

Pour mon projet de **gestion de comptes bancaires** en Java, j'ai suivi une approche méthodique en commençant par concevoir les diagrammes de cas d'utilisation et de classes. Ces diagrammes ont aidé à clarifier les fonctionnalités attendues et les relations entre les différentes entités de mon application.

Ensuite, j'ai entamé la programmation en implémentant les classes principales telles que **Compte**, **Client** et **Operation**. J'ai d'abord défini les attributs de ces classes comme privés pour assurer une encapsulation appropriée des données. Ensuite, j'ai traduit les associations entre ces classes, en veillant à représenter correctement les relations un-à-un et un-à-plusieurs.

Par la suite, j'ai défini les **constructeurs** ainsi que **les méthodes getters et setters** pour accéder et modifier les attributs de mes objets. Cette étape était cruciale pour permettre une manipulation sécurisée des données.

Une fois les bases en place, j'ai ajouté la méthode **addOperation()**, qui permet d'ajouter une nouvelle opération à un compte et de mettre à jour le solde en conséquence. Si l'opération est un versement, le solde est augmenté, tandis que s'il s'agit d'un retrait, le solde est diminué, à condition que le montant du retrait ne dépasse pas le solde disponible.

Enfin, j'ai créé la méthode **main** dans la classe **App** pour démarrer l'exécution de mon programme. Dans cette méthode, j'ai initié un client, créé un compte et l'ai associé au client. Ensuite, j'ai ajouté plusieurs opérations au compte, chacune étant associée à ce dernier. Enfin, j'ai affiché les détails du client, du compte et les opérations effectuées, y compris le solde actuel du compte.

Ce processus m'a permis de développer un système de gestion de comptes bancaires fonctionnel en Java, en suivant une approche progressive et méthodique, depuis la conception initiale jusqu'à l'implémentation et la vérification des fonctionnalités.