APPLICATION DE GESTION BANCAIRE =====

--- Création du client ---

Nom :

Prénom :

CIN :

Adresse :

3- Créer un client

- Saisir le nom, prénom, CIN, et adresse.
- Un objet Client est créé et stocke les informations.

4- Créer un compte

- L'application demande :

Type de compte (COURANT / EPARGNE / ENTREPRISE) :

Numéro du compte :

Solde initial :

- Selon le type choisi :

- Compte Courant : saisir le découvert autorisé.
- Compte Épargne : saisir le taux d'intérêt (ex: 0.03 pour 3%).
- Compte Entreprise : saisir le plafond de transaction.
- Le compte est créé via CompteFactory et associé au client.
- Possibilité de créer un deuxième compte pour le même client.

5- Menu principal :

Après la création des comptes, le menu s'affiche :

===== MENU =====

1. Dépôt
2. Retrait
3. Virement
4. Afficher solde
5. Historique
6. Appliquer intérêts (épargne)
0. Quitter

Fonctionnalités

1. Dépôt : saisie du montant → solde crédité.
2. Retrait : saisie du montant → solde débité si autorisé (respect du découvert / plafond).
3. Virement : choisir compte source, compte destination et montant → transfert automatique.
4. Afficher solde : montre le solde actuel du compte sélectionné.
5. Historique : affiche les transactions avec date, type et montant.
6. Appliquer intérêts : uniquement pour les comptes épargne.
7. Quitter : termine l'application et propose de changer de compte si plusieurs existants.

6- Règles et validations :

- Montants : ne peuvent pas être négatifs → `MontantInvalideException`.
- Retrait supérieur au solde ou au découvert/plafond → `SoldeInsuffisantException` ou `LimiteDepasseeException`.
- Taux d'intérêt : doit être ≥ 0 → `MontantInvalideException`.

7- Exemple d'utilisation

Création

- Client : Alice Dupont, CIN: AA12345, Adresse: Paris.
- Compte Courant : N°001, solde 1000, découvert 500.
- Compte Épargne : N°002, solde 2000, taux 0.03.

Menu

1. Dépôt 500 sur N°001 → solde = 1500.
2. Retrait 200 sur N°002 → solde = 1800.
3. Virement 300 de N°001 vers N°002 → N°001: 1200, N°002: 2100.
4. Appliquer intérêts sur N°002 → $2100 + 2100 \times 0.03 = 2163$.

Scénario test

ID	Scénario	Compte concerné	Action	Données d'entrée	Résultat attendu
T1	Création compte valide	Tous	Créer compte	Solde initial = 1000	Compte créé avec succès
T2	Création solde négatif	Tous	Créer compte	Solde initial = -100	<code>MontantInvalideException</code>
T3	Dépôt valide	Tous	Dépôt	Montant = 500	Solde augmenté de 500

ID	Scénario	Compte concerné	Action	Données d'entrée	Résultat attendu
T4	Dépôt invalide	Tous	Dépôt	Montant = -50	MontantInvalideException
T5	Retrait valide	CompteCourant	Retrait	Montant ≤ solde + découvert	Retrait effectué
T6	Dépassement découvert	CompteCourant	Retrait	Dépasse découvert autorisé	SoldeInsuffisantException
T7	Retrait solde insuffisant	CompteÉpargne	Retrait	Montant > solde	SoldeInsuffisantException
T8	Application intérêts	CompteÉpargne	Appliquer intérêts	Taux = 0.03	Solde = solde + intérêts
T9	Plafond dépassé	CompteEntreprise	Retrait	Montant > plafond	LimiteDepasseeException
T10	Virement valide	Deux comptes	Virement	Montant = 300	Source ↓, Destination ↑
T11	Virement solde insuffisant	Deux comptes	Virement	Montant > solde source	SoldeInsuffisantException
T12	Historique	Tous	Consultation	—	Transactions affichées