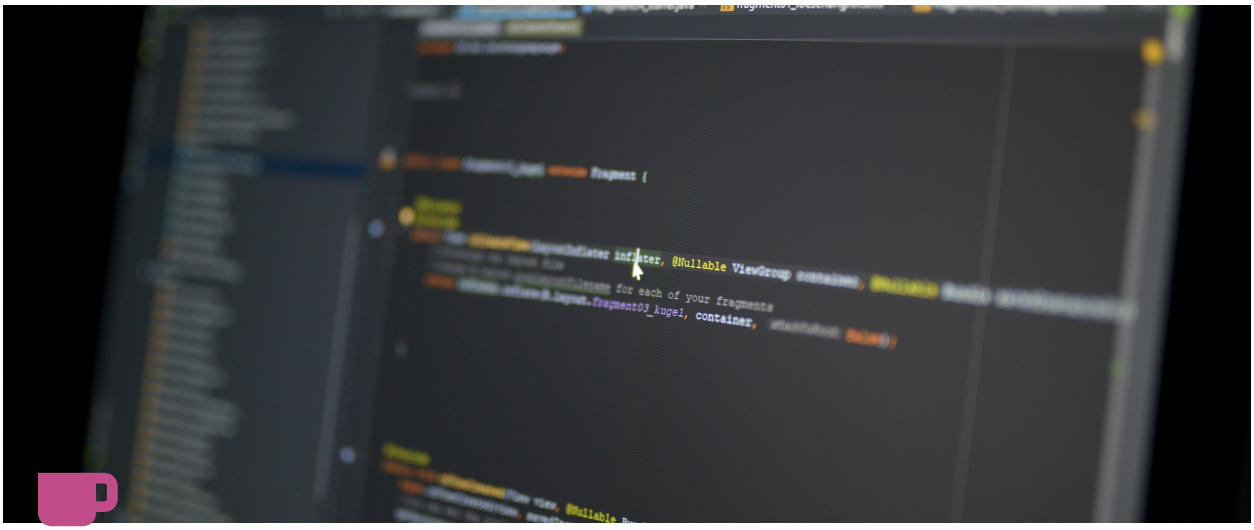


Java Projects

Cette page recense les différents projets en Java que j'ai réalisé :

Shifumi :

Ce script Java est une version du jeu "Shifumi", aussi connu sous le nom de "Pierre-Feuille-Ciseaux", avec une variante qui inclut le "puits". Voici un résumé de son fonctionnement :

1. **Structure générale :**
   - Le script commence par importer la classe `Scanner` pour gérer l'interaction avec l'utilisateur via la console.
   - La classe `shifumi` contient la méthode principale `main`, qui gère l'exécution du jeu.
2. **Déroulement du jeu :**
   - L'utilisateur choisit d'abord le nombre de points nécessaires pour gagner la partie parmi trois options (3, 5, ou 10).
   - Ensuite, l'utilisateur sélectionne un mode de jeu :

    - **Mode classique** : pierre, feuille, ciseaux.
    - **Mode avec puits** : ajoute l'option "puits" aux choix disponibles.
  - Une fois le mode sélectionné, le jeu se déroule avec une série de tours où l'utilisateur et l'IA choisissent chacun une option. Les choix sont comparés pour déterminer qui gagne chaque tour.
3. **Gameplay :**
  - L'IA génère aléatoirement son choix (pierre, feuille, ciseaux, ou puits selon le mode).
  - L'utilisateur choisit ensuite sa propre action.
  - Selon les choix faits, le script détermine le gagnant du tour :
    - Si les deux choisissent la même action, c'est une égalité.
    - Sinon, des conditions vérifient qui est le gagnant du tour et mettent à jour les scores en conséquence.
  - Les scores des deux joueurs (utilisateur et IA) sont affichés après chaque tour.
4. **Fin de partie :**
  - La partie continue jusqu'à ce que l'un des deux joueurs atteigne le score maximum défini par l'utilisateur.
  - Quand la partie se termine, le script annonce si l'utilisateur a gagné ou perdu la partie.
  - L'utilisateur peut ensuite choisir de rejouer ou de quitter.
5. **Points importants du code :**
  - **Utilisation de la boucle** `while` : Pour permettre de rejouer la partie à volonté tant que l'utilisateur le souhaite.
  - **Mode de jeu étendu** : En mode "avec puits", le puits a des interactions spécifiques, par exemple, la pierre perd contre le puits, mais les ciseaux gagnent contre le puits.
  - **Interaction console** : Le jeu repose entièrement sur des saisies utilisateur via la console, utilisant la classe `Scanner` pour les entrées.

En résumé : Ce script Java propose une version interactive du jeu "Pierre-Feuille-Ciseaux", avec une option supplémentaire (puits) et un système de points pour décider du gagnant. Il permet à l'utilisateur de rejouer plusieurs fois en ajustant les paramètres de la partie. Le code est écrit de manière assez directe, utilisant des boucles `while` et des structures `switch` pour gérer les choix de l'utilisateur et de l'IA.

Application bancaire :

Cette application est une application bancaire de gestion de clients et de comptes, construite en Java avec une interface graphique moderne utilisant **Swing**. Cette application est composée de quatre principales fonctionnalités :

1. **Fenêtre Principale (`BanqueIHM`) :**
    - **Description** : La fenêtre d'accueil de l'application qui permet de naviguer vers les différentes fonctionnalités : création de clients, gestion des clients, création et gestion des comptes, et opérations bancaires.
    - **Fonctionnalités** : Quatre boutons principaux redirigent vers les fenêtres correspondantes.
2. **Fenêtre de Création de Client (`CreerClientWindow`) :**
    - **Description** : Une interface permettant de saisir les informations de base d'un client (nom, prénom, adresse) pour l'ajouter à la banque.
    - **Fonctionnalités** : Les utilisateurs peuvent entrer les informations et créer un client. Un message de confirmation s'affiche une fois l'opération

réussie.

3. **Fenêtre de Gestion des Clients (`GestionClientWindow`)** :
    - **Description** : Une interface permettant de visualiser, modifier, et supprimer les clients enregistrés dans la banque.
    - **Fonctionnalités** : Une liste des clients est affichée, et les utilisateurs peuvent sélectionner un client pour voir ou modifier ses informations. Une option de suppression est disponible, avec une confirmation avant l'exécution.
4. **Fenêtre de Création/Gestion des Comptes (`CreerGererCompteWindow`)** :
    - **Description** : Une interface permettant de créer des comptes pour les clients existants et de gérer ces comptes (modification et suppression).
    - **Fonctionnalités** : Les utilisateurs peuvent choisir un client, définir le type de compte (compte courant ou épargne), entrer le solde initial et d'autres détails, puis créer le compte. Une liste des comptes existants est également affichée pour permettre la gestion.
5. **Fenêtre d'Opérations Bancaires (`OperationsBancairesWindow`)** :
    - **Description** : Une interface pour effectuer des opérations bancaires sur les comptes, comme le crédit, le débit, et l'ajout d'intérêts pour les comptes d'épargne.
    - **Fonctionnalités** : Les utilisateurs peuvent sélectionner un compte, entrer un montant, et choisir l'opération à effectuer. Un historique des opérations est affiché pour plus de clarté.

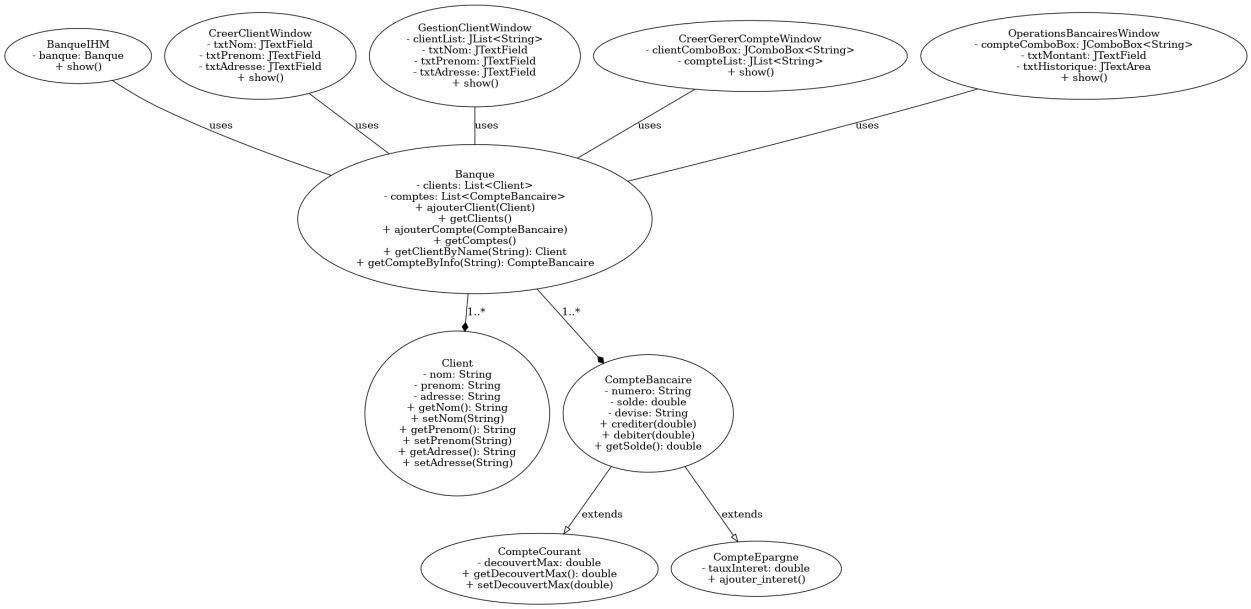
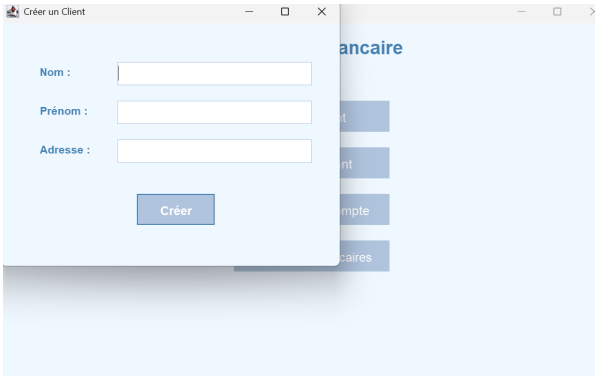
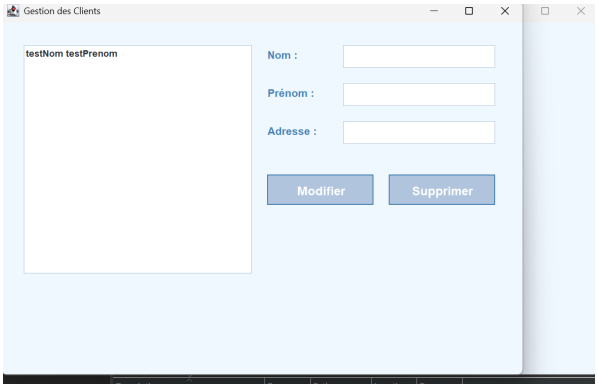


Diagramme de classe UML

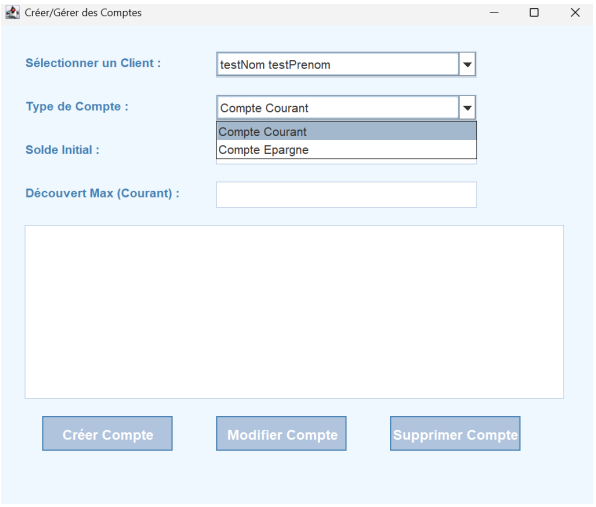
fenêtre principale



Fenêtre pour créer un client



Fenêtre pour gérer les clients



Fenêtre pour créer et gérer les comptes