

PLATEFORME DE GESTION D'UNE BANQUE

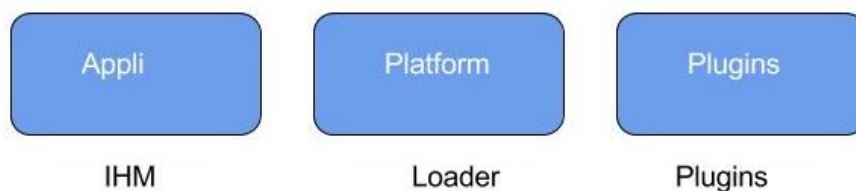
Introduction

Pour ce projet, nous avons choisi de modéliser la gestion d'une banque. Ainsi, en accédant à l'application, un utilisateur pourra visualiser la liste des clients de la banque, la liste des comptes associés à un client, la liste des opérations liés au compte sélectionné. Enfin il pourra créer des nouvelles opérations de crédit ou de débit.

Nous avons réalisé cette application sous forme de plateforme à plugins. Chaque fonctionnalité de l'application est un plugin. L'avantage de cette structure est de permettre à l'application de fonctionner en partie malgré l'absence de certains plugins. L'ajout de fonctionnalité est aussi facilité. Une nouvelle fonctionnalité correspondra à une légère modification de l'IHM, l'ajout d'un bouton par exemple, et à l'ajout d'un plugin.

Structure du projet

Structure des packages

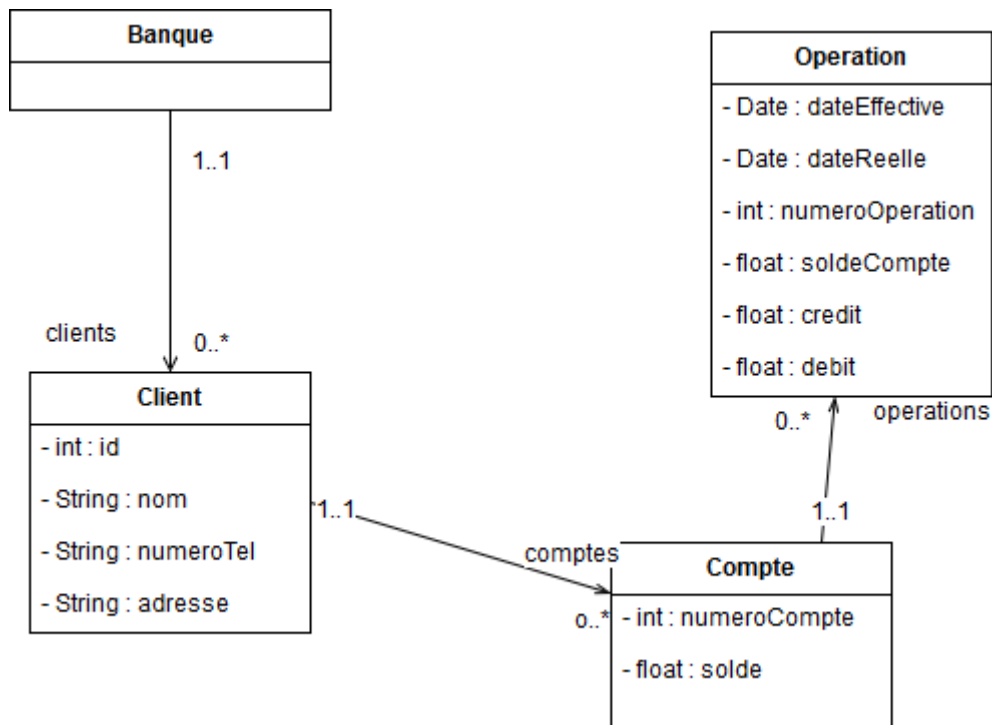


Afin de bien correspondre à une architecture modulaire, notre application est divisée en trois packages. Le package appli correspond plus précisément à la partie client de notre application, on y trouve l'IHM, les interfaces, les fichiers de config et le modèle de données.

Dans l'IHM, nous avons choisi de charger les plugins à partir d'un nom ou d'une partie d'un nom d'un plugin. Ainsi, même si nous ne connaissons pas bien les noms des plugins et que l'IHM ne doit pas les savoir, le nom d'une fonctionnalité correspond au moins en partie au nom du plugins. Nous pouvons donc rechercher le plugin parmi la liste des descriptions de plugins qui a été chargée au lancement de l'application.

L'application doit s'adapter à chaque environnement d'exécution, aussi, le répertoire contenant le fichier de configuration n'est pas toujours placé au même endroit selon l'utilisateur. Pour remédier à cela, nous avons choisi de passer le chemin du répertoire en paramètre.

Modèle de données



Dans cette application, nous avons choisi de ne représenter qu'une seule banque. Ainsi, dans le constructeur d'un objet Banque, nous créons également tous les clients, les comptes et les opérations qui lui sont rattachés.

IHM

The screenshot shows the user interface of the 'Ma Banque' application. It features two main sections: 'Liste des clients :' and 'Liste des comptes :'. The 'Liste des clients :' section displays a table with one client named 'alae'. The 'Liste des comptes :' section displays a table with one account with a balance of 1000000.0. At the bottom, there is a 'Montant :' input field with the value '-10' and two buttons: 'Valider' and 'Opérations'.

Code Client	Nom	Numéro de téléphone	Adresse
1	alae	066666666	Nantes

Numéro Compte	Solde
2	1000000.0

Montant :

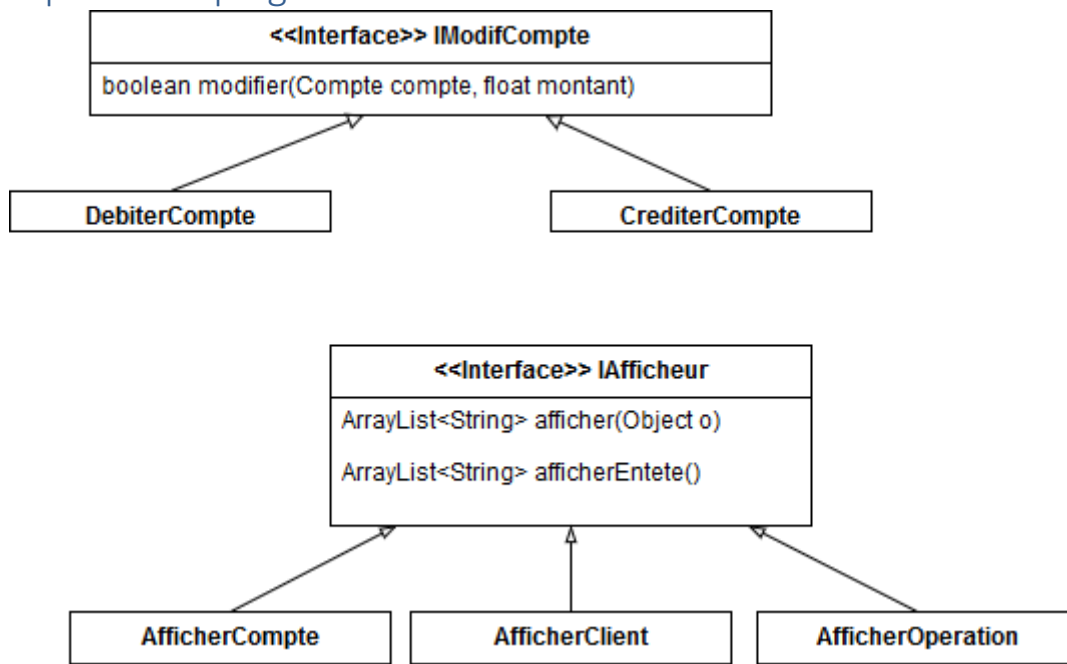
Nous avons créé deux écrans. Le premier présente un tableau qui contient la liste des clients. Lors du clic sur une ligne, la liste des comptes est à son tour chargé dans le tableau des comptes.

Enfin, il y a un champ permettant de renseigner un montant. A la validation, ce montant est ajouté au solde du compte sélectionné s'il est positif, ou bien soustrait.

Numéro d'opérati...	Date effective	Date Réelle	Crédit	Débit	Solde du compte
1	02/05/2017	02/05/2017	0.0	10.0	999990.0
2	02/05/2017	02/05/2017	20.0	0.0	1000010.0

Le bouton "opération" permet d'afficher un deuxième écran contenant un tableau avec la liste de toutes les opérations associées au compte sélectionné.

Description des plugins



Dans ce projet, nous avons réalisé 5 plugins divisés en deux catégories et donc implémentant deux interface : les modifieurs et les afficheurs :

AfficherCompte, AfficherClient et AfficherOperation implémentent tous les trois l'interface IAfficheur. Ces plugins permettent de mettre en forme les données d'un objet sous forme de tableau de chaîne de caractère afin qu'il soit affiché dans un tableau de l'IHM. Il y a ainsi deux fonctions :

- afficherEntete retourne un tableau contenant les noms des attributs de l'objet à afficher dans l'entête du tableau,
- afficher : retourne un tableau avec les valeurs de l'objet

Nous utilisons ces plugins afin de compléter le tableau des clients dans l'IHM. Lors de la sélection d'un client, nous chargeons le plugin suivant : AfficherCompte, afin de charger la liste des comptes associés à un client. Enfin, lors du clic sur le bouton "opérations", et si un client est sélectionné, nous chargeons le dernier plugin AfficherOperation afin de compléter le tableau des opérations dans le deuxième écran.

DebiterCompte et CrediterCompte implémentent l'interface IModifCompte. Tous deux agissent sur le solde d'un compte et créent une nouvelle opération. Nous trouvons la fonction modifier qui prend en paramètre un compte et un montant. Si c'est un montant positif, il est ajouté au solde du compte, s'il est négatif, il est soustrait du solde du compte. Dans les deux cas, une nouvelle opération avec la date, le montant et le solde du compte est créée.

Ces deux plugins sont chargés lors du clic sur le bouton "valider" et utilisés en fonction du montant indiqué par l'utilisateur.